Wide-band speech teleconferencing over an integrated network

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

Additional Information:

• A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: [https://dspace.lboro.ac.uk/2134/14410](https://dspace.lboro.ac.uk/2134/14410)

Publisher: © Victoria Jayne Hardman

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
BL-DSC no.:- DX 180955
Wide-band Speech Teleconferencing 
over an Integrated Network

by

Victoria Jayne Hardman

A Doctoral Thesis

Submitted in partial fulfilment of
the requirements for the award of

Doctor of Philosophy
of the Loughborough University of Technology

November 30, 1993

© by Victoria Jayne Hardman, 1993
To Michael,
and also my Mum, Dad, Aunty Mary,
Cathy, Paul and Samantha,
and to the memory of Gar, Neanan,
Mark and Nana.
Acknowledgements

I would like to thank my supervisors, Professor J.W.R. Griffiths, and recently Dr. D.J. Parish, for their invaluable help during this research.

I would also like to thank my friends on the Unison project: Dave Parish, Tim Rodgers, Brendan Murphy, Guojun Lu, Mahboob Siddiqui; and my other friends from around the department: Andrew Afzal, Murray Holt, Dave Goodsdon, Tahseen Rafik, Sheila Carson, and all in the workshop and stores.

I would like to extend a very big 'thank you' to the people who have helped publish this thesis: Michael, Geoff, Dave and especially Karyn.

I would like to say how much I appreciate Michael, my Mum and Dad, Aunty Mary, Cathy, Paul, and Samantha for their long suffering emotional and often financial support. I would also like to thank Marge, Chris, Lee and my adopted Grandmother.

Last but not least, I would like to say a big 'thank you' to Chris+Ash, James, Lesley, Ron, Janet+John, Andrea+Paul, Karyn+Geoff, Monica+Shaun, Pepsi, Marmalade, Sylvestor, Gordon's and Murphy's for relieving my stress, and to the salamander for contributing to it.
Abstract

There is a lot of interest at present in the provision of an integrated network which can carry both voice and data traffic. The recent emergence of asynchronous transfer mode (ATM) networks has made this desire a reality. The Unison project built a network consisting of local area networks (LANs) and exchange LANs connected at local and remote sites by an ATM-like backbone network. This extends the facilities of the LAN to a wider area. The Unison network is also an intelligent network, because it provides a service to the user (such as dynamic data-bases, which ease call set-up and facilitate user migration).

An important feature of the Unison network is also the provision of applications which demonstrate the suitability of the network to carry both voice and data traffic, and which exploit the intelligent network features. This thesis describes a very important application: the provision of a two- and three-party wide-band speech teleconferencing system.

The first part of the thesis deals with the provision of a two-party teleconferencing system, based on a wide-band speech codec. The codec is interfaced to the Unison Network via transputers (parallel processors). This thesis considers in detail the voice protocols which make up part of the network interface. The work includes the set-up and control benefits gained from interaction with a desk-top workstation, which can also be used to guide other multi-media services (such as video). A topic which has been greatly under-stressed in other similar research (i.e. the acoustic aspects of the system) is also investigated.

The second part of the thesis deals with the logical expansion of the two-party system to a three- or more party scenario across the Unison network. Towards this end, a bridge has been designed and implemented based on transputers. The problems associated with matching the DSP algorithm used in the codecs with that implemented in the bridge is also discussed. The same systems considerations addressed by the two-party version are expanded to operate with the three-party teleconferencing system.
Table of Contents

Table of Contents ......................................................................................................................... i
List of Figures ............................................................................................................................... vi
List of Tables ................................................................................................................................. ix

Chapter 1 ....................................................................................................................................... 1
  1.1 Introduction ............................................................................................................................ 1
  1.2 Background ............................................................................................................................ 2
    1.2.1 Networks ......................................................................................................................... 2
    1.2.2 Terminals ......................................................................................................................... 4
  1.3 Evolution ................................................................................................................................. 5
    1.3.1 Networks ......................................................................................................................... 5
    1.3.2 Terminal Evolution .......................................................................................................... 6
    1.3.3 Project Unison Research ............................................................................................... 7
  1.4 Aims ....................................................................................................................................... 9
    1.4.1 Unison Aims ................................................................................................................... 9
    1.4.2 Thesis Aims .................................................................................................................... 9
  1.5 Thesis Structure .................................................................................................................... 11
  1.6 Collaboration .......................................................................................................................... 12

Chapter 2 ....................................................................................................................................... 13
  2.1 Introduction ............................................................................................................................ 13
  2.2 The Integrated Office ............................................................................................................. 14
    2.2.1 Advantages of Integrating the Office ............................................................................ 14
    2.2.2 Design Considerations ................................................................................................ 14
    2.2.3 Distributed Computing - System Interconnection ....................................................... 16
    2.2.4 Intelligent Network Services ....................................................................................... 18
    2.2.5 The Unison Integrated Office ....................................................................................... 19
  2.3 A Suitable Network ................................................................................................................ 21
    2.3.1 The Ideal Situation ........................................................................................................ 21
    2.3.2 The Present Situation .................................................................................................... 22
  2.4 Unison Network Overall Architecture .................................................................................. 25
  2.5 The Unison Network Initial Version ..................................................................................... 27
    2.5.1 The Cambridge Ring Architecture ............................................................................... 27
    2.5.2 Protocols ....................................................................................................................... 28
    2.5.3 Naming, Addressing and Routing in Universe and the Initial Unison Network .......... 30
  2.6 The Unison Network Final Version ....................................................................................... 31
    2.6.1 The Cambridge Fast Ring ............................................................................................ 31
    2.6.2 Protocols ....................................................................................................................... 32
    2.6.3 Naming and Addressing in Unison ............................................................................... 34
    2.6.4 Architecture Rationale ............................................................................................... 34
  2.7 Conclusion .............................................................................................................................. 39
    2.7.1 Unison: Research into Multi-Media Communications ................................................ 39
    2.7.2 Unison: Research into Multi-Media Networks ............................................................. 39
    2.7.3 The Importance of Voice within Unison ...................................................................... 39
Chapter 3 ........................................................................................................................................ 41
  3.1 Introduction ................................................................................................................................. 41
  3.2 The Component Parts of a Voice System ...................................................................................... 42
  3.3 The SB-ADPCM Codec, Voice Coding Schemes and Bit-Rate Reduction ................................... 44
    3.3.1 Introduction .......................................................................................................................... 44
    3.3.2 The Voice Signal ...................................................................................................................... 44
    3.3.3 Performance Criteria ................................................................................................................. 48
    3.3.4 Wave-form Coding Techniques ................................................................................................. 48
    3.3.5 Conclusion ................................................................................................................................ 59
  3.4 Packetised Voice .......................................................................................................................... 61
    3.4.1 Introduction ................................................................................................................................. 61
    3.4.2 Basic Functions of a Voice Protocol ......................................................................................... 61
    3.4.3 Packetised Voice Problems ....................................................................................................... 61
    3.4.4 Voice Protocol Design ................................................................................................................. 67
    3.4.5 Conclusion ................................................................................................................................ 71
  3.5 Acoustics and Audio Systems ..................................................................................................... 73
    3.5.1 The Performance of the Hearing Component ............................................................................ 73
    3.5.2 Audio Equipment ....................................................................................................................... 78
    3.5.3 Conclusion ................................................................................................................................ 79
Chapter 4 ........................................................................................................................................... 81
  4.1 Introduction .................................................................................................................................. 81
  4.2 The Universe Network and Toll Quality Voice System ............................................................... 82
    4.2.1 Introduction ................................................................................................................................. 82
    4.2.2 The Universe Network ................................................................................................................. 82
    4.2.3 The Universe Voice Protocol ....................................................................................................... 82
    4.2.4 Conclusion ................................................................................................................................ 86
  4.3 Modifications made for the Unison Prototype .............................................................................. 87
    4.3.1 Introduction ................................................................................................................................ 87
    4.3.2 Equipment Modifications .......................................................................................................... 87
    4.3.3 Software Modifications .............................................................................................................. 88
    4.3.4 Conclusion ................................................................................................................................ 89
  4.4 Performance Assessment of the Prototype Unison Two-party Wide-band Speech System .......... 90
    4.4.1 Introduction ................................................................................................................................. 90
    4.4.2 Performance Over the Prototype Unison Network ..................................................................... 90
    4.4.3 Network Performance ............................................................................................................... 92
    4.4.4 Performance Conclusions for the Voice Protocol ................................................................... 93
Chapter 5 ............................................................................................................................................. 95
  5.1 Introduction .................................................................................................................................. 95
  5.2 Hands-Free Teleconferencing Audio System ............................................................................... 96
    5.2.1 Requirements ............................................................................................................................... 96
    5.2.2 Basic Components ...................................................................................................................... 96
    5.2.3 Design Considerations ................................................................................................................. 97
    5.2.4 Audio Equipment ....................................................................................................................... 99
    5.2.5 A Hands-free Teleconferencing System .................................................................................. 102
    5.2.6 Subjective Audio Performance ................................................................................................. 105
    5.2.7 Conclusion ................................................................................................................................ 105
## Table of Contents

### Chapter 5

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5.3</strong> Voice Protocol Design</td>
<td>106</td>
</tr>
<tr>
<td>5.3.1 Requirements</td>
<td>106</td>
</tr>
<tr>
<td>5.3.2 Unison Network Performance Survey</td>
<td>106</td>
</tr>
<tr>
<td>5.3.3 Design</td>
<td>108</td>
</tr>
<tr>
<td>5.3.4 Conclusion</td>
<td>111</td>
</tr>
<tr>
<td><strong>5.4</strong> Choice of Processor - the Transputer and Occam</td>
<td>112</td>
</tr>
<tr>
<td>5.4.1 Choice of Processor</td>
<td>112</td>
</tr>
<tr>
<td>5.4.2 The Suitability of the Transputer as a Real-Time Embedded Controller</td>
<td>117</td>
</tr>
<tr>
<td>5.4.3 Conclusion</td>
<td>117</td>
</tr>
<tr>
<td><strong>5.5</strong> Hardware Design and Implementation</td>
<td>119</td>
</tr>
<tr>
<td>5.5.1 Requirements</td>
<td>119</td>
</tr>
<tr>
<td>5.5.2 Design Considerations</td>
<td>119</td>
</tr>
<tr>
<td>5.5.3 Circuitry</td>
<td>121</td>
</tr>
<tr>
<td>5.5.4 Circuitry Initial Conditions</td>
<td>124</td>
</tr>
<tr>
<td>5.5.5 Servicing the Hardware</td>
<td>125</td>
</tr>
<tr>
<td>5.5.6 Final Circuit and PCB Layout</td>
<td>126</td>
</tr>
<tr>
<td>5.5.7 X21 Interface Board Testing</td>
<td>128</td>
</tr>
<tr>
<td>5.5.8 Conclusion</td>
<td>130</td>
</tr>
<tr>
<td><strong>5.6</strong> Concurrent Real-Time Software on the Transputer</td>
<td>131</td>
</tr>
<tr>
<td>5.6.1 Introduction</td>
<td>131</td>
</tr>
<tr>
<td>5.6.2 Real-Time Concurrent Software Problems</td>
<td>131</td>
</tr>
<tr>
<td>5.6.3 The ALT Construct</td>
<td>132</td>
</tr>
<tr>
<td>5.6.4 PRI ALT Construct</td>
<td>134</td>
</tr>
<tr>
<td>5.6.5 Buffer Structure</td>
<td>134</td>
</tr>
<tr>
<td>5.6.6 Conclusion</td>
<td>136</td>
</tr>
<tr>
<td><strong>5.7</strong> Voice Protocol Implementation in Occam - A Parallel Approach</td>
<td>137</td>
</tr>
<tr>
<td>5.7.1 Requirements</td>
<td>137</td>
</tr>
<tr>
<td>5.7.2 Specification</td>
<td>137</td>
</tr>
<tr>
<td>5.7.3 System Testing</td>
<td>147</td>
</tr>
<tr>
<td>5.7.4 Conclusion</td>
<td>152</td>
</tr>
<tr>
<td><strong>5.8</strong> Integrating the Wide-band Speech System into the Unison Office</td>
<td>153</td>
</tr>
<tr>
<td>5.8.1 Requirements</td>
<td>153</td>
</tr>
<tr>
<td>5.8.2 Distributed Control Implementation</td>
<td>153</td>
</tr>
<tr>
<td>5.8.3 Office Integration</td>
<td>156</td>
</tr>
<tr>
<td>5.8.4 Conclusion</td>
<td>156</td>
</tr>
<tr>
<td><strong>5.9</strong> Conclusion</td>
<td>157</td>
</tr>
</tbody>
</table>

### Chapter 6

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>159</td>
</tr>
<tr>
<td><strong>6.2</strong> Three-party Teleconferencing Systems and SB-ADPCM</td>
<td>160</td>
</tr>
<tr>
<td>6.2.1 Requirements</td>
<td>160</td>
</tr>
<tr>
<td>6.2.2 Stream Mixing</td>
<td>160</td>
</tr>
<tr>
<td>6.2.3 Digital Teleconferencing Systems Using PCM Coded Speech</td>
<td>161</td>
</tr>
<tr>
<td>6.2.4 Digital Teleconferencing Systems Using ADPCM Encoded Speech</td>
<td>164</td>
</tr>
<tr>
<td>6.2.5 Conclusion</td>
<td>164</td>
</tr>
<tr>
<td><strong>6.3</strong> Equipment Available and Processor Choice</td>
<td>165</td>
</tr>
<tr>
<td>6.3.1 Requirements</td>
<td>165</td>
</tr>
<tr>
<td>6.3.2 Equipment Available</td>
<td>165</td>
</tr>
<tr>
<td>6.3.3 Choice of Codec</td>
<td>166</td>
</tr>
<tr>
<td>6.3.4 Processor Choice</td>
<td>166</td>
</tr>
<tr>
<td>6.3.5 Conclusion</td>
<td>167</td>
</tr>
</tbody>
</table>
6.4 System Development Considerations ................................................. 168
  6.4.1 Requirements .................................................................................. 168
  6.4.2 Development .................................................................................. 168
  6.4.3 Conclusion ...................................................................................... 171

6.5 The Effect of Finite Word Length Arithmetic in Digital Systems .... 172
  6.5.1 Introduction ..................................................................................... 172
  6.5.2 Number Representations ................................................................. 172
  6.5.3 Quantization and Overflow Effects in Digital Filters ...................... 175
  6.5.4 Conclusion ...................................................................................... 178

6.6 Non-Standard SB-ADPCM Algorithm Development ...................... 179
  6.6.1 Introduction ..................................................................................... 179
  6.6.2 Information Available .................................................................... 179
  6.6.3 Comparison of Processor Behaviour .............................................. 179
  6.6.4 SB-ADPCM Algorithm Overview .................................................. 180
  6.6.5 SB-ADPCM Component Operation .............................................. 182
  6.6.6 Simulation Studies and Algorithm Verification ............................... 185
  6.6.7 Transputer Implementation ............................................................ 185
  6.6.8 Conclusion ...................................................................................... 186

6.7 SB-ADPCM Mixing ........................................................................... 187
  6.7.1 Requirements .................................................................................. 187
  6.7.2 SB-ADPCM Mixing Options ......................................................... 187
  6.7.3 Conclusion ...................................................................................... 188

6.8 Conclusion .......................................................................................... 189

Chapter 7.................................................................................................... 191

7.1 Introduction .......................................................................................... 191

7.2 Design Considerations ......................................................................... 192
  7.2.1 Introduction ..................................................................................... 192
  7.2.2 Existing Components ..................................................................... 192
  7.2.3 Expected System Layout and Extra Equipment Needed ................. 192
  7.2.4 Extra Software Required ............................................................... 193
  7.2.5 Conclusion ...................................................................................... 194

7.3 Bridge Hardware .................................................................................. 195
  7.3.1 Requirements .................................................................................. 195
  7.3.2 Processors Required for the Bridge ................................................ 195
  7.3.3 Bridge System Clocking ................................................................. 197
  7.3.4 Hardware Implementation .............................................................. 197
  7.3.5 Hardware Testing ........................................................................... 198
  7.3.6 Conclusion ...................................................................................... 198

7.4 Bridge System Software - A Parallel Approach ................................. 199
  7.4.1 Requirements .................................................................................. 199
  7.4.2 Software Required ......................................................................... 199
  7.4.3 Software Specification ................................................................. 199

7.5 Conclusion ............................................................................................ 211

Table of Contents
Chapter 8 .................................................................................................................. 212
  8.1 Conclusions ....................................................................................................... 212
      8.1.1 Introduction ............................................................................................... 212
      8.1.2 The Local Area Network ........................................................................... 213
      8.1.3 The Wide Area Network .......................................................................... 213
      8.1.4 Unison Integrated Office ......................................................................... 214
      8.1.5 Managing Voice Traffic Across an ATM-like Network ......................... 214
      8.1.6 Acoustical Aspects of Hands-Free Teleconferencing .............................. 215
      8.1.7 The Suitability of the Transputer as a Real-Time Embedded Controller .... 215
      8.1.8 Three-party Teleconferencing ................................................................. 217
  8.2 Possible Future Extensions of this Research ..................................................... 218
      8.2.1 Introduction ............................................................................................... 218
      8.2.2 Full Integration of the Bridge into the Unison Environment ................... 218
      8.2.3 System Clocking ....................................................................................... 218
      8.2.4 The CFR Interface ................................................................................... 219
      8.2.5 Extension of the Wide-band Speech System to More than Three Parties ... 219
      8.2.6 Alteration of Three-party Wide-band Speech System to Use the G.722 Standard .......................................................... 219
      8.2.7 Graceful Bit-Rate Reduction Using Embedded Coding ......................... 219
      8.2.8 Voice Mixing ............................................................................................. 220
      8.2.9 Voice Imaging ........................................................................................... 221
References ................................................................................................................. 222
List of Figures

Figure 1: Tightly-coupled system ................................................................. 16
Figure 2: Loosely-coupled system ............................................................... 17
Figure 3: The Unison integrated office ....................................................... 19
Figure 4: Unison network architecture ...................................................... 25
Figure 5: Cambridge Ring slot structure .................................................. 27
Figure 6: Cambridge Ring interface hardware ......................................... 28
Figure 7: Basic block protocol ................................................................. 28
Figure 8: Single-shot protocol ................................................................. 29
Figure 9: Open/Openack protocol ............................................................ 29
Figure 10: Cambridge Fast Ring slot structure ....................................... 31
Figure 11: Cambridge Fast Ring interface hardware ............................... 32
Figure 12: Call set-up within the same site ............................................. 37
Figure 13: Call set-up across the ISDN ................................................... 38
Figure 14: Packet Voice System Composition ......................................... 42
Figure 15: Long-Term Power Density Spectrum .................................... 45
Figure 16: The Contribution of Various Frequencies to Articulation ........ 46
Figure 17: Short-Term Statistics for Typical Voiced and Un-Voiced Speech Segments ................................................................. 47
Figure 18: Generalised Block Diagram of a Time-Domain Coding Algorithm ............ 50
Figure 19: Block Diagram of an Adaptive Quantization Scheme ............ 51
Figure 20: Block Diagram for a DPCM Codec ......................................... 52
Figure 21: Block Diagram of an ADPCM Codec ..................................... 53
Figure 22: The Effect of Noise Spectral Shaping using a Pre/De-Emphasis Filter ........ 56
Figure 23: Pre and Post Filtering Arrangement for Noise Spectral Shaping ........ 56
Figure 24: Frequency Band Split into Sub-Bands ..................................... 57
Figure 25: Embedded SB-ADPCM Codec ............................................. 59
Figure 26: Sources of Echo in the Circuit-Switched PSTN Resulting from 2/4 wire Conversion Points ........................................................... 64
Figure 27: Contour Lines of Equal Loudness for Normal Ears .................. 74
Figure 28: The Articulation of a Speaker in Auditoriums of Various Volumes ............ 76
Figure 29: Illustration of Normal Modes in Response of a Room 5.2x6.4x2.7 metres .................. 77
Figure 30: The Syllable Articulation of a Sound Reproducing System as a Function of the Reproduced Sound Level for Two Conditions of Operation .................. 78
Figure 31: The Universe Voice System Hardware .......................................................... 83
Figure 32: The Universe Toll Quality Voice State Machine ........................................... 85
Figure 33: Unison Prototype Wide-band Voice System .................................................. 87
Figure 34: Local Ring Only Environment ........................................................................ 90
Figure 35: Looped Back Through a Single Portal Environment ....................................... 91
Figure 36: Over the Interconnecting Megastream Environment ..................................... 92
Figure 37: Basic Audio ‘Hands-free’ Teleconferencing Equipment .................................. 96
Figure 38: The Frequency Response of the Complete Codec ........................................... 97
Figure 39: Public Address System Layout ....................................................................... 100
Figure 40: The Unison Teleconferencing Desk Layout ......................................................... 102
Figure 41: Transputer Architecture .................................................................................. 113
Figure 42: Transputer’s Registers in a Linked Process List .............................................. 115
Figure 43: Bit and Byte Clock Generation Circuitry ......................................................... 121
Figure 44: Serial to Parallel Data Conversion Circuitry ..................................................... 122
Figure 45: Hand-shake Signals Circuitry ............................................................................ 125
Figure 46: Full Circuit for the Transputer to X.21 Interface ............................................ 127
Figure 47: X21 Interface Test Program ............................................................................. 128
Figure 48: A Truly Parallel Buffering System .................................................................... 135
Figure 49: Buffer pool and Regulator .............................................................................. 135
Figure 50: Voice Protocol Design Proposal ...................................................................... 141
Figure 51: Voice Protocol Network Decomposition ......................................................... 143
Figure 52: Voice Protocol Testing Software Structure ....................................................... 148
Figure 53: Software Latency Measurement - Process Structure ....................................... 152
Figure 54: The Unison Wide-band Speech State Machine ................................................. 155
Figure 55: Network Bandwidth Required for Analogue Mixing ....................................... 160
Figure 56: Bandwidth Required When Using the Network’s ........................................ 162
Figure 57: Network Bandwidth Requirements for Bridge Teleconferencing ............... 163
Figure 58: Algorithm Verification ........................................................................... 169
Figure 59: Real-Time Verification ........................................................................... 170
Figure 60: Mixing Test Circuitry ........................................................................... 170
Figure 61: Mixing Test Configuration ...................................................................... 171
Figure 62: The Rounding Quantization Characteristic ............................................ 173
Figure 63: The Value Truncation Characteristic ..................................................... 173
Figure 64: The Magnitude Truncation Characteristic ............................................. 173
Figure 65: The Saturation Overflow Characteristic .............................................. 174
Figure 66: The Zeroing Overflow Characteristic .................................................... 174
Figure 67: The ‘Saw-tooth’ Overflow Characteristic ............................................. 175
Figure 68: Limiting the Word-Length in a Recursive Digital Filter ....................... 176
Figure 69: A Second Way of Limiting the Word Length in a Recursive Digital Filter ......................................................................................................................... 177
Figure 70: Block Diagram of One Band of the Non-Embedded .............................. 181
Figure 71: Block Diagram of One Band of the Embedded ..................................... 181
Figure 72: A 3-Bit Uniform Mid-riser Quantizer ................................................... 182
Figure 73: A Fourth Order All-Pole Linear Predictor with Feed-back ..................... 184
Figure 74: Encoder and Decoder Process Structure ............................................. 185
Figure 75: Expected Three-party System Layout ................................................ 193
Figure 76: Bridge System Processor Diagram ..................................................... 196
Figure 77: Error Signals Circuitry ........................................................................ 198
Figure 78: Bridge Design Proposal ..................................................................... 201
Figure 79: Three-Stream CFR Interface - Network Decomposition ..................... 202
Figure 80: Voice Protocol Sub-system - Network Decomposition ....................... 203
Figure 81: Mixing Process Sub-system-Network Decomposition ......................... 204
Figure 82: Statistical Reporting Sub-system ....................................................... 205
Figure 83: CFR Interface Statistical Reporting Sub-system-Network Decomposition ......................................................................................................................... 206
List of Tables

Table 1: Perceptual performance degradation for various packet lengths .................. 70
Table 2: Results for the Receive Buffer and an Increase in Delay ......................... 149
Table 3: Results for the Receive Buffer and a Decrease in Delay ....................... 150
Table 4: Behaviour of the Voice Protocol Software .......................................... 151
Chapter 1
Introduction

1.1 Introduction

Recent advances in communications have led to the development of *Integrated* networks, which aim to transport all types of media, such as voice, video and data communications in one 'network'. Future development will lead to the provision of *Intelligent* networks (intelligent, because the network itself provides a service), that will facilitate the rapid deployment of applications, and simplify their management.

As networks have evolved, so too have the information processing terminals. Originally, terminals were either voice or data; now, multi-media terminals are developing that can handle additional types of media, such as video. These *Multi-media systems* will enhance communication in both the office environment, and the home.

This thesis is concerned with the development of voice terminals as part of a multi-media system, to provide higher quality speech communications over an integrated, intelligent network. The work is practical, and investigates the incorporation of high-quality voice into experimental integrated offices, interconnected by a prototype integrated, intelligent network.

A further focus of this research has been the use of parallel processing techniques to connect an existing voice terminal to the integrated, intelligent network, thereby providing a two-party facility. Later research concentrated on a three-party voice connection. The system not only used parallel processing techniques to interface with the network, but also used concurrent software to implement the voice coding scheme, and to mix three streams of speech in real-time.
1.2 Background

1.2.1 Networks

Traffic can be categorised as being one of two types: constant bit-rate (CBR) traffic, or variable bit-rate (VBR) traffic. CBR traffic (such as is transmitted by most voice applications) requires connections with low delay, but is fairly tolerant of errors. Conversely, VBR traffic (such as is transmitted by data applications) requires error-free connections, but is relatively immune to variations in delay. Voice and data traffic have traditionally been carried by two different networks: a circuit-switched network for voice, where the route through the network has been fixed for the duration of the call, and a packet-switched network for data, where each packet can take a different route through the network.

The two types of network can each be subdivided further: private and public networks for the circuit-switched type, and local and wide networks for the packet-switched type. These distinctions are important, as evolution in the two sizes of network is driven by different factors.

Networks for the private and local area are generally intended for the business arena, and are more tailored to specific customers. The asset outlay for the network is relatively small, and as network improvements and the provision of better services increases the customer's competitiveness, the customer (and hence the network provider) are driven to invest in the newest technology.

Networks for the public and wide area are usually owned by a public switching company. Until recently, these companies, provided a service which had little or no competition in the market place. The public and wide-area networks also have a greater asset outlay, and so can only be gradually replaced with new technology. Consequently, evolution in the public and wide areas tends to be slow, and the networks consist of a range of technologies.

1.2.1.1 Private Voice Networks

Private voice networks are based on key systems and private branch exchanges (PBXs). Originally analogue but increasingly digital, key systems and PBXs offer services such as direct dial-in capability, answering positions, line hunting, call forwarding, call transfer, call pick-up, etc.
1.2.1.2 Local Area

Local data networks were composed originally of terminal access to a main-frame computer via a dedicated cable. More recently, the main-frame has been replaced by a number of stand-alone personal computers (PCs), networked together by a local area network (LAN). LANs offer the capability of sharing expensive resources, such as printers, and large data-bases. The most common LAN technologies are Ethernet and token ring.

1.2.1.3 Public Voice Networks

Until recently, public voice networks were mostly electro-mechanically based. While they have large digital portions today, the final connection to the subscriber still uses analogue technology. Digitalisation of the network has mainly been in the trunk network (CCITT #7 signalling trunks). The trunk network is composed of a back-bone network which connects large regional switches or exchanges together. Each regional exchange connects a number of local exchanges together, and gives access to the trunk network. The services offered by the network are a sub-set of those typically offered by private voice networks: facilities such as answering position, call transfer, and call pick-up are not available.

1.2.1.4 Wide-area Data Networks

Wide-area data networks use leased voice lines to interconnect sites. The facilities offered include electronic mail and access to remote hosts. The most common wide-area data network protocol is X.25.

1.2.1.5 A Combined Voice and Data Network

Recently, the provision of the integrated services digital network (ISDN) has provided the basis for a single wide-area network that can carry voice and data traffic across the same route, and which uses end-to-end digital connectivity all the way to the subscriber. Using the voice network exchanges (and its long-haul, no. 7 signalling trunk network), the ISDN gives the subscriber the option of choosing a method of communication on a per-call basis: voice, slow-scan video, or data.

The ISDN network provides more complex services, such as calling line identification. The network is still essentially circuit-switched, and occurs in two variations of channel combinations: 2 fixed-size channels for either voice or data, a signalling channel and a synchronisation channel (basic rate interface), or 30 fixed-size channels, a signalling channel and a synchronisation channel (primary rate interface).
The deployment of ISDN access capabilities is limited, and the more complex services (such as calling line identification) are lost if the call has to route over an analogue portion.

1.2.2 Terminals

1.2.2.1 Data Processing Terminals

The field of information processing in the office today is based on the PC and the workstation. The PC is typically used for word-processing/desk-top publishing in the office, and the more powerful workstation is used in the engineering community for software and hardware development. Recent technological advances and component cost reduction have seen the terminals’ development converging; the more powerful processing capabilities of the workstation are now available to the office user.

1.2.2.2 Voice Terminals

The telephone was originally a key-pad or rotary-dial hand-set. Recent developments to the telephone incorporate an extended key-pad and liquid crystal display (LCD) to indicate caller information (at least in the office environment). The emergence of ISDN has promoted the development of some enhanced terminals:

- teleconferencing voice
  This service uses the wide-band teleconferencing voice terminal, which includes an optional low-rate data channel.
- group 4 facsimile
  Facsimile is the transmission of a scanned document over the network. While this facility is available for normal subscriber access, the emergence of ISDN has lead to its deployment in many more markets, because of the quality increase resulting from digital end-to-end connection.
- video telephony terminals
  These terminals incorporate telephone voice and slow-scan video in one terminal.

The terminals are all constrained by the unit bandwidth (or multiples thereof) offered by the ISDN.
1.3 Evolution

1.3.1 Networks

1.3.1.1 A Single Network for Voice and Data

The original provision of separate networks for voice and data is now being superseded by evolution towards a single network that can carry both types of traffic. This evolution is driven by the promise of cost reduction in maintaining and operating the network, and by the desire to offer more services.

The recent emergence of ISDN has begun to realise some of the benefits above, but the underlying (mainly circuit-switched) network is still too inflexible to provide the type of full integration necessary for true multi-media terminals.

An integrated network should be packet-switched, for the following reasons:

- *integrated architecture*
  
  Packet switching can provide an integrated customer interface, and give a single network that can carry both voice and data traffic. This leads to reduced operation and maintenance costs.

- *efficient transmission resource usage*
  
  VBR services, such as data, only use the transmission resources part of the time: a circuit-switched connection would be under-used. Packet-switching offers the capability of automatically multiplexing many data sources together to use the bandwidth more efficiently.

1.3.1.3 Private Networks

In the private network area, research has concentrated on transmitting telephone-quality voice across high-speed LANs. It is at best difficult to begin to guarantee the tight delay requirements for voice, and some topologies (such as the token ring) are more suited to carrying voice than others (such as Ethernet).

The different types of LANs currently available are evolving towards higher speed versions that can carry multi-media traffic, such as voice, video, and data. The evolving LANs can be extended to wider areas, and become metropolitan area networks (MANs). The resulting networks are hybrid, however, in that they are basically data networks which have circuit-switched-like characteristics for voice (dual queue data bus (DQDB) and fibre distributed data interface (FDDI)).
1.3.1.4 Larger Private Networks

Data networks in the corporate business area have seen the evolution of frame relay, which offers virtual circuit connections existing for the life of the call. Frame relay is the evolution of the packet networks and the X.25 protocol. The switched multi-megabit data service (SMDS) extends LAN capabilities to the national area. It is a switched service, which represents the evolution of ISDN for the business environment.

The latest technology in the corporate business area is ATM (Asynchronous Transfer Mode). The ATM network is the true marriage of data and voice networks. It can operate at Giga bits per second, and uses small fixed-length packets or cells. The size of the cell is a compromise between that required for efficient data connections and that required for low voice delay.

1.3.1.5 Public Networks

ISDN is presently being deployed in the Public Switched Telephone Network (PSTN), although such networks still include some analogue trunk portions. The PSTN will evolve towards ATM and broad-band ISDN services, once the network can operate at the required bandwidth. This depends on future evolution of the underlying optical transmission medium and optical switching.

1.3.1.6 Intelligent Network Features

PSTNs are beginning to utilise the capability of intelligent network features in the current circuit-switched networks. Using large data bases, intelligent routing of service calls is now possible: using calling line identification, for example, a user that has dialled a standard number for a car break-down service will be automatically routed to the nearest station.

The evolution of intelligent network features will provide universal personal telecommunications (UPT), or 'follow-me'. Calls will be routed to the telephone nearest to the person receiving the call, be it at home, or at work.

1.3.2 Terminal Evolution

1.3.2.1 Voice Communications

The 7kHz audio terminals developed for ISDN will see more rapid deployment as the cost of the signal processing power decreases. The demand in the area of voice services will be for lower bit-rate terminals, since higher bandwidths than are currently provided by 7kHz terminals give little improvement in quality. The emergence of lower
bit-rate coding schemes, such as CELP (Code Excited Linear Prediction) will be favoured when a lower bit-rate means a lower network access cost; as users are charged for the bandwidth they use.

Other voice communications that are likely to see more rapid deployment in the future, are services such as voice recognition, speech recognition, text-to-speech conversion and voice-annotated text.

1.3.2.2 Video Communications

In an evolution similar to that of voice terminals, video will see more rapid deployment as the cost of the processing power decreases. In contrast to voice terminals, there is also likely to be demand for great improvement in quality. High definition television (HDTV) offers a greater clarity of picture, but requires very large bandwidths (20-900 Mbps).

Video on demand, planned to be offered soon by cable television (CATV) operators, will be a market leader in the promotion of broad-band ISDN in the home.

1.3.2.3 Multi-Media Terminals

Personal computers and workstations will be the 'hub' of future multi-media terminals. The terminals will initially be deployed with boards that plug into the PC, especially where slow-scan video is concerned, since this offers lower costs than separate equipment. Other multi-media services will use voice mail as a secondary service, leading to paging, telephone answering services, text mail, and facsimile mail etc.

The reduction in price of the actual components of the multi-media ‘terminal’ will lead to greater use in the office (not only for executives), particularly as the cost of providing facilities decreases, and the quality improves. Multi-media terminals based on LANs will see rapid deployment as the networks evolve towards supporting voice and video with the required level of performance.

1.3.3 Project Unison Research

1.3.3.1 Unison - Research into Integrated Intelligent Networks

The Unison project created an integrated, intelligent network. Based on the Cambridge Fast Ring (CFR) LAN to provide a user (or client) network and a local exchange, the Unison project also developed an ATM-like back-bone network to interconnect sites. This network was truly integrated; VBR and CBR traffic was carried in an identical form by the network. Intelligent network facilities, in the form of
dynamically updated data-bases, greatly facilitated ease of network connection, and allowed an enhanced form of UPT to be developed.

1.3.3.2 Unison - Research into Multi-Media Terminals

The Unison project created a multi-media office. This part of the research investigated the orchestration of applications, and provided actual services to evaluate the behaviour of the network in carrying multi-media traffic. The Unison multi-media office was based around a workstation, but used a LAN to interconnect applications. Consequently, high quality applications that exploited the intelligent network features were developed. These were of higher quality than could have been provided by using plug-in boards in the workstation.
1.4 Aims

1.4.1 Unison Aims

The research described in this thesis was part of the Unison project (Unison 90). The project was part of the Communications and Infrastructure programme of the Alvey project, funded by the DTI. The collaborators in this research were Loughborough University of Technology, Cambridge University, SERC Rutherford and Appleton Laboratories (RAL), Logica plc., Acorn Computers, Cambridge Olivetti Research Laboratory, and Oxford Polytechnic.

The objectives of the Unison project were to provide a network architecture suitable for carrying all types of traffic generated by an integrated office, and to investigate the provision of distributed multi-media services locally in the office and at remote offices. The objectives of the Unison project lead to the provision of an ATM overlay on the back-bone network, in order to provide a suitable wide-area connection facility. Consequently, the applications techniques developed as part of this project are directly applicable to ATM networks.

The applications research was carried out at Loughborough University. RAL and Cambridge University developed the network architecture, and Logica provided the project management.

1.4.2 Thesis Aims

This thesis is concerned with the development of voice terminals as part of a multi-media system, to provide higher-quality speech communications over an integrated, intelligent network. The work is practical, and investigates the incorporation of high-quality voice into experimental integrated offices, interconnected by a prototype integrated, intelligent network. Describing the use of a transputer as an embedded controller, the thesis also investigates how parallel processing techniques can be used to give real-time performance.

The practical work uses the 7 kHz wide-band speech system which was developed for connection to today's ISDN networks. The thesis investigates the incorporation of the codec into the Unison integrated office: a multi-media system that is easy to use, reliable, and one which exploits 'intelligent' network features.

The Unison integrated office is based on a distributed architecture of components, which use a LAN for connection. Although the LAN has traditionally been a network for
data, this thesis shows how a suitable voice protocol can guarantee the correct delay and error properties required for voice connections.

The Unison network uses an ATM-like back-bone network to interconnect remote offices together. This thesis shows how the voice protocol can operate equally well in both a local- and wide-area context. However, a voice protocol cannot 'change' the characteristics of a network that has not been designed to carry all types of multi-media traffic. Consequently, this research demonstrates the suitability of the Unison ATM-like network to carry multi-media traffic, specifically because the developed application is real-time.

The 7 kHz codec is used to provide two- and three-party 'hands-free' teleconferencing in the Unison office. To date, teleconferencing systems have been implemented with voice switching or echo suppression as a means of providing 'hands-free' communication. This thesis adopts a different approach, illustrating how good quality performance can be obtained from low cost audio equipment, and investigating how future teleconferencing systems might provide better audio performance.

The voice application developed uses the transputer as the basis for connection to the LAN, and to provide the bridging function required for three-party teleconferencing. The transputer is not generally favoured for real-time applications, due to its inherent execution non-determinism. This thesis shows how the desired real-time performance can be obtained.

The thesis also highlights how parallel processing leads to modular reusable software. The software is easier to produce, because it enables functional decomposition into very small components, with well-defined interfaces. Modular, well-defined software is easier to maintain, and has a longer deployment life.
1.5 Thesis Structure

Chapter 2 describes the Unison network architecture, to give the reader an understanding of the frame-work within which the wide-band speech system was developed.

Chapter 3 provides background investigation into voice coding schemes, packet voice problems, and audio systems. This discussion provides a context and a foundation for the thesis.

Chapter 4 describes the prototype two-party wide-band speech system that was used to investigate the suitability of the Unison prototype network for carrying voice applications. The speech system was also used to assess the difficulties and investigate possible solutions to the problem of finding a voice protocol that would work equally well in a local and wide-area context.

Chapter 5 describes the development of the Unison two-party wide-band speech system, including a discussion of the audio system, the hardware, the voice protocol, and the parallel processing approach adopted.

Chapter 6 investigates a three-party teleconferencing system structure, and describes the development work necessary before connection to the Unison network was possible. The latter parts of this chapter are intended as suggestions for those who wish to build on this research.

Chapter 7 describes the three-party wide-band speech system. The text discusses the hardware and software of the system, and highlights the reusability and modularity of the two-party system considered in Chapter 5.

Chapter 8 presents the conclusions drawn from this project and discusses future work that builds upon the research.
1.6 Collaboration

This thesis is my own work. The background material was collected and edited by the author, and the work to produce the Unison two- and three-party wide-band speech systems was entirely my own except for the following components:

- The X21 hardware and audio system in the prototype two-party system were developed by M.T. Yin as part of an MSc.

- The software for the prototype Unison two-party system was developed by the author, with some assistance from M.T. Yin.

- The hardware in the final Unison two-party system was initially developed by J. Chapman as part of a BSc final year project; however the hardware was substantially re-engineered for this project, as the original version was not work appropriate.

- The X21 interface testing software was initially developed jointly by the author and D.J. Clarke of RAL.
Chapter 2
The Unison Network Architecture and Integrated Office

2.1 Introduction

This chapter provides the conceptual foundation for this research; describing the Unison network, the 'intelligent' network features, and the other applications developed as part of Project Unison. It also identifies the importance of the wide-band speech system with respect to the Unison network and as an application which forms a constituent part of the integrated office.

The first part of this chapter explores the principles and design considerations behind the integrated office and investigates likely means of connecting and controlling the components.

The second part of the chapter investigates what constitutes an 'ideal' network in the light of the conflicting requirements from multi-media applications. A brief resume of network technologies that are currently available and those likely to be available in the future is also included.

The third part of the chapter describes the Unison network. The architecture evolved from the Universe network (Burren 89) and continued to mature throughout the project. This evolution was due to the emergence of new technology and also as research efforts into intelligent network features began to come to fruition.

The last part of the chapter discusses the importance of wide-band voice as a measure of the suitability of the Unison network for carrying multi-media traffic, together with its relevance to the provision of other multi-media applications.
2.2 The Integrated Office

2.2.1 Advantages of Integrating the Office

The motivation behind integrating the office world is one of increased productivity (Klapman 82).

For an executive, this means:

- *Enhanced communication with others*
- *Decreased interruptions in communication*
- *The provision of more meaningful communication with other executives*
- *Improved personal organisation*

To achieve the design goals set out above, the basic facilities within an office (such as a personal computer, file-server, printer and a telephone) need to be supplemented by the provision of video, better quality speech, and the capability of orchestrating the services in a simple, reliable, and user-friendly manner.

2.2.2 Design Considerations

In designing such an integrated office, considerations in addition to those above also need to be addressed. These are performance, cost-effectiveness, potential growth and reliability.

2.2.2.1 Performance

Current technology can not give us a single machine capable of providing all the services required in an integrated office at a reasonable level of performance. In such situations, the traditional way forward is distributed computing.

The services required in an integrated office have wildly differing characteristics. For example:

- *File Transfer*
  A variable bit-rate (VBR) service needing storage facilities, but not requiring a fixed time response
- *Voice Communications*
  A constant bit-rate (CBR) service, which, while it does not require storage facilities, does necessarily require a small fixed end-to-end delay

Employing distributed computing in the integrated office means that the machine delegated to provide the specific service can be 'tailor made' to the characteristics of the function; a fast, real-time, embedded controller can provide the speech communications.
2.2.2.2 Cost Effectiveness and Potential Growth

An integrated office can be made cost-effective if its architecture is flexible and re-configurable.

This could mean resource sharing with other executives, where individual components are expensive, such as laser printers.

System upgrades should be limited to software changes or the addition of supplementary pieces of equipment.

This is preferable to the complete obsolescence of existing machines.

Inherent in considering this second point is equipment functionality. Specifically, if multi-media devices are kept as simple as possible, they can have an extended service life. Consequently, the approach of distributed computing should be taken further, in order to produce a system whereby call set-up, for example, is not a function of the individual piece of equipment, but is the responsibility of other devices, such as the central workstation.

In some cases, it will become necessary to completely replace certain pieces of equipment. The task of doing so will be greatly eased if the original integration of the now obsolete machine was achieved in accordance with the relative standard.

2.2.2.3 Reliability

 Whilst considerations of cost-effectiveness and expandability dictate a distributed system, the reliability requirements of the integrated office decree otherwise. In particular, if the responsibility for a function inherent in each piece of multi-media equipment, call set-up for example, is always given to the workstation, then if this fails, the office is inoperable until it is replaced.

Most of the intelligence associated with a machine should reside in the central workstation, with at least some of the office equipment being able to act independently, albeit in a basic fashion.

For voice services, this would mean maintaining the dialling facilities offered by the present telephone hand-set.
2.2.3 Distributed Computing - System Interconnection

When considering distributed computing system interconnection, there are two basic architectures:

- tightly-coupled systems
- loosely-coupled systems

2.2.3.1 Tightly-coupled Systems

This classification is aimed primarily at systems that are confined to a very small area (desk-top). The system is connected in a star fashion, and the communication medium could be the workstation’s bus.

The advantage of employing this type of architecture is obviously one of ease of integration.

The disadvantages of using this type of configuration, however, are numerous:

- Lack of expansion
  However many system expansion slots are provided within the workstation, the users will always want more!
- No resource sharing
  The laser printer for example, is expensive and will not be used for most of the time
- Reliability is poor
  All the multi-media devices are connected to the workstation; if this crashes, the system is unusable
• Performance bottlenecks: the workstation's bus

Whatever the processing power of the machine, the data bus would not be able to support simultaneous use of all the multi-media equipment without degradation in performance of individual services, or other processor tasks.

• Performance bottleneck: the network interface

If the network interface is limited in its throughput, then simultaneous use of all the multi-media equipment will produce degradation in the individual services. If the network has a large potential bandwidth, this implies a large amount of bandwidth available on the network. The network, for the most part, will be under used and therefore not cost-effective.

2.2.3.2 Loosely-coupled Systems

This classification is aimed at systems that may cover areas from a few offices to a large geographical region. The means of system inter-connection is a Local Area Network (LAN), or a number of networked LANs (Wide Area Network (WAN)) for larger geographical areas.

![Figure 2: Loosely-coupled system](image)
Whilst system integration is a more difficult task than for a tightly-coupled system, there are many advantages to employing this type of architecture:

- **Expansion is only limited to the capacity of the LAN**
  The capacity of the LAN is usually high in comparison to the traffic that a few applications can generate
- **Resource sharing is guaranteed**
  An executive and personal assistant could share the same laser printer
- **Reliability can be improved**
  Independent operation for each of the multi-media machines can be maintained
- **There are no obvious performance bottlenecks**

### 2.2.4 Intelligent Network Services

#### 2.2.4.1 Directory Service

An up-to-date directory service of all the facilities offered within the office and by the network is an important intelligent network feature. It removes the overhead of maintaining routing information from the office components, and can be dynamically managed to accurately reflect the present services offered.

In a similar way to the manner in which mobile telephone users contact the local station to register their presence within a 'cell', and the information is relayed to their home-base, so this concept can be adjusted to cope with users who wish to move around the network and wish to use an office wherever they may be; simply specifying a user's name and the required service type (<Vicky><Voice>) will contact this person.

#### 2.2.4.2 Stream Replication

An integrated office will probably require some means of stream replication to transmit from a single source to more than one receiver; this is of particular importance when providing teleconferencing facilities. Replication at source could conceivably overtax the processor and network interface and also increases the complexity of the server. The increased complexity would make such a solution very expensive, since the existing server may have to be replaced.

The provision of a service separate to the hosts, which needs to replicate output streams, offers many advantages. These advantages include cost reduction, in that the service can be offered to different users at different times; and network bandwidth savings, where a collection of replicators located at different sites act together to provide wide-area coverage.
2.2.4.3 Stream Synchronisation

The coordination of streams from functionally different sources is likely to be an important requirement in an integrated office. A particular example is the synchronisation of voice and video streams during a teleconference.

A method of synchronising two services might be to insert flags in the voice and video streams, so that the rates of playback can be adjusted to keep in time with one another. This method unfortunately requires co-operation between the two devices in order to insert the flags at the correct intervals, and also some means of co-ordinating playback. Such a mechanism becomes complicated in a distributed environment.

A further option might be to time-stamp the information, but this requires the provision of an accurate network-wide clock facility.

2.2.5 The Unison Integrated Office

The Unison integrated office is a loosely-coupled system, based on relatively simple applications hardware. Office component control is essentially distributed, except where the provision of services retaining the capability of independent operation is pertinent, notably in the wide-band voice system. The components of the office all present a well-defined interface to one another, and to the applications software, with LAN communication following a rigorously defined protocol (Unity RPC, see section 2.6.2.3).
2.2.5.1 Office Components

The components of the Unison integrated office are as follows:

- workstation
- slow-scan video
- still image transfer
- toll quality voice
- wide-band 'hands-free' voice
- laser printer
- scanner

With the exception of the toll-quality speech system (which remained independent), all the above components were controllable from the workstation (Murphy 90).

2.2.5.2 Directory Service and Office Manager

Whilst a local directory service (see section 2.6.4.4) was maintained to facilitate call set-up, the basis of a more advanced global directory service was also implemented (Chapman 89). Negating the need for a potentially very large, static store of domain names and locations at every site, the directory service and the office manager (Murphy 90) also offered the facility of user mobility; communications intended for a particular user could automatically be routed to the domain where the user was logged on to a workstation.

2.2.5.3 Multi-cast Facility

A multi-cast facility (Unison 90) to enable stream replication was also provided within the Unison integrated office. Primarily used for multi-party video (Murphy 90 and Lu 89), it was also controllable from the workstation.
2.3 A Suitable Network

2.3.1 The Ideal Situation

A single network that is capable of carrying all the multi-media traffic well would be ideal. This is desirable from a financial perspective, where a single network does not involve the maintenance and operation expenses incurred by multiple networks. A further, and perhaps more important, consideration is the possibility of true integration within a single network, where an integrated management and operation capability can be offered via the 'intelligent network features' stored in intelligent network databases. These additional facilities can be added to the basic services to allow subscriber control, for example, in multi-media teleconferences.

The types of traffic likely to be presented to such an ‘ideal’ network, can be simply and broadly classified:

- ‘data-like’ (or VBR) services
  Of paramount importance is that the information arrives error-free, but a small end-to-end delay is not a critical requirement

- ‘voice-like’ (or CBR) services
  A small number of errors can be tolerated, but large or wildly varying delay makes the system unusable

‘Data-like’ services are presently carried by packet-switched networks, and ‘voice-like’ services are carried by circuit-switched networks. Packet-switching provides the flexibility of accommodating a wide variety of services, where the information can be transmitted and switched in a unified manner using a unified packet format.

The two types of traffic place conflicting requirements on a packet network:

- Data transmission is best accomplished using large blocks, to minimise the overhead introduced by error control mechanisms
- Voice transmission is best accomplished using small blocks to minimise delay, with no error control mechanisms (re-transmissions are of little use in a packetised voice stream, and merely increase the delay)

An ‘ideal’ network should be based on Local Area Networks (LANs), with interconnection to remote locations being provided by long-haul segments. Such a network must be able to accommodate both types of traffic well, and the expansion of the LAN to a WAN should not alter the characteristics of the network.
2.3.2 The Present Situation

2.3.2.1 Local Area Networks

The LANs available at present all meet the requirements of high bandwidth, low error rates and low end-to-end delay. When considering the suitability of a LAN for carrying real-time traffic, a further parameter, that of jitter, (the maximum possible delay variation from the norm suffered by a packet) is extremely important (Lines 88).

Bus-structured CSMA/CD (Carrier Sense Multiple Access/Collision Detection) LANs, such as Ethernet, are the most popular LANs in the automated office to date. The possibility of using the Ethernet LAN for carrying voice traffic in a multi-media office has consequently been well-researched (Lines 88, Swinehart 87), despite not being seen as the most appropriate solution; the Ethernet LAN can only offer a reasonably controlled jitter performance when operating at 50-80% of full utilisation, with block sizes of 20 ms (Swinehart 84).

Where the maximum number of voice channels is limited, the possibility of increasing the bus-structured LAN’s performance in this area by using a prioritised access scheme for voice and data has also been investigated (Limb 83). While better performance for real-time traffic compared to a pure Ethernet LAN can be expected from such schemes, penalties in terms of increased network interface complexity are accrued.

It is widely accepted that the slotted ring is a far more suitable medium for carrying multi-media traffic, showing a stable throughput curve even at high loading. The Cambridge Ring (CR) and the Cambridge Fast Ring (CFR) (Hopper 86) are empty slotted rings, whose suitability for carrying real-time traffic can be summarised as follows (Ades 86):

Equal Bandwidth sharing among equal requesters

If 'N' requesters simultaneously request maximum bandwidth, they will each receive an equal share of the sum. This is a common property of LANs, but note that the total usable bandwidth does not fall off as the load increases (cf. bus structured CSMA/CD LANs).

Fine Granularity of bandwidth sharing

Bandwidth sharing can be done at the slot level without loss of efficiency, and it is normal for different users to be using widely differing packet lengths. This is not the
case when considering CSMA; if the packet size is small then the available bandwidth drops off significantly (Stallings 84).

**Bandwidth Sharing amongst unequal requesters**

Where 'P' small requesters and 'Q' large users of bandwidth make requests simultaneously, the 'P' user’s needs are satisfied quickly and the 'Q' users then take equal shares of what is left.

The CR and the CFR are well suited for carrying multi-media traffic.

**2.3.2.2 Long-Haul Networks**

If teleconferencing facilities within the integrated office are to be used in conferences on a national or international basis, then some form of long-haul network is required to convert the LAN to a WAN.

At the beginning of the 1980’s, any long-haul segment would have been a circuit-switched satellite connection, as this was then the only means of obtaining Mbps links between sites (Burren 89). Within this circuit-switched channel, a dynamic window allocation scheme was implemented which effectively superimposed a packet-switched structure on the long-haul network. Although the Bit Error Rate (BER) was very good ($10^{-9}$ due to Forward Error Correction (FEC)), the end-to-end delay was very large (260 ms), which meant that voice communication was likely to be poor.

Towards the end of the 1980’s, Narrow-band ISDN became available. Essentially a circuit-switched network with a small amount of packet-switched capacity for data traffic (Turner 86 and Suzuki 88), narrow-band ISDN represented the evolution of common carrier networks. The circuit-switched channel is divided into frames, and within each frame into slots. Consequently, the Synchronous Transfer Mode (STM) service adds no labelling to any traffic presented to it, and each call is identified by its slot position within a frame.

To the user, the above system means that traffic can only be presented to the long-haul networks in units of 64 Kbps or multiples thereof (Turner 86). This is clearly not very cost effective, since a traffic rate of slightly more than 64 Kbps requires the use of two slots. As with the satellite network, the channel can have a dynamic window allocation scheme superimposed on it to achieve the characteristics of a packet-switched network (Unison 90), although the required solution is non-trivial in this case.
Converting the STM service to packet-like service necessarily incurs some amount of time overhead due to the queuing and mapping operations required in such a system. The performance obtained for this system is very good (showing a low packet loss probability) (Siddiqui 89) and the service shows superiority over satellite long-haul networks. The end-to-end delay for this system is approximately 0.5 ms, indicating that the system is far more manageable for voice communications.

The N-ISDN service will evolve into the Broad-band ISDN service. This is expected to be still of an STM nature, initially with bit-rates of 155 Mbps and 600 Mbps and later 1.2, 1.8 and 2.4 Gbps (Decina 90). Further evolution could see a B-ISDN possibly maintaining some of its structure as an STM service, but with significant portions of a 'packet-like' network (Asynchronous Transfer Mode service (ATM)) (Suzuki 88). Instead of slots within frames, traffic will be transferred in units called 'cells'. The protocol used will be of the simplest form possible; flow-control, error detection and recovery etc. will be on an end-to-end basis. As a result, any necessary switching will be extremely fast, since only limited processing on a per-packet basis will be required.

The performance expected from this Broad-band ISDN will be exceptionally good, offering a cell loss probability of less than $10^{-6}$, with a few milliseconds end-to-end delay at worst and a large throughput (initially 45 Mbps) (Decina 90). This network will be eminently suitable for voice transmission (and other real-time services, such as HDTV).
2.4 Unison Network Overall Architecture

The Unison project designed and built a WAN suitable for carrying the types of multi-media traffic produced by integrated offices (Unison 90). Much of the experience used throughout the Unison project was gained from a previous project, Universe (Burren 89), which used satellite connections to provide the inter-site links.

![Unison network architecture](image)

Figure 4: Unison network architecture

The Unison network can be seen to be made up of the following components:

- **Long-haul segments**
  These are used to interconnect sites
- **Exchange LANs**
  These provide site interconnection between separate offices
- **LANs**
  These interconnect the distributed office components
- **Ramps**
  These provide a gateway between the exchange LAN and the long-haul segment
• **Portals**

These provide a gateway between the client LAN and the exchange LAN.

During the Unison project, the network was upgraded as new technology became available. The prototype implementation of the Unison office was based on the CR, and used much of the equipment developed under the Universe project, (such as Z80 small servers). The inter-site connections for this prototype Unison network were provided by the B.T. Megastream service.

The final implementation of the Unison network was based on the CFR, with the inter-site connection being provided by primary-rate ISDN. The ISDN was used to carry ATM-like cells between sites. Transputers and RISC processors were used in this second version to improve the throughput of the network interfaces and increase the processing power available for applications.
2.5 The Unison Network Initial Version

2.5.1 The Cambridge Ring Architecture

The Cambridge Ring (CR) is an empty slotted-ring LAN, which operates at 10 Mbps. Mini-packets are transmitted in slots, the structure of which can be seen in figure 5.

![Cambridge Ring slot structure](image)

Figure 5: Cambridge Ring slot structure

The response bits are used to provide a means of flow control around the CR. Successfully received mini-packets are removed by the source or transmitting station, which marks the mini-packet as empty using the Full/Empty bit.

Of particular importance for voice connections is the fact that a transmitter cannot re-fill a slot it has just emptied; it must wait for another slot to pass by, before it can try to transmit again. This method of operation ensures that the bandwidth is automatically shared out around the users.

The hardware interface to the CR can be seen in figure 6. The repeater regenerates the signals on the ring and provides slot access. The station presents a series of registers to the host.
2.5.2 Protocols

2.5.2.1 Mini-Packet Protocols

The mini-packet protocol used on the CR is performed in hardware and includes processing for a 'back-off' strategy in the event of a 'busy' response from a remote station.

2.5.2.2 Basic Block Protocol

As the amount of user-data within a mini-packet is small (16 bits), data is transmitted around the ring using the Basic-Block Protocol. This protocol provides for block sizes of up to 2048 bytes of data. The format of a basic block can be seen in figure 7. Apart from the data mini-packets, the basic-block includes header, size, port and checksum mini-packets.
2.5.2.3 Protocols Above the Basic Block Protocol

There are three protocols defined above the Basic-Block Protocol:

- **The Datagram Protocol**
  
  This provides for the transmission of a single unacknowledged block across the network. The datagram protocol is not relevant to this thesis and is therefore not discussed further.

- **The Single Shot Protocol**
  
  The single-shot protocol provides for the transmission of two blocks, a request block and a reply, across the network (figure 8). This protocol is normally used for control purposes, such as retrieving a remote address etc. from the name-server.

![Figure 8: Single-shot protocol](image)

- **The Open/Openack Protocol**
  
  The open/openack protocol is used to establish a light-weight virtual circuit across the network. The virtual circuit provided is light-weight because it does not include any means of flow control or error recovery. The formats of the open/openack blocks can be seen in figure 9. The single significant difference between the single-shot and the open/openack protocols is that the latter specifies a port in the open/openack response for further communication. The light-weight virtual circuit remains in existence, until it decays through disuse; there is no termination transaction as such, the circuit times out.

  Flow control and error recovery can be performed by higher protocol layers.

![Figure 9: Open/Openack protocol](image)
2.5.3 Naming, Addressing and Routing in Universe and the Initial Unison Network

Naming in Universe was done on a site basis, with conversion between a name and a global address being performed by machines called name-servers. The name-server, of which there was one per local site or domain, maintained a list of all the local services available, and had knowledge of every other naming domain and the address of the local bridge which provided access to other sites. Call set-up across the Universe network consequently involved interaction between the application and the name-server, which returned an address on the local network for subsequent information transfer; this could be the address of the bridge, for example. The name-server was also responsible for setting up inter-site connections across the network; bridges were instructed as to the mapping operations required to correctly route subsequent packets to their ultimate destination.
2.6 The Unison Network Final Version

2.6.1 The Cambridge Fast Ring

The Cambridge Fast Ring (CFR) is a development of the CR, and was designed to operate at speeds up to 100 Mbps (Hopper 86). The empty slotted-ring architecture can transmit larger mini-packets than the CR; a result of the higher clocking rate.

![CFR Slot Structure Diagram](image)

The actual size of the slots is variable, although within the Unison environment the slot size is fixed. The values chosen are a compromise between providing a few larger slots (attractive for file transfer and VBR services) and providing a large number of small slots (attractive for CBR services, such as voice, and also guaranteeing efficient bandwidth sharing).

The CFR was designed as the basis for a WAN. To this end, the CFR node may be configured as a bridge between two adjacent rings, with a flat address space across the network (Hopper 86).

As can be seen in figure 10, there are no response bits in the mini-packet structure. Instead, the last four bits of the CRC can be used to convey a response.

- **The bits are not corrupted**
  
  Re-transmission is not required, because the transmission was successful

- **The last four bits of the CRC are returned corrupted**
  
  Re-transmission should occur, because this either indicates that the receiver was busy, or that the 12 bits of the CRC implied an error in the mini-packet

Although the response mechanism outlined above is not fool-proof, it is sufficiently reliable, since within the context of a WAN (for which the CFR was initially designed), responses cannot pass through bridging nodes.
The CFR node interface hardware is based on VLSI technology. Like the CR interface hardware, it consists of a repeater and a station, although for the CFR these are individual chips rather than boards.

The repeater is a fast, ECL gate array, whose primary function is to convert the serial data on the ring into bytes, which are then passed to the station chip.

The station is a slower CMOS chip, which accepts parallel data from the repeater chip and presents a set of registers to the host.

One performance benefit of the CFR interface (as compared to the CR interface) is that CRC processing is done in hardware, which represents a major processor time-saving.

2.6.2 Protocols

2.6.2.1 Mini-packet Protocols

As was the case with the CR, the CFR station hardware provides a suitable mini-packet protocol. This protocol is in the form of hardware retries (inter-spaced by suitable periods of delay), in the event of transmission failure. The protocol is programmable to the extent of the number of retries attempted, and the delay between each one. As
mentioned earlier, the protocol does not give a definite indication of an error, and so this becomes the responsibility of higher protocol layers.

Transmission across the CFR between two attached hosts is based on the concept of an association; a station address and port for each host is bound to the virtual connection. This is similar to the light-weight virtual circuit concept developed for the Universe project (Burren 89).

2.6.2.2 Unison Data-Link Protocol

The Unison data-link (UDL) protocol provides a data-link service across an association, and is similar in functionality to the service provided by the Basic-Block Protocol for the CR (Tennenhouse 86). This is in the form of CRC checking and mini-packet assembly/disassembly into larger blocks. Since CRC checking is implemented by the hardware, the first layer implemented in software is block assembly.

The UDL structure includes four bytes of protocol information: the port number and reassembly parameters. This leaves 28 bytes available as user data in each mini-packet.

2.6.2.3 Unity RPC

Most types of multi-media traffic require some sort of flow control on an end-to-end basis, and also due to the multi-hop nature of the network, some means of validating that the blocks are being received at the remote end. The Unity Remote Procedure Call (RPC) provides such a mechanism, and is the basis of distributed computing in the Unison office (McAuley 87).

The service can be subdivided into marshalling and data-transport layers. The marshalling layer is responsible for the semantics of the RPC parameters and the data-transport layer for the actual exchange of information between hosts.

- Marshalling Layer
  The marshalling layer packs and unpacks the RPC parameters into the packet buffers. Because this may need to be done on a byte basis, the marshalling layer can account for a large amount of processing time, especially if byte swapping is involved.

- Data-Transport Layer
  This is the means of validating that the information has been received at the remote end. It requires the use of the UDL service, since information exchange is accomplished in blocks of approximately 1 Kbyte.
Protocol Semantics

There are three different types of interaction defined in Unity: At Most Once, Exactly Once, At Least Once. The Exactly Once interaction is the closest to a local procedure call, but places the biggest demand on the data transport service. Consequently, better performance can be expected from the other types of interaction.

A more complete description of the RPC mechanism can be found in (Hamilton 84).

2.6.3 Naming and Addressing in Unison

Similar to the naming technique used in Universe, the Unison naming rationale was organised on a per-site basis. Within the Unison network, there consequently existed a number of domains (usually a domain was a single client CFR). The function of the secretary was to map a <service> <instance> name to a <station> <port> address, but unlike the system used in Universe, this binding between the symbolic reference to the application host and the actual address of the application host was dynamically managed (Adams 87). The secretary is therefore the only host machine that is at a fixed address. The other machines in the domain can be added/deleted/moved about at will, without the need for directory manipulation. There is one secretary per domain or CFR, and interaction between the various secretaries is necessary to identify remote service names.

The secretaries’ purpose is to create associations between peer hosts; the secretary is responsible not only for mapping the service name and location to an actual address, but also for interacting with other secretaries to provide private port numbers. Each host application must establish and maintain an association with its local secretary to advertise its existence to the ‘world’. If an application fails to maintain this ‘S-association’, then it is automatically removed from the list of services that the secretary ‘knows about’. In order for an application (or client) to establish an association with a peer service, the client sends an association.create request to the secretary, who, if the server exists within the local domain, returns a success message and a <station> <port> address to the client, identifying the newly established association.

2.6.4 Architecture Rationale

The ISDN is a connection-oriented service which presents a fixed bandwidth for use between sites. The system is somewhat inflexible, since the bandwidth can only be allocated in relatively large units, meaning that a service requiring slightly more than one such unit has to either be allocated two units, or must be prepared to suffer some information loss.
Hosts attached to client LANs will use connectionless protocols and so must be isolated from the effects of the long-haul segment.

Traffic from a multi-media office will usually consist of a number of calls of different and varying bandwidth. If the traffic from a multi-media office, presented for transmission between sites, can be multiplexed onto the WAN, then obvious bandwidth savings will result.

The Unison exchange LAN is a parallel version of the CFR, which was chosen for its automatic sharing of bandwidth between attached machines, and provides a fair means of switching traffic (Hopper 86).

2.6.4.1 The ISDN

The ISDN (originally a Megastream link, and later primary-rate ISDN) of bandwidth 2.048 Mbps was available as 30x64 Kbps channels (called B-channels), with one 64 Kbps channel reserved for Ramp signalling (called a D-channel). The remaining bandwidth was used to support synchronisation and maintenance (Tennenhouse 87). Each B-channel can be configured or switched to interconnect a local site to any other remote site. The bandwidth available between two sites is made up of a varying number of B-channels and is called a U-channel. The composite bandwidth of a U-channel may be dynamically varied according to demand. Within the Unison architecture, the structuring of a number of B-channels into a U-channel is the responsibility of the Ramp. This architecture is extendible to include many Ramps operating over primary-rate ISDN links.

2.6.4.2 The Ramp

The D-channel on the ISDN is used to set-up and clear-down calls on the B-channels. Constructing a machine to compose B-channels into a U-channel is difficult, since B-channels may not always follow the same path between sites, which results in different transmission delays through different B-channels. Data presented to the U-channel is spread across the constituent B-channels on a byte basis, whereas data is handled by the LANs on a packet basis, meaning that the data transmitted across the ISDN has to be unscrambled at the receiving ramp (Unison 90).

Once a B-channel call has been set-up, the delay experienced across it is fixed. Consequently, the transmission of marker signals within the B-channels when the U-channel is being set-up or altered in size enables the relative delays to be ascertained and the data then unscrambled. The U-channel bandwidth can be dynamically varied without affecting the flow of data.
The Ramp maintains two queues of packets waiting to be transmitted onto the ISDN: one for priority traffic (real-time delay sensitive traffic such as voice) and one for non-priority traffic (Burren 87).

2.6.4.3 Exchange Operation

The Unison CFR exchange connects the ramps to portals, and also allows connection of other hosts that control the exchange operation (Tennenhouse 87). As would be the case with client LANs connected to an ATM network, communication to other client LANs is dealt with by portals. Communication between portals is facilitated by the use of the RPC mechanism of associations, which is achieved ‘out-of-band’ (McAuley 87).

- Channel Manager
  An important management function of the exchange, the creation and alteration of U-channels, is achieved by the channel manager. The channel manager uses loading information from the ramp and bandwidth allocation requests to dynamically adjust the U-channel.

- Window Manager
  The window manager organises the routing of information between portals on an association basis, binding the associations to suitable U-channels and ramps. The window management function operates independently at each exchange. The values allocated are exchanged between peer window services when the ISDN is set-up, and released when no longer required. Window mapping is performed by the ramp.

  In the context of a very large ATM network, the window management function would enable access to unlimited potential customer sub-nets, although the scope of simultaneous access would be limited.

- Signalling Service
  The signalling service initiates calls across B-channels.

2.6.4.4 Secretary

The Unison secretaries provide the translation of names to addresses and establish associations between clients and servers; a server provides a service and a client uses a service (Adams 87). Secretaries also cooperate with window managers to allocate routes to associations.

Within a local site, there exist a number of secretaries, usually one per client or exchange LAN (domain). The exchange secretary provides the same function as a client LAN secretary, that of mapping names to addresses. The exchange secretary also acts as a central reference point; client LAN secretaries refer to the exchange secretary for connections to other local client LANs and for off-site locations.

The exchange secretary is responsible for creating associations to services that can reside anywhere in the WAN. It achieves this via portals for inter-site communications.
and with the help of peer exchange secretaries for inter-site communication. An exchange secretary therefore needs to maintain two look-up tables: a transient table, which contains the names of services available within its domain, and a static table of domains for use by its clients.

As mentioned previously, each ‘service’ within a domain registers its presence with its local secretary. This ensures that S-associations (secretary to other machine association) are present at the ‘start of day’, enabling call set-up information to be transferred around the network.

For connection to a service in another domain, but on the same site, association.creation requests flow as per figure 12; the secretaries providing domain or service to local address mapping and the portals providing LAN to LAN station port mapping.

![Diagram of call set-up within the same site](image)

**Figure 12: Call set-up within the same site**

For connection to a service in another domain, and at another site, the window manager requests a window to the remote site. Association.creation requests are passed through this window to the remote exchange secretary (figure 13).¹

¹ A more detailed description of the Unison architecture and the exchange functions can be found in Tennenhouse (87) and Unison (90).
Figure 13: Call set-up across the ISDN
2.7 Conclusion

2.7.1 Unison: Research into Multi-Media Communications

The Unison project as an investigation into the multi-media office structure can be seen to have achieved many of the design goals set out at the beginning of the chapter. The distributed nature of the office, its loosely-coupled architecture, and the provision of a directory service and office manager, make the system easy to use, flexible, cost-effective, and reliable.

2.7.2 Unison: Research into Multi-Media Networks

The Unison project as an investigation into multi-media networks is important in the emerging scenario of B-ISDN, since its structure and transport mechanisms are very similar to those envisaged for ATM networks. The provision of intelligent databases providing intelligent network features, such as service registration and routing to the desired service, without relying on the intelligence of the individual applications machines is of particular importance for ATM networks. The applications research encountered many of the problems that future multi-functional terminals operating over ATM networks are likely to face, and such information is therefore invaluable.

2.7.3 The Importance of Voice within Unison

The Unison project was established to investigate multi-media packet network architectures. Of the various types of multi-media traffic presented to the Unison network, the most alien services are real-time ones, since these are traditionally carried by circuit-switched networks that have very different characteristics to a packet-switched architecture. Within the Unison project, the applications that were of a real-time nature were the voice systems (toll and wide-band), and the slow-scan video. Unfortunately, the bandwidth available on an inter-site basis prohibited the transmission of real-time video. Consequently, the only real-time services passed between sites were the voice services.

The voice systems developed under this project were therefore of particular importance in assessing the architecture of the Unison network. In considering the two voice systems developed for this project, the more interesting of the two was the Wide-band system. In this system, the more complex signal processing algorithm imposed stricter performance criteria on the network; the elimination of a greater amount of redundancy in the speech stream means that block loss can have a greater impact on the resulting speech quality, if large segments of information are lost.
An investigation of the voice protocols required for this system will give a valuable insight into those protocols that might be required for other real-time services, such as real-time video.
Chapter 3
Voice Systems

3.1 Introduction

The intention of this project was to provide a hands-free teleconferencing system for use over the Unison network, with as high a quality as possible. This chapter aims to identify the component parts of such a system and describes any associated research.

The first part of this chapter identifies the component parts of the Unison wide band voice system.

The second part of this chapter describes the parameters of voice and shows how they can be exploited either to give improvement in quality or lead to bit-rate reduction.

The third part of this chapter describes the problems associated with voice packetisation and asynchronous transmission across a packet network.

The fourth part of this chapter provides an understanding of the human hearing system in order to facilitate the identification of problems which result from the acoustic environment.
3.2 The Component Parts of a Voice System

In its most basic form, a voice system will collect analogue speech from a person in one acoustical environment and transmit it across the network to another person in another acoustical environment. To transport the collected speech around the network, the sound waves must first be turned into an analogue electrical wave-form (via a microphone), digitised, and finally packetised before being injected into the network transport medium. At the receiving end the opposite process occurs; the packets of speech are disassembled into a continuous digital stream, converted into an analogue electrical wave-form, which is finally fed through a loudspeaker to produce sound waves which the other person can hear.

A block diagram of the proposed system can be seen in figure 14.

Figure 14: Packet Voice System Composition

If we break the system down into its individual components and consider each in isolation, we must be aware that characteristics pertaining to one unit can effect other units. It is also possible to use the parameters of one part of the system to mask unavoidable problems in another part of the system. The result becomes a trade-off between good quality for some of the time and some of the possible inputs, and reasonable quality for all of the time and all of the inputs.

Voice coding schemes are based on the exploitation of voice parameters, to give either bit-rate reduction or improvement in quality. The area of voice coding schemes relevant to this research is wave-form coding. The techniques aim to preserve part of the speech wave form for transmission to the receiver. The text on time-domain coding techniques discusses the family of algorithms from pulse code modulation (PCM) to adaptive differential pulse code modulation (ADPCM), showing how each is an
improvement on the previous mechanism. Frequency-domain coding techniques, such as sub-band coding (SBC) and noise spectral shaping, are intended to be used in conjunction with a time domain coding technique. The sub-band adaptive differential pulse code modulation (SB-ADPCM) codec uses both types of algorithms to achieve significant performance improvements over PCM, while maintaining the same bit-rate.

Voice packetisation and transmission over an asynchronous network leads to two major problems; errors and delay are investigated to show the effect each has on the resulting speech quality. A voice protocol provides the means of transporting a synchronous stream of bytes across an asynchronous network, while minimising the problems of errors and delay. Techniques for reducing the perceptibility of block loss include careful choice of voice packet size, and the use of various block reconstruction methods in the event of block loss.

The audio system and its interaction with the acoustic environment leads to many different problems such as reverberation, feed-back and background noise. The choice of microphone can alleviate, but can also add to some of these problems.
3.3 The SB-ADPCM Codec, Voice Coding Schemes and Bit-Rate Reduction

3.3.1 Introduction

This section describes the component parts of the SB-ADPCM codec used in this research. The section begins by investigating the characteristics of speech signals. The second part of the section shows how the characteristics can be exploited to achieve subjective performance improvements. The third part of the section describes how further performance improvements can be obtained by exploiting the human perception capabilities. The section concludes with a description of the non-standard SB-ADPCM codec.

This section is relevant to the thesis, in that it supports discussions about voice reconstruction techniques (see section 3.4.4.6), and is the basis for the three party wideband speech system investigation (see chapters 6 and 7).

3.3.2 The Voice Signal

The speech wave-form has many properties that can be exploited in efficient coder design (Flanagan 79).

These properties can be classified:

- Frequency Range
- Amplitude Distribution
- Adjacent Sample Correlation
- Power Spectral Density
- Activity

3.3.2.1 Frequency Range

Speech wave-forms are not inherently band-limited (Rabiner 78). To accurately represent all voice signals would require a bandwidth of greater than 20 kHz. The effective frequency range of voice systems is therefore usually dictated by the bandwidth of the transmission system.

3.3.2.2 Amplitude Distribution.

The probability density function (PDF) of speech is characterised by very high probabilities of low and zero amplitudes, a significant probability of very high amplitudes and a monotonically decreasing function of amplitude for in-between amplitudes.
(Flanagan 79). This feature is exploited in the non-uniform quantizers employed in Pulse Code Modulation (PCM) systems.

3.3.2.3 Adjacent Sample Correlation

It is well known that in signals like speech, there is a strong correlation between adjacent samples, and therefore redundancy in the information. Consequently, the formation of an error signal (by subtracting an estimate of the previous sample from the present sample) gives a signal which has a smaller variance than the original signal. The smaller variance of the signal translates either to fewer bits per sample for the same speech quality, or a much improved speech quality for the same number of bits.

3.3.2.4 Power Spectral Density

The long-term average probabilities of different frequency components in speech can be seen in figure 15.

![Diagram](image)

**Figure 15: Long-Term Power Density Spectrum**

(Flanagan 79, Rabiner 78)

The diagram shows that most of the total energy of speech comes from the lower frequency components. Higher frequencies also contribute to articulation (figure 16), and so should be included in the pass-band of the system.
Use of this property is employed in frequency-domain coding techniques such as sub-band coding (see section 3.3.4.2.1), where the frequency range is split into bands and,
according to the contribution to speech quality and articulation, the bands are encoded with more or fewer bits.

Short-term statistics are also of use in coding schemes (figure 17) (Flanagan 79). In particular, voiced segments of speech display local resonances (called formants).

![Figure 17: Short-Term Statistics for Typical Voiced and Un-Voiced Speech Segments (Flanagan 79)](image)

This feature is exploited in the design of a linear predictor (see section 3.3.4.1.3).

3.3.2.5 Activity

It is well known that speech consists of talk-spurts interspersed with silent intervals. Use can be made of this property when silence suppression (as a means of bitrate reduction) is employed (Ades 87).
3.3.3 Performance Criteria

In general, the subjective performance of a wave-form coding speech system is based on two criteria:

- Signal to Quantization Noise (Signal to Noise Ratio, SNR)
- Frequency Pass-Band of the System

For networking applications, the constraint on the system performance is the maximum available bandwidth per call.

3.3.3.1 Signal to Noise Ratio

In a speech system, noise is added by every piece of electrical equipment. By far the most significant contributor to noise in a digital speech system, however, is the quantizer, where an individual sample is rounded up or down to the nearest representable level. So, when digital systems are evaluated for their performance, the terms SNR and quantization noise are taken to mean the same thing: quantization noise.

The SNR value does not contribute to intelligibility above a certain point (Olson 57). Consequently, performance can only be increased further, once the maximum SNR has been achieved, by passing a larger range of frequencies through the system.

3.3.3.2 Frequency Pass-Band of the System

Figure 16 shows the relationship between articulation (intelligibility) and the frequency range of the system. From this diagram, it can be deduced that adequate intelligibility (0.925) is achieved for a frequency range of up to 3.7 kHz, with improvement to 0.975 by increasing the frequency range to 7 kHz. For analogue and PCM systems in use today, the pass-band is 3.5 kHz.

3.3.4 Wave-form Coding Techniques

Reasonable voice quality can only be achieved by using wave-form coding techniques (Flanagan 79). As this dissertation is only concerned with toll (PCM at 16 to 64 Kbps) and broadcast speech qualities (PCM at 64 Kbps and above), lower quality coders (vocoders, a greater understanding of which can be found in Rabiner (78)) are not considered.

Wave-form coders transmit digital information which directly represents the analogue speech wave-form.
This range of coding techniques can be split into two categories:

- Time-domain coding techniques
- Frequency domain coding techniques

Time-domain coding techniques exploit certain characteristics of the speech signal such as amplitude distribution, adjacent sample correlation and activity (see sections 3.3.2.2, 3.3.2.3, 3.3.2.5). Conversely, frequency-domain coding techniques exploit other features of the speech signal, such as the long and short-time power spectral densities (see section 3.3.2.4).

### 3.3.4.1 Time-Domain Coding Techniques

In time-domain coding techniques, the analogue signal is quantized in time and amplitude:

- The continuous input wave-form is sampled at various instances in time
- The continuous amplitude range of the input signal is replaced by a finite set of discrete amplitude levels (quantization)

Sampling and quantization both introduce errors into the speech wave-form. If the input signal is sampled at a frequency ‘2f’, information about frequencies greater than ‘f’ is lost (the Nyquist frequency, Rabiner 78). Only a limited number of amplitude levels can be described by a ‘b’ bit word, which means that the sampled amplitude is rounded up or down to the nearest available level (quantization noise). A fixed number of levels also implies a maximum and a minimum magnitude, which means that signals exceeding the maximum or minimum are effectively clipped (clipping noise). Obviously, the greater the sampling frequency and the greater the number of available amplitude levels, the better the resulting speech quality will be. Unfortunately, the data-rate increases with both of these parameters.

A generalised block diagram of a time-domain coding algorithm can be seen in figure 18. The input speech signal is first band-limited to frequency ‘f’, and then sampled at frequency ‘2f’. The encoding algorithm aims to accurately represent the digitised signal, while minimising the number of bits per sample. The encoder output is then converted to a form suitable for transmission across a network.
The receiving end of the wave-form coder takes the coded samples as input (which may have been corrupted by transmission through the network), and performs the inverse operations of the encoder. The output speech wave-form is obtained by interpolating the samples using a low-pass filter.

### 3.3.4.1.1 Pulse Code Modulation (PCM)

PCM is the most widely used method of digitising speech.

The subjective quality of the output speech depends upon the following parameters:

- the greatest frequency transmitted by the system
- the number of bits per sample used
- the distribution of the levels of the quantizer

The quantizers used in common implementations of the PCM algorithm use time invariant non-uniform (logarithmic) quantization (Rabiner 78). The quantizers have been optimised to suit the probability density function of the input signal: the occurrence of small signal amplitudes is more likely than the occurrence of large amplitudes (see section 3.3.2.2). The quantizers have also been designed to achieve a constant SNR over a large range of signal variances (Rabiner 78). There are two such commonly used schemes: $\mu$-law (European) and $A^*$-law (American).
3.3.4.1.2 Adaptive Pulse Code Modulation (APCM)

In typical voice communications systems, the dynamic range of the speech signal can be as much as 40 dB (Olson 57). With standard PCM systems, the solution to this problem has been one of using logarithmic quantizers. Better results can be obtained if the quantizer is made variable like the input speech (Flanagan 79).

An adaptive quantization scheme uses a quantizer characteristic (uniform or non-uniform) that shrinks or expands in relation to the input speech. Despite the fact that over a long period of time a speech signal can exhibit a large dynamic range, in the short term, the speech power level varies only slowly (Flanagan 79). Consequently, a simple adaptive algorithm can be employed to track the power variations of the input signal (figure 19).

![Figure 19: Block Diagram of an Adaptive Quantization Scheme](image)

Many techniques towards this end exist, a greater understanding of which can be found in Rabiner (78). The range extends from those which estimate the input power from a number of preceding samples, to the simpler one-word memory algorithm, which was improved for use over packet networks by Goodman (Goodman 75). This last technique gives an SNR improvement over law PCM of 4-7 dB.

3.3.4.1.3 Differential Pulse Code Modulation (DPCM)

As mentioned previously, it is well known that in speech signals there is a strong correlation between adjacent samples (see section 3.3.2.3). This means that the formation of an error signal (by subtracting the sum of a few immediately preceding weighted samples from the present sample) gives a signal which has a smaller variance than the
original signal. Consequently, fewer bits per sample are needed to describe the error signal than the original speech wave-form. This translates to a bit-rate reduction or an improvement in the SNR.

The block diagram for a DPCM codec can be seen in figure 20.

![Block Diagram for a DPCM Codec](image)

Figure 20: Block Diagram for a DPCM Codec

The linear predictor in the feedback loop of the encoder produces an estimate of the incoming speech sample, based on previous decoded speech samples. The estimate is subtracted from the incoming speech sample to produce an error sample, which is then quantized. It is important that the quantizer is included inside this predictive closed loop, so that the quantization noise associated with the input to the predictor is the same as that of the error sequence which is transmitted to the decoder. The predictor is matched (in an average sense) to the long-term spectrum of the speech signal (see section 3.3.2.4).

The improvement in SNR of the DPCM system over the PCM case can be evaluated by recognising that the quantization error variance tends to be proportional to the quantizer input variance for a given number of code word bits. Consequently, if the input quantizer variance has been reduced by a gain factor ‘G’, the coding errors have also been reduced by the same factor. This translates to an SNR improvement of ‘G’. Even with the simplest predictor, an SNR improvement of 6 dB can be realised for the DPCM system as opposed to the PCM system (Rabiner 78).
3.3.4.1.4 Adaptive Differential PCM (ADPCM).

Differential coders can be made adaptive in two ways:

- DPCM with adaptive quantization
- DPCM with adaptive prediction

a) DPCM with adaptive quantization

The theory behind adaptive quantization has already been discussed (see section 3.3.4.1.2). The block diagram for the ADPCM codec can be seen in figure 21.

![Block Diagram of an ADPCM Codec](image)

Figure 21: Block Diagram of an ADPCM Codec

The adaption is of the feed-back rather than the feed-forward type. The feed-back scheme is more sensitive to transmission errors, but it does have the advantage of not having to transmit any extra information about the step-size; the step-size is recovered from the transmitted code word.

The SNR improvement to be achieved with this type of configuration is approximately 10-12 dB over a standard law PCM coder; approximately 5 dB (4-7)
increase in SNR by using adaptive quantization and approximately 6 dB increase in SNR by using the differential configuration.

b) DPCM with adaptive prediction

The DPCM coder discussed previously had a fixed predictor, with its coefficients matched to the long-term power spectrum of the speech signal. Thus, a reduction in the performance of this type of coder can be expected when the characteristics of the input speech differ substantially from the long-term average (e.g. a female speaker as opposed to a male speaker).

The predictor coefficients can be made to adapt to the short-time speech power spectrum, giving a better performance than DPCM with a fixed predictor. A greater understanding of adaptive prediction can be gleaned from Rabiner 78.

3.3.4.1.5 Silence Detection Techniques

Silence detection is important in many areas of speech processing, the suppression of echoes, and the economy of channel capacity by ‘time-sharing’.

The problem of locating talk-spurts in the presence of background noise is not trivial, except in very good acoustic environments (Ades 87). In these situations, the energy of the lowest level speech sounds exceeds the background noise level, and a measure of the energy in the speech signal is sufficient to give excellent performance. For most practical situations, an enhancement to the technique, (that of zero-crossing rate measurement), gives a better performance. A fuller understanding of this enhancement can be gained from Rabiner 78.

3.3.4.1.6 Embedded Coding Schemes

The ability to dynamically reduce the number of encoding bits can be useful for voice that is to be transmitted over a packet network. This capability leads to more graceful degradation of the speech quality than if the network were overloaded and packets of voice lost.

The DPCM coder can be adjusted to a lower bit-rate. However, reducing the number of bits in the feed-back loop of both the encoder and decoder parts of the codec is necessary in order to prevent the accumulation of noise in the system (Jayant 84).
3.3.4.2 Frequency-domain Coding Techniques

Time-domain wave-form coding techniques exploit the statistical properties of the speech signal (see section 3.3.4). A further bit-rate reduction (or SNR improvement) can be achieved by taking advantage of human perceptual capabilities.

The following frequency-domain coding techniques are relevant to this research: noise spectral shaping, sub- or split-band coding.

3.3.4.2.1 Noise Spectral Shaping

The power of the noise in a speech signal recovered from a differential codec is proportional to the power of the prediction error (Jayant 84). Also, the noise in the recovered signal is equal to the noise introduced by the quantizer, which has a flat spectrum (Flanagan 79).

Minimisation of the power of the quantization noise does not, however, ensure that the distortion of the voice signal is minimised. Figure 22 shows that the power of the quantization noise can actually exceed the short-term power spectrum of voiced sounds at high frequencies. Consequently, a subjectively better performance from a differential encoder can be achieved by shaping the noise spectrum so that it is no longer flat, but is less than the power of the voiced signal at all frequencies (figure 22).
The noise power has been increased in the lower frequency region where the speech signal contains much of its energy (formants). The high signal power of speech in the low frequencies will mask any increased noise, so that the noise power spectrum can be reduced in the higher frequency regions.

The shaping of the noise spectrum can be achieved by using a pre- and post-filtering arrangement (figure 23).
spectral shaping, it is attractive given its better performance for higher frequencies (Jayant 84).

### 3.3.4.2.1 Sub/Split Band Coding

In this coding method, the frequency band is split into a number of sub-bands (figure 24), by a bank of band-pass filters.

![Figure 24: Frequency Band Split into Sub-Bands](image)

Low-pass translation of each band to zero frequency (by a process similar to that of single side-band modulation) enables the sampling rate of each band to be reduced to its Nyquist rate (Crochiere 76). Each individual band can then be coded using time-domain wave-form coding techniques, according to the perceptual criteria of that particular band.

There are a number of advantages to this method of coding (Flanagan 79):

- **The noise over a particular frequency band is contained in its own band**

In time-domain coding methods, quantization noise is spread over the whole frequency range as white noise. Using sub/split band coding, the noise over a particular frequency band is contained in its own band. This means that possible masking of speech information by quantization noise from another frequency range is avoided.

- **Separate adaptive quantizer step-sizes can be used in each band**

This means that bands with low signal energy have lower step-sizes, and experience less quantization noise than would be the case if the step-size were constant over the whole pass-band.

- **The number of bits allocated to each individual band can be varied according to the perceptual importance of that band**

Consequently, in the lower bands (where pitch and formant information is held), more bits can be allocated to preserve the structure efficiently. This is at the expense of the upper bands which contain noise-like sounds and fricatives and so do not require such high resolution. This technique effectively shapes the quantization noise spectrum on a 'constant level per band' basis.
3.3.4.3 The Non-Standard Embedded SB-ADPCM Codec

The codec used throughout this research is an Embedded SB-ADPCM codec (Lee 84).

It uses most of the various methods of improving quality or bit-rate reduction described in this section:

- Adaptive quantization (see section 3.3.4.1.2)
- Differential coding (see section 3.3.4.1.3)
- Sub-band coding (see section 3.3.4.2.2)
- Noise shaping (see section 3.3.4.2.1)
- Embedded coding (see section 3.3.4.1.6)

Quadrature mirror filters are used to split the speech spectrum (7 kHz) into two bands, a greater understanding of which can be found in Johnston (80). Each band is coded into a number of bits: five bits to the lower band, three bits to the upper band. The coders in both the low and high band operate with the least significant bit deleted in the feed-back loop, to allow a low-rate data channel to be multiplexed with the speech (embedded coding). The bit-rate of the codec is 64 Kbps, which is suitable for use over the basic and primary rate ISDN links currently available in the PSTN. A block diagram of the encoder and decoder parts of the codec can be seen in figure 25.
The non-standard codec algorithm described above has now been replaced by a standard. The G.722 standard codec uses the concept of adaptive prediction as well as the methods described for the non-standard embedded codec, and allocates 6 bits for the low band and 2 for the high band (CCITT G.722).

3.3.5 Conclusion

Wave-form coders exploit the speech signal properties and the human perceptual capabilities to achieve bit-rate reduction or SNR improvement. SNR improvement only
increases quality up to a point, and after that, the only way of improving performance is by extending the bandwidth.

Wave-form coders can be split into two types: time-domain coders and frequency-domain coders.

PCM is a time-domain coder, which uses a fixed quantizer and samples the signal in both time and amplitude. Instead of using a fixed quantizer, as is the case for PCM, making the quantizer variable, like the speech signal, gives an SNR improvement.

A further SNR improvement or bit-rate reduction can be gained from producing an estimate of the input speech signal, and subtracting it from the input. The resulting residual signal has a smaller variance than the input speech, and consequently requires fewer bits to represent the speech with the same SNR.

Both adaptive quantization and differential PCM can be combined in a single coder to achieve SNR improvement or reduce the bit-rate substantially.

Noise spectral shaping is a frequency-domain technique that makes the noise level less than that of speech at all the relevant frequencies. Using this method in conjunction with a wave-form coder actually reduces the SNR, but leads to subjectively better performance. Sub-band coding splits the bandwidth into two parts, which can then be individually coded by a suitable time-domain coding method. This leads to subjective improvements, since both different numbers of bits can be used to code each band according to perceptual criteria, and different step-sizes can be used for the two bands.

Embedded coding is a method of achieving graceful performance degradation for decreases in the bit-rate. This can be used to either transmit a low-rate data channel alongside the speech, or to adapt the bit-rate in a packet network according to loading variations.

The non-standard SB-ADPCM codec used in this research employs all of the above methods of improving the quality of the received speech, whilst achieving the same bit-rate used for PCM in the PSTNs.
3.4 Packetised Voice

3.4.1 Introduction

Voice connections to date have been carried by circuit-switched networks (see section 2.3.2.2). Conveying voice by packet networks will necessarily lead to different problems in managing stream transmission and reception.

The first part of this section identifies the basic functions of a voice protocol for use over a packet network. The parameters important to voice connections (errors and delay) are also described, and the section considers how the parameters have been quantified for circuit-switched networks.

The second part of the section investigates how packet networks can affect voice connections in terms of errors and delay.

The last part of this section investigates voice protocols that have been used for packet networks. These protocols display similar characteristics to those expected for the Unison network.

3.4.2 Basic Functions of a Voice Protocol

A voice protocol designed for use over a packet network must perform certain functions. Voice codecs produce a synchronous stream of bytes, which must be conveyed across the network and played back to the receiving codec in the same synchronous fashion and at the same rate. Therefore, the first function that a voice protocol must perform is to collect the codec samples into a suitable packet to transmit across the network.

At the receiving end, the packet must be disassembled into samples and played back to the codec at the desired rate.

3.4.3 Packetised Voice Problems

General performance criteria for voice connections relate principally to two parameters:

- errors
- delay

The likely variations in these two parameters over a packet network must be taken into account when designing voice protocols for use over such a connection.
The performance expected of a voice connection across a packet network can be gauged by referring to data available for toll quality voice connections across circuit-switched networks.

3.4.3.1 Errors

There are three types of errors relating to circuit-switched networks:

- **Bit error rate (BER)**
- **Burst errors**
- **Speech clipping**

3.4.3.1.1 Bit Error Rate

Bit errors can be caused by many factors, such as interference and transient disturbances. For 56 Kbps PCM, the limit of perception is a BER of $10^{-5}$ (Gruber 83). This figure can be revised for 64 Kbps SB-ADPCM systems to $10^{-4}$ (Gruber 83, Decina 88), since this more complex coding algorithm has been designed to be more robust in the face of transmission errors.

3.4.3.1.2 Burst Errors

Error bursts are transmission errors which occur clustered in time, rather than randomly spread out. Since error bursts are usually short, their occurrence for small periods can be tolerated at a much higher BER within the disturbance than is the case with a constant BER. The limit on perception has been shown to be approximately a BER of $10^{-3}$ provided it lasts less than 0.1% of the time (Gruber 83).

3.4.3.1.3 Speech Clipping

Speech clipping is the loss of speech energy for any time interval (Gruber 83). This type of degradation can be found in systems which use silence detection to reduce the bit rate: speech interpolators and carrier communication systems, due to switching activity and uncontrolled frame slips.
The subjective effect of speech clipping depends on the following factors:

- **speech clipping duration**
- **proportion of active speech that is clipped**
- **frequency of speech clipping occurrences**
- **speech activity**

There are two particular manifestations of speech clipping: front-end clipping and mid-speech clipping.

- Front end clipping is usually associated with interpolators (the limit of perception for high activity speech is \(15 \text{ ms} \leq 0.5\% \text{ of the time}\)) (Gruber 83).
- Mid speech burst clipping is usually associated with packet voice systems (the limit of perception is \(\leq 5 \text{ ms} \leq 1\% \text{ of the time}\)) (Gruber 83).

**3.4.3.1.4 Packetised Voice and Errors**

BERs on LAN-based packet networks are usually very good \((1 \times 10^{-9} \text{ to } 1 \times 10^{-12})\) (Hopper 86). As this figure falls well below the limits of perceptibility for both PCM and SB-ADPCM, then no corrective action is necessary. Packet networks however, suffer from block loss, which produces speech clipping. Block loss in packet networks can principally be found at congested nodes; the nodes are programmed to throw away packets as their loading increases.

**3.4.3.1.5 Error Recommendations**

*The circuit-switched error performance requirements relevant to packet networks are those of front-end and mid-speech clipping. The goal for packet voice systems can be taken to be 5 ms for less than 1% of the time.*

**3.4.3.2 Delay**

The effect of delay on the subjective performance of a voice connection is more important for ‘hands-free’ teleconferencing than it is for standard telephones. This is due to the presence of a closed audio loop in the system, where sounds emanating from the loud-speaker can be picked up by the microphone.

The following information on delay relates to voice connections across circuit-switched networks, where the connection has at least two analogue parts and associated 2/4 wire conversion points. This information gives some idea of how different values of delay might affect the Unison wide-band voice system, since in a ‘hands-free’ teleconferencing situation, echoes are possible.
The effect of delay on a voice connection can be initially split into two main categories:

- absolute delay without echoes
- delay in the presence of echoes

3.4.3.2.1 Absolute Fixed Delay without Echoes

Data relating to actual maximum allowable delay in circuit-switched networks indicates 80 ms for terrestrial connections (Gruber 83). Above this limit, the effect of the propagation time is to cause any expected reply to reach the inquirer later than expected, which may lead to confusion.

3.4.3.2.2 Delay in the Presence of Echoes

The interaction between delay and echoes is very complex (Emling 63). Data available on the relationship is concerned with PSTNs (analogue, mixed and digital), where a number of echoes are possible, due to reflections caused by discontinuities in the network (e.g. electrical coupling in a telephone hand-set, and 2/4 wire hybrids) (figure 26).

Figure 26: Sources of Echo in the Circuit-Switched PSTN Resulting from 2/4 wire Conversion Points
The relationship between echoes and delay in an analogue or mixed network is quantified in terms of three parameters:

- *talker echo*
- *side-tone*
- *listener echo*

**a) Talker Echo**

Talker echo is the result of an acoustic signal originating from the mouth of the speaker being reflected back by one major discontinuity in the connection. The delays relevant to talker echo are in the region of greater than 35 ms and upwards, which result in a definite recognisable echo after the original sound (Gruber 83). Talker echo is characterised primarily by its amplitude and delay relative to the original sound, and echo suppressers or cancellors are generally recommended.

**b) Side-tone**

Electrical side-tone is the result of an acoustic signal originating from the speaker’s mouth being returned through the voice terminal to the speaker’s ear (Gruber 83). Side-tone is characterised primarily by its amplitude relevant to the original sound, where the delay expected approaches zero.

Data relating to side tone indicates that this situation gives maximum exposure to the ambient acoustic noise impairment. This effect has a direct influence on cases where ambient noise levels are different at each end of a call. A person with a low side-tone path loss may reduce the volume of their speech, which may have an adverse effect on the listener in a high acoustic noise environment (Richards 73).

The effect of side-tone is not necessarily harmful, because it gives the impression that the circuit is ‘alive’ (Emling 63).

**c) Listener Echo**

Listener echo results from an acoustic signal originating from the speaker’s mouth being received more than once at the listener’s ear (Gruber 83). This is due to two principle reflections in the connection. Listener echo is characterised primarily by the delay and loss of the listener echo path. Since this echo path readily suffers from positive feedback, and consequently instability, the listener echo path loss is often specified by the singing margin. The singing margin is the minimum loss of the listener echo path at a single frequency in the voice band. It must be greater than 0 dB to give stability, and much greater than 0 dB to give good performance. A singing margin of more than 6 dB is
required for delays less than 5 ms and more than 10 dB is required for delays greater than 10 ms (Gruber 83).

3.4.3.2.3 Delay in Packet Voice Systems

The delay experienced across packet networks where contention access methods are used can be split into a number of components:

- packetisation
- transmission delay
- statistical queuing jitter

A further component of delay, possible in some packet networks:

- clock drift

a) Packetisation

When samples are collected into packets at the transmitter, a delay equal to the length of time taken to collect the samples is automatically inserted into the end-to-end delay.

b) Transmission Delay

The transmission delay for terrestrial connections is the length of time taken to transmit the packet along the wire or optical fibre, and is usually very short (being proportional to the inverse of the speed of light).

c) Statistical Queuing Jitter

Statistical queuing jitter is a direct result of the access methods used at various parts of the network: queuing to get onto the LAN in the first place and queuing to get onto different networks at nodes. The resulting delay inserted into the end-to-end delay varies from packet to packet.

d) Clock Drift

The clocks at either end of a voice connection, even though they may be highly accurate crystals, will never be identically equal in frequency. This means that one end of a connection will gradually run out of blocks to play back, while the other end receives packets at a faster rate than it can play them back. Delay on an end-to-end basis therefore slowly increases as the call proceeds, until one buffer is full and the other empty.
3.4.3.3 Delay Recommendations

Echo will not happen in the wide-band speech system because of any reflections in the transmission path, but rather will occur because of the 'hands-free' method of operation. The perception of echoes in the wide-band speech system depends upon the delay experienced across the network.

The allowable round-trip delay in the absence of echoes should be less than 80 ms. In the presence of echoes, the round-trip delay should be less than 35 ms to avoid a definite echo after the original sound (talker echo). Delays less than 35 ms, where feedback is possible in the system, are likely to suffer from singing, where a margin of more than 10 dB is required for delays greater than 10 ms (listener echo). While a small level of echo for low delays is required to give the listener comfort, the situation gives maximum exposure to ambient noise impairment.

*For voice connections, the delay experienced across the network should be kept as low as possible. Singing margin improvement is likely to be required.*

*The delay experienced across packet networks can be minimised by reducing the packetisation delay (reduce the block size), reducing the jitter (priority access for voice packets at gateways) and by reducing the effect of clock drift (occasional timing adjustment).*

3.4.4 Voice Protocol Design

For systems operating in environments with good BERs, delay consideration and block loss become the most important factors, since voice streams are very robust against transmission errors.

3.4.4.1 Light Weight Protocols

A voice protocol should be as light weight as possible as far as errors are concerned (Ades 87). There is no point in re-transmitting corrupted or lost blocks, since unless the end-to-end delay is increased sufficiently they will arrive too late to be played back. Within a light weight voice protocol, there is also no need for any flow control mechanisms; a node that cannot source or sink voice packets at the prescribed rate cannot communicate with other speech nodes.

*Packet voice connections do not require any flow-control mechanisms.*
### 3.4.4.2 Block Sequencing

Even though there is no need to re-transmit lost blocks, it is still important to know when there has been loss and how much; when a segment has been thrown away during transmission, the receiving end experiences a gap in its reception. Due to statistical queuing jitter however, it is impossible to tell whether the gap is the result of an abnormally long delayed packet or a lost one. Blocks that are missing need to be substituted with a suitable fill-in, whereas slightly delayed blocks do not. Excessively delayed blocks are useless, and are therefore also replaced; a fill-in packet has already been played back before the true packet arrives.

In order to detect the presence of lost blocks, each block is tagged with a number at the transmitting end (block sequencing). Block sequencing is also useful when silence detection/suppression is being used as a means of reducing the bit-rate across the network.

*Block sequencing is always required in packet voice systems, in order to indicate when blocks have been lost.*

### 3.4.4.3 Packet Buffering

For packet networks, buffering is necessary at both ends of a voice link. At the transmitter, packets may have to queue to get onto the network. At the receiver, an output buffer provides the means of smoothing out the statistical queuing jitter experienced by each packet transmitted across the network, so that samples can be delivered to the codec synchronously and at the desired rate (Ades 87).

### 3.4.4.4 Receiver Buffer Timing

There are two types of receiver buffer timing:

- *strict timing*
- *delayed timing*

#### 3.4.4.4.1 Strict Timing

In networks where the delay dispersion is small, a fixed value should be inserted in the receive buffer before 'play-back' begins (Ades 87). The value taken will normally be 95% of the delay dispersion, since some will always be lost. This ensures that most of the packets received will be in time to be played back to the listener.
Unfortunately, delay characteristics are often not available before the voice protocol is designed and, in any case, the delay characteristics will always change with time as the network loading increases.

3.4.4.4.2 Delayed Timing

If delays across a packet network are expected to be large or are expected to vary wildly, (as is the case if part of the network was being made up of satellite links), then an adaptive algorithm could be employed (Ades 87). A suitable adaptive algorithm would monitor the number of packets that arrived outside of the present reconstruction delay and, if this value increased, would infer that the network loading (and therefore average delay) had increased. In such a situation, moves to retard the timing (by increasing the reconstruction delay) could be made. A delayed timing algorithm would also have to be able to reduce the delay should network loading ease.

*If the delay variations are expected to be small, then the simpler strict timing mechanism can be used. If delay is expected to vary significantly over the duration of a call, then a more complicated delayed timing algorithm should be used.*

3.4.4.5 Block Length

The choice of a suitable packet length is complex and involves compromises. In order to minimise the packetisation time at the transmitter and to limit the perceptual effect of lost blocks, the block length should be as short as possible. Some networks offer the capability of varying the block size according to the nature of the traffic being carried (Hopper 86), but where the packet size is fixed, this could be achieved by only partially filling a packet with speech samples (Ades 87). The resulting disadvantage is a reduction in the network bandwidth utilisation; the proportion of overhead bits to speech samples is increased.

A further consideration in packet networks is that for a given codec data rate, an increase in the number of packets transmitted across the network results, as the block size is reduced. An increase in the throughput (packets per second) across the network may begin to congest network nodes. Since congested nodes throw away packets, this will have a detrimental effect on the speech codec, forcing the use of longer block lengths.

Related research indicates that perceptually, since the number of lost blocks is inversely proportional to the packet length, the amount of voice samples in a packet should be in the range 16-32 ms worth. This represents a trade-off between the number of speech losses per second and the probability of losing a phoneme (Jayant 81).
Perceptual performance for PCM over packet networks is summarised, where the results are based on lost blocks being replaced by all zeroes, as follows:

<table>
<thead>
<tr>
<th>Packet Length</th>
<th>Nature of Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td>less than 4 ms</td>
<td>Crackles</td>
</tr>
<tr>
<td>16 to 32 ms</td>
<td>Glitches</td>
</tr>
<tr>
<td>greater than 64 ms</td>
<td>Phoneme Losses</td>
</tr>
</tbody>
</table>

Table 1: Perceptual performance degradation for various packet lengths

To minimise the perception of block loss, the packet length should be kept in the range of 16-32 ms worth of speech. If lower block sizes can be used without increasing the loading on the network, then they should be adopted. Block loss of less than 4 ms worth of speech represents the limit of perception for PCM coded speech.

3.4.4.6 Block Reconstruction Techniques

Block loss in SB-ADPCM (see section 3.3.4.3) is potentially more detrimental than for PCM, despite having been designed to be more robust in the face of transmission errors. This is because prediction residual loss not only produces gaps in the output speech, but also disrupts the decoder tracking of subsequent speech for a time interval afterwards (Choi 89).

The distortion introduced depends on a number of factors:

- The position of the lost block in the decoded speech: during silence, unvoiced speech, voiced speech. Examples of voiced and un-voiced speech can be found in figure 17. A greater understanding of the difference can be found in Rabiner 78.
- The length of the lost block (table 1)
- The reconstruction technique employed. Most distortion results during voiced sounds, and depends substantially upon the block length and the number of consecutive blocks lost.

Reconstruction techniques assume that the speech properties do not change drastically during one packet (Choi 89). The techniques will therefore fail if a large number of blocks have been lost consecutively, or the lost packet(s) corresponded to a transition between a voiced/unvoiced region of speech.

There are three relatively simple methods of reconstruction available:

- Zero substitution
  This is the most obvious solution to the problem of block loss. A missing packet is replaced by one containing all zeroes and then played back to the decoder.
• **Packet repetition**  
  In this method, a lost block is replaced by a copy of the packet preceding the missing one, and is then fed to the decoder. This method gives good performance for ADPCM, since it usually fills the missing segment with a packet of comparable properties to the original (Choi 89).

• **Wave form substitution**  
  The missing speech segment is replaced at the decoder output by the previous interval of decoded speech, and the adaptation logic is frozen for the duration of the packet. This method gives similar results to packet repetition for SB-ADPCM (Choi 89).

More complicated reconstruction techniques can be found in the following reference sources:

• **Lockhart 88**  
  Pitch replication based on the decoded output of PCM.

• **Valenzuela 89**  
  Phase matching at the decoder output; using a pitch estimate to reconstruct the properties of the lost segment.

• **Jayant 81**  
  Odd-even sample interpolation procedure for DPCM coded samples and ADPCM samples, where the step-size is transmitted separately as part of the block header. The samples are also re-arranged so that all the odd samples from two packets are in the first packet and all the even ones are in the next.

• **Yuito 89**  
  Sample Interpolation for PCM coded samples, using a pattern matching technique.

The above techniques all have the disadvantage of increasing the end-to-end delay of a voice connection by at least one packet, (where one packet is usually taken to be 16ms worth of speech), and substantially increasing the processing power required.

Other approaches to the problem of block loss involve gradually reducing the bitrate of embedded coding schemes, marking the importance of a particular packet to resulting speech quality, and arranging for congested nodes to throw away the least important ones (silence detection !) (Lockhart 88, Kitawaki 90).

*If delay across the voice connection is to be minimised, then voice reconstruction methods of packet repetition or wave-form substitution are recommended. These methods both give good results for ADPCM.*

### 3.4.5 Conclusion

Voice connection problems relate to two areas: errors and delay. The only error consideration for packetised voice systems is block loss (front-end or mid-speech burst
clipping). The delay experienced by a voice connection should be kept as low as possible to minimise the perception of echoes.

In packet voice systems, the delay can be minimised in the following way:

- by reducing the packetisation delay - reduce the block size
- by reducing the jitter - implement priority access for voice packets at gate-ways
- by reducing the effect of clock drift - occasional timing adjustment

Voice protocols should be kept as light-weight as possible; no flow-control mechanism is required. Block sequencing is always required for packet voice connections, to indicate when blocks have been lost. If the delay variations in the packet network are expected to be small, then a strict timing algorithm can be used for voice reconstruction, otherwise a delayed timing algorithm should be used.

To minimise the perception of block loss in a packetised voice system, the block size should be kept as low as possible, without risking network overload. Where delay minimisation is a priority, block loss should be repaired by using packet repetition or wave form substitution mechanisms. Both these voice reconstruction mechanisms give good performance results for ADPCM.
3.5 Acoustics and Audio Systems

This section provides the background material necessary to identify the problems associated with a 'hands-free' teleconferencing system. Careful consideration of acoustics and audio system characteristics is important, as voice systems are assessed subjectively.

The first part of the section deals with the performance of the hearing system. It has been split into two sub-sections: transmission system effects and environment acoustic effects.

The second part of this section describes some basic problems associated with microphones.

3.5.1 The Performance of The Hearing System

The performance of a system depends to a great extent on the performance of the human ear. The ear is not equally sensitive to sounds which are uniformly related when measured by objective means. Consequently, it is important to assess the characteristic response of the ear when designing audio systems.

3.5.1.1 Transmission System Effects

The following describes effects which are due to the transmission system:

- perception of intensity and frequency
- non-linear distortion

3.5.1.1.1 Perception of Intensity and Frequency

Intensity and frequency are objective measurements of sound, which subjectively translate into quantities of loudness and pitch (Olson 57). From the graph (figure 27) (Olson 57), it can be seen that loudness depends on both intensity and frequency.
The main observations to be made from this graph are that the equal loudness contours decrease as frequency increases, flatten out for mid-frequencies, and rise again as frequency is further increased. This flattening of the curves has consequences for sound reproducing systems. If a sound is reproduced at a lower volume than that at which it was recorded, then the sound will appear to lack bass and vice versa.

3.5.1.1.2 Non-Linear Distortion

Non-linear distortion occurs when there is a deviation from linearity in the input/output characteristics of the system (Olson 57). The result of non-linearity is the production of harmonics as well as sum and difference tones from the input frequencies (Nisbett 93). With speech, this means that a lot of energy is thrown into the higher frequencies, so the speech sounds very harsh and distorted.\(^2\)

Non-linear distortion can be classified into many different types. For the purposes of this thesis however, it will be divided into two general sections, which provide enough detail for later chapters: amplitude distortion and frequency distortion.

a) Amplitude Distortion

Amplitude distortion is a change in the gain of the system for different input signal amplitudes. The most common occurrence of this is amplifier or loudspeaker

\(^2\) The natural voiced speech spectrum has most of the energy in the lower frequencies, where the formants are located (see section 3.3.2.4); the higher frequencies do not contain much energy and so are easily masked by these spurious components.
overloading, which results in the wave-form being clipped. Overloading most often occurs for high frequency transients, sibilants (such as the letter 's'), not having a harmonic structure, tend to bring out any distortion. The distortion of transients has a very marked effect on the subjective quality of speech, which makes this a major design issue if good quality voice is to be produced.

b) Frequency Distortion

Frequency distortion is the variation of the gain of the system with the frequency of the input signal. Although the human ear is fairly tolerant to minor irregularities in the frequency response of the system, a generally flat frequency response curve is still an important design goal if transient distortion is to be minimised.

Noise colouration (see section 3.5.1.2) has a far greater effect on the frequency spectrum than any distortion introduced by the transmission system, so any small variations in the frequency response of the transmission system are swamped by the frequency response of the speaker's environment.

Differences in the recording and reproduction levels may alter the subjective quality of speech.

Amplitude distortion has a very marked effect on speech quality and should be avoided. The effects of frequency distortion can be ignored, since perception will be masked by the frequency response of the speaker's environment.

3.5.1.2 Environment Acoustic Effects

The following describes the effect on audio system performance of environmental acoustics. The topics which are considered here are reverberation, noise colouration, room resonances, and background noise.

3.5.1.2.1 Reverberation

The sound waves emitted by a speaker in a room are reflected many times from the walls, ceiling and floor. At each point of reflection, some of the energy of the sound is absorbed and some reflected. The perceived sound at any one instant is a combination of the original sound source and reflected, delayed versions of previous instantaneous sounds. Consequently, when the sound starts, the energy does not build up instantaneously, but rather takes a finite time to build up to its maximum. Similarly, when the sound stops, the sound energy in the room takes a finite time to die away. The
reverberation time of a room is defined as the time for a sound to decrease to one millionth of its original intensity after the sound source has stopped (Nisbett 93).

The reverberation time varies from room to room and depends not only on the dimensions of the individual room, but on the reflectivity of the surfaces of the room; large halls have a long reverberation time, and a room with highly sound-reflecting walls will have a further increased reverberation time.

Reverberation time in a room actually increases articulation up to a certain optimum value, after which longer reverberation times can seriously degrade intelligibility (figure 28).

![Figure 28: The Articulation of a Speaker in Auditoriums of Various Volumes (Olson 57)](image)

The graphs show that for weak speakers (in fairly large rooms), optimum articulation is achieved for reverberation times of approximately one second, with a lower optimum value for amplified speech.

While some reverberation is necessary for optimum articulation, large amounts can seriously degrade speech quality.

3.5.1.2.2 Noise Colouration and Room Resonances

Sound colouration occurs when certain frequencies in the reverberation content of a perceived sound are emphasised.

This is due to two factors:
- efficient absorption of some frequencies during reflection, and the efficient reflection of other frequencies
- the setting up of standing waves or room resonances

Chapter 3 Voice Systems
a) Absorption and Reflection

The material which a surface is made out of shows selective absorption of speech sounds (Eargle 86). A heavy carpet on a wooden floor absorbs more than thirty times as much energy at high frequencies as it absorbs at low frequencies. A wood floor absorbs more than twice the sound energy at low frequencies than it absorbs at high frequencies.

Reasonable acoustic provisions in the speaker's environment, such as carpeting the floor, improve the subjective quality of speech.

b) Room Resonances

The dimensions of the room give rise to standing waves between parallel walls. The basic resonant frequency corresponds to that of a sound wave with a half wavelength extending from one reflecting wall to the opposite wall. The basic resonant frequency is accompanied by an array of harmonic resonant frequencies (a room's normal modes) (figure 29).

![Figure 29: Illustration of Normal Modes in Response of a Room 5.2x6.4x2.7 metres (Eargle 86)](image)

The intensity of the resonance is determined by the reflectivity of the wall surfaces.

Acoustic resonance effectively increases the audibility of the resonant frequency and its harmonics, and decreases the audibility of non-resonant frequencies.
3.5.1.2.3 Background Noise

The presence of background noise can greatly reduce articulation. Apart from being distracting to the listener, its presence has a masking effect on the speech, reducing the optimum gain range of the transmission system (figure 30).

![Figure 30: The Syllable Articulation of a Sound Reproducing System as a Function of the Reproduced Sound Level for Two Conditions of Operation](image)

1. No Noise 2. With Residence Room Noise of 43 decibels (Olson 57).

*The subjective quality of the reproduced speech can be improved by minimising the background noise level.*

3.5.2 Audio Equipment

All sound reproducing systems employ the same basic audio equipment, with additional pieces being used depending on the purpose of the individual system. The basic set of audio equipment includes one or more microphones and microphone preamplifiers, a power amplifier, and an arrangement of one or more loudspeakers. For ‘hands-free’ teleconferencing, the most important part of the system is the microphone, since this device is the one for which a change of type has the greatest effect on the resulting quality.

3.5.2.1 Microphone Parameters

For normal sound levels, a microphone should produce an electrical signal which is well above its own electrical noise level. The resulting signal must be substantially undistorted in both amplitude and frequency (see section 3.5.1.1.2). The field of capture, robustness to interference, distortionless sound pick-up, and unobtrusiveness are important factors when choosing a suitable microphone.
3.5.2.1.1 Directional Response

An omni-directional microphone can pick up sounds arriving at all angles; it measures sound pressure. A directional microphone has a small range of angles that it can pick-up sound in, and will reject sound waves arriving outside of its field of capture; it measures air pressure gradient (Nisbett 93).

3.5.2.1.2 Frequency Response in Practice

There are a number of distortions that occur when using microphones: High-frequency effects, low-frequency effects, and interference.

a) High-frequency Effects

The pick-up pattern of a microphone deteriorates at high frequencies; the pattern tends to sharpen. This can be due to many factors, such as the shielding effect of the microphone body, reflection within the diaphragm casing, the formation of standing waves in front of the diaphragm etc.

b) Low-frequency Effects

When used at close range, all directional microphones suffer from the proximity effect. This produces a rise in the frequency response. Some microphones are designed to have a frequency response which dips at low frequencies, so that the close hand-held application is relatively flat (Borwick 87).

c) Interference Effects

When sound reaches a microphone by way of two major paths (directly and from a first order reflection off a desk top, for example), the waves interfere to produce a series of peaks and dips in the response (Eargle 86).

The position of the microphone relative to the speaker can greatly improve the subjective quality of speech.

3.5.3 Conclusion

The performance of an audio system depends to a great extent on the response of the ear.
Transmission system effects can alter the subjective quality of the speech. The most important effect is amplitude distortion, which should be avoided. Differences in the recording and reproduction levels may also alter perception.

Environment acoustic effects can have a more damaging effect on the subjective quality of speech. Whilst some reverberation is necessary, large amounts can seriously degrade articulation. Reasonable acoustic provisions in a room, such as carpeting the floor and minimising background noise, improve the subjective quality of the reproduced speech.

The choice of microphone is an important consideration, since it is the first element in the system. The position of the microphone can greatly improve the subjective quality of the received speech.
Chapter 4
The Prototype Unison Two-party Wide-band Speech System

4.1 Introduction

This chapter describes the work performed on the prototype Unison two-party wide-band speech system. It is based on a toll quality two-party system that was developed for the Universe Project (Burren 89).

The aim of this work is to adapt an existing system used in the Universe Project to operate over a prototype Unison network. This chapter gives a preliminary investigation into the behaviour of a two-party wide-band speech system over an integrated services network.

The topics of research are as follows:
• a hands-free teleconferencing facility
• how SB-ADPCM behaves across a packet network in comparison to PCM
• the effect of delay on the system
• the effect of block loss
• the behaviour of the voice protocol designed for the Universe network

The first part of the chapter briefly looks at the Universe network architecture and the voice protocol developed.

The second part of the chapter describes the modifications made to the Universe hardware and software in order to investigate the operation of the system over the prototype Unison network.

The concluding part of the chapter describes the performance of the system, and assesses how the problems might be alleviated in future work.
4.2 The Universe Network and Toll Quality Voice System

4.2.1 Introduction

The Universe Toll quality speech system was jointly developed by Stephen Ades (Cambridge University) (Ades 87), and C.J. Adams (Rutherford and Appleton Research Laboratories (RAL)). A brief summary of the Universe network architecture and voice system hardware and software is included. This is to give the reader some insight into the likely behavioural differences between the Universe system and the Unison prototype two-party system.

4.2.2 The Universe Network

The Universe network consisted of a number of LANs (Cambridge Rings (CR)) connected on a per-site basis by bridges, and on an inter-site basis by satellite connections (Burren 89). As discussed in section 3.4.3.3, the most important consideration for voice systems is delay. For the Universe network, the propagation delay was very large (260 ms) due to the satellite connection (Burren 89). Since satellite access was shared between many types of users on a contention basis, the delay was also subject to frequent and statistically large variations in delay (jitter, see section 3.4.3.4.3). The satellite bridges, on becoming congested, began to throw away packets.

4.2.3 The Universe Voice Protocol

Given the wide variations in delay on the Universe network, a delayed timing mechanism (see section 3.4.4.4.2) was employed to try to 'iron-out' the delay fluctuations. Timing adjustment, where alteration is likely to be significant (as was the case for the Universe voice system), is most imperceptibly achieved by the contraction and expansion of 'silent' intervals (Ades 87). The Universe voice system incorporated silence detection (see section 3.3.4.1.6) both to facilitate imperceptible timing adjustment, and to save bandwidth (since satellite bandwidth was expensive and limited).

4.2.3.1 Universe Voice System Hardware

The Universe voice system can be seen in figure 31.
The standard CR node consists of station and repeater circuits, which are connected to the interface processor (Z-80) as peripherals. The hardware contains a 64 kbps A-law PCM codec (see section 3.3.4.1.1). The hardware also contains logic for dialling pulses and silence detection. The Z-80 module contains a timer (used to interrupt the Z-80 every 125s), so that speech samples can be read from the codec logic.

4.2.3.2 Universe Voice System Software Structure

The software was written in Z-80 assembly language, and used 80-100% of the processor power available (Ades 87), depending on the block size used. The size can be varied between 16 and 2048 ms worth of speech per block.

The code can be split into three basic sections:

- **CR network interface**
- **voice sample collection and play-back**
- **system set-up and 'in-call' control**

4.2.3.2.1 CR Network Interface

This section of the codec handles the transmission and reception of Basic Blocks (see section 2.5.2.2). Since the basic block protocol is quite complex, the control of this interface is consequently very processor intensive. The reception of each mini-packet is communicated to the Z-80 via an interrupt.

4.2.3.2.2 Voice Sample Collection and Play-back

This part of the code uses the timer module to collect samples, and to play received ones back to the codec logic.
4.2.3.2.3 System Set-Up and ‘In-Call’ Control

The system is controlled by a state machine (figure 32).

To contact a remote telephone, a number (dialled on the telephone key pad 
<service type> <site> <user number>) is sent across the network to the local name-server. 
The name-server returns a physical station address, and a function number. The function 
number reflects the characteristics of the proposed route across the network; it is an initial
'jitter' estimate.

As discussed earlier, the delayed timing algorithm (see section 3.4.4.4.2) is used
to control speech ‘playback’, with timing adjustments being made during periods of
silence.
Figure 32: The Universe Toll Quality Voice State Machine
4.2.4 Conclusion

This section has given a brief overview of the Universe toll quality voice system.

In order to cope with the large variations in delay across the satellite-based Universe network, a delayed timing algorithm was used. This algorithm modified periods of silence to minimise the perception of any adjustments.

The hardware was based on the Z-80 processor, with a timer module to interrupt the processor every 125 $\mu$s. The timer module was used to pass samples to and from the PCM codec logic.

Communication to the CR used the Basic Block protocol. The block size was selectable in software from 16 to 2048 ms.

An initial jitter estimate was provided by the name-server on call start. This was used to retard the timing in the buffer before speech 'play-back' began. Subsequent call control was supervised by a state machine.
4.3 Modifications made for the Unison Prototype

4.3.1 Introduction

The prototype Unison network (see section 2.5), was based largely on the Universe network (Burren 89), with the exception that the satellite connection was replaced by B.T. Megastream links. This had a large effect on the likely performance of a voice system, since the delay and delay fluctuations expected in the Unison prototype network were significantly less than for the satellite links used in the Universe network.

The Unison prototype system used the non-standard SB-ADPCM codecs used throughout this research (Lee 84, CCITT SG XV111), which code 7 kHz of speech into 64 Kbps (the same data rate as that of the toll quality voice system).

4.3.2 Equipment Modifications

4.3.2.1 Hardware Modifications

The hardware modifications were performed as part of an MSc project by Thwe Yin (Yin 86). The layout can be seen in figure 33.

![Figure 33: Unison Prototype Wide-band Voice System](image)

The equipment is nearly the same as that used for the Universe network toll voice system. The toll quality codec board has been replaced by an X21 interface (CCITT X.21), which converts the Z-80 data bus to the X21 serial interface for the codecs.
4.3.2.2 Audio System

The audio system used for the Unison prototype system was also developed by M.T. Yin (Yin 86), and consisted of a directional microphone (see section 5.2.4.1.1), a pair of loud speakers and, an ordinary home power amplifier. A pair of transformers was also included to match the power levels of the codec input and the microphone output.

4.3.2.3 Equipment Difference Considerations

The wide-band system, as designed by M.T. Yin, had some key differences to the toll quality voice system developed under the Universe project:

- no means of dialling a 'number'
- no silence detection circuitry

4.3.3 Software Modifications

The software modifications were undertaken jointly by the author and M.T. Yin. The modifications were rather crude, but were sufficient to get the system working.

4.3.3.1 No Dialling facilities

This problem was overcome simply by placing the 'number' in memory and always configuring one voice system as the 'calling' party and the other as the 'called' party. Since no tone facilities were available to indicate call status information, the LEDs on the front of the X21 circuitry were used to indicate the call state (i.e. engaged, idle, talking).

4.3.3.2 No Silence Detection Circuitry

A means of detecting silent intervals in talk spurts was not included in the X21 circuitry, since this was felt to be an unnecessary complication to the design. The Z-80 software was modified, so that every block was marked as 'non-silent'. The immediate result of this was that no timing adjustment was then possible, and 64 Kbps plus network protocol overheads were transmitted between the stations.

4.3.3.3 Other Modifications

4.3.3.3.1 Lost Blocks

A packet network is always prone to losing blocks across the network. In the Universe system, while most blocks marked as 'silent' would not be transmitted across the network, some were regularly sent in order to keep the connection through the bridges
‘alive’. ‘Silent’ blocks at the end of a talk spurt were also transmitted, to ensure that the ends of words were not clipped; the amplitude of vowels, at the end of words such as ‘three’, can fall below the silence threshold. During block loss, the previously recorded block of ‘silence’ was inserted in the ‘play-back’ stream.

With the modifications described above, the receiving station did not have a block of ‘silence’ or white noise to play back when blocks were lost across the network. The software was modified so that lost blocks were replaced by the previously recorded block.

4.3.3.3.2 Timing Adjustment

As described previously, the delayed timing mechanism implemented for the Universe project used periods of silence to adjust the timing. In the Unison project, the software was modified so that adjustment could take place as soon as it was required.

4.3.4 Conclusion

This section has described the hardware and software modifications to the Universe toll quality voice system in order to perform some initial investigations on the Unison prototype network.

The Universe toll quality voice system hardware was modified to give an X.21 interface to the B.T. wide-band speech codecs, instead of the PCM codec logic. A suitable audio system was also included to give ‘hands-free’ teleconferencing capabilities.

The lack of silence detection/suppression circuitry in the wide-band speech interface precipitated significant software modifications.

These can be summarised as follows:

- **Lost blocks were replaced by the previously recorded block.**
- **Timing adjustment occurred as soon as it was required.**
4.4 Performance Assessment of the Prototype Unison Two-party Wide-band Speech System

4.4.1 Introduction

The aim of this work was to investigate how a two-party wide-band speech system might behave over the prototype Unison network.

The first part of this section describes the subjective quality of speech transmitted over the prototype network under different conditions.

Performance measurements by Siddiqui (89) are then compared to the subjective results.

The last part of the section identifies the problems that should be addressed in the next version of the voice protocol.

4.4.2 Performance Over the Prototype Unison Network

Experiments on the network were performed, and the subjective speech quality recorded.

Speech was monitored in three different network environment conditions with other traffic present:

- *local ring only*
- *looped back through a single portal*
- *over the inter-connecting Megastream*

4.4.2.1 Local Ring Only

The performance in this environment (figure 34) was generally of very good quality, even when other traffic was present.

![Figure 34: Local Ring Only Environment](image)
4.4.2.2 Looped Back Through a Single Portal

This environment was created by using the concept of a 'ghost' ring (figure 35).

![Diagram of a single portal environment]

Figure 35: Looped Back Through a Single Portal Environment

Stations were defined as belonging to either the real ring or the 'ghost' ring; two stations that might actually be physically next to one another would be specified (in the name-server) as belonging to two separate rings. Data transfer between the two stations was therefore sent to the portal, flowed around the CFR exchange, and was sent back through the same portal to the receiving station on the 'ghost' ring.

In the presence of other traffic, the only block size that would give a reasonable voice connection was 2048 bytes. Adverse effects (noticeable speech syllable repetition and loud clicks) were noticeable when the video service was being set-up or cleared down. The effect was limited to a very short duration (1 second). Loud clicks also were also heard when the level of traffic was stable.
4.4.2.3 Over the Inter-connecting Megastream

The effects noted in this situation (figure 36) were mostly the same as for the above case, except that here, a more perceptible end-to-end delay was also noticeable.

![Figure 36: Over the Interconnecting Megastream Environment](image)

4.4.2.4 Performance Summary

Under ideal conditions, over a local ring and with no other traffic, the resulting speech quality was very reasonable. On an inter-site basis, and with no other traffic, the system performance was also good. Significant problems were encountered however, when a reasonable amount of other traffic was present, and the connection was not only over the local CR.

4.4.3 Network Performance

The following is cited in reference to Siddiqui (89):

Results obtained from measuring the throughput of the portal for various different block sizes show the likely occurrence of problems for voice connections. The throughput for block sizes of 16 bytes shows a maximum achievable data rate of less than 25 Kbps, with only a single voice connection being sustainable at a block size of 64 bytes. This is because the portal was limited both by packets per second and by K bits per second. In order to pass both voice and slow-scan video (at any reasonable update rate) through the portal, the largest block sizes of 1024 or 2048 bytes (128 or 256 ms worth of speech) would have to be used.
The results also show 65 ms block delay for a block size of 2048 bytes, where the greater component of this is from block launching and formation at the portal CR interface.

Results obtained for the Ramp and Megastream show that the throughput depends on the performance of the portals, with an even greater block delay for larger block sizes (85 - 40 ms).

4.4.4 Performance Conclusions for the Voice Protocol

4.4.4.1 The Effect of Block Loss and Block Repetition

Block loss in ADPCM systems can be very detrimental (see section 3.4.4.6). Block reconstruction techniques are likely to fail if the length of the loss is great (>16-32 ms). In the Unison prototype voice system, a block size of 2048 bytes (256 ms worth of speech) had to be used. This means that even if blocks were lost individually, significant performance degradation can be expected.

Loud clicks (or speech clipping, see section 3.4.3.1.3) occurs due to discontinuities in the speech wave form at the transition between a replaced packet and the next one (Valenzuela 89).

Block repetition can be heard as syllable repetition for block sizes as large as 2048 bytes.

4.4.4.2 The Effect of Delay on Voice Connections

The effect of delay on a voice connection is very detrimental when it is greater than 35 ms (see section 3.4.3.3.2). Echoes and feed-back (singing and howl-round) are likely to occur.

4.4.4.3 Unison Prototype Wide-band Speech system Performance

The lack of silence suppression/detection facilities made timing adjustment very perceptible when change occurred during periods of actual speech. Block loss over the network would still have been a problem in this system however, even if silence detection techniques had been employed. The perceptibility of block loss was due to the size of the blocks used, which are too long for voice connections on packet networks.

The large delay experienced across the Unison prototype network meant that echoes were very perceptible, and the system was prone to feed-back. The directional
microphones used in the system were also very difficult to use; head movements reduced the reproduced sound level significantly, and the effect was annoying.

4.4.4.4 Network Performance

The results above show that the prototype Unison network was not really suitable for integrated traffic. The poor portal throughput, which was mostly due to the Basic Block formation and launching time, prohibits the use of voice.
Chapter 5
The Unison Two-party Wide-band Speech System

5.1 Introduction

This chapter describes the work done on the two-party wide-band speech teleconferencing system in the Unison project.

The aims of this work are to provide the following:

- *a system that will cope with variations in network loading, and the mixture of types of traffic that are necessary for an integrated office*
- *a reasonably priced system that provides optimum voice performance for different acoustic environments, and different untrained speakers*

The first part of the chapter discusses the two-party audio system design. The choice of microphone and provision of extra equipment to provide a robust system is considered in detail.

The second part of the chapter describes the voice protocol design. The design was based on an assessment of the expected Unison network performance (Siddiqui 89) and a survey of related research.

The third part of the chapter briefly investigates the hardware and software of the transputer, and looks at the scheduling and priority mechanisms available.

The fourth part of the chapter describes the hardware developed for the codec interface, and the test software that was written to verify its correct operation.

The fifth part of the chapter investigates the problems associated with concurrent real-time software, and the peculiarities of the Occam language in this type of application. Established buffering techniques are assessed for their suitability.

The sixth part of the chapter discusses the parallel approach taken to the software design, and uses the MASCOT design method to describe the structure of the software (Axford 89). The importance of unobtrusive statistics gathering is described in detail as is the software designed for the two-party wide-band speech teleconferencing system.
5.2 Hands-Free Teleconferencing Audio System

5.2.1 Requirements

The requirements of a hands-free teleconferencing system are that it should provide optimum voice performance, given the following constraints:

- reasonable cost
- different acoustic environments
- varying speakers; male, female, and particularly those whose first language is not that in which they are conversing

Further considerations are that the audio system should be easy to set-up, easy to use, and unobtrusive. This design should also form the basis for the three-party audio system.

A less-than-ideal acoustic teleconferencing situation does not mean however that the resulting speech quality need be mediocre. Apart from a pre-requisite of high intelligibility, the speech must sound natural, and give the effect of warmth, live-ness and intimacy.

5.2.2 Basic Components

The basic components of a hands-free teleconferencing audio system for each user are a microphone, a microphone pre-amplifier, a power amplifier and a loudspeaker. For two-party teleconferencing, the audio system layout can be seen in figure 37.

![Figure 37: Basic Audio 'Hands-free' Teleconferencing Equipment](image-url)
The most important point to consider when designing an audio system, is that the reproduced sound quality is only as good as the first element in the chain; it is therefore important to correctly choose the microphone.

5.2.3 Design Considerations

5.2.3.1 Prototype Codec Frequency Response

The prototype codec's response shows attenuation for frequencies above 3.5 kHz for gains of unity (figure 38).

![Figure 38: The Frequency Response of the Complete Codec (Lee 84)](image)

This response is suitable for teleconferencing speech, since most of the total energy of speech comes from the low frequencies (see section 3.3.2.4). When high-amplitude, high-frequency sounds occur, it is usually as a result of howl-round or room resonances, and consequently undesirable. The codec response gives some immunity to this problem.

5.2.3.2 Reverberation

The environments where the 'hands-free' teleconferencing system will be deployed are likely to be less than optimum acoustically. Reverberation (see section 3.5.1.2.1) is likely to be significant in magnitude, at least for first-order reflections (Borwick 87). Interference effects (see section 3.5.2.1.2) are also likely to be a problem in the Unison integrated office.
5.2.3.3 Feedback

The basic components of a ‘hands-free’ teleconferencing system give a closed audio loop; a microphone will almost always pick up sound emerging from the loudspeaker. For a small room, the sound picked up may be direct loudspeaker output, or it may be part of the associated reverberation. A first-order reflection can be quite significant in magnitude, and hence can be nearly as troublesome as direct sound.

Feedback can be split into two categories: echo and howl-round.

5.2.3.3.1 Echoes

Reverberation is heard closely following the original sound. Echoes are those sounds that manage to be transmitted around the teleconferencing system, and are played back locally at a significantly greater time delay than the reverberation. The perceptibility of the echo depends on the round-trip delay of the system (see section 3.4.3.4.2).

5.2.3.3.2 Howl-round

Howl-round is well known in public address systems. Here, the delay through the system is very small, and feedback causes colourations in the reverberation field of the speech sound. As the gain of the system is increased, parts of the speech appear to ‘ring’, with continuous howl-round occurring as the system gain approaches unity.

The colouration of speech and room resonances were discussed in section 3.5.1.2.4. The ringing described above occurs when speech sounds excite room resonances.

Speech sounds echoed back by the closed audio loop of a hands-free teleconferencing system are coloured by both rooms. Apart from the fact that the frequency responses of the rooms are different (for example, resonant frequencies for one room maybe troughs in the response of the other room), sound also decays in proportion to the distance from the source. Howl-round consequently occurs at a higher system gain than is the case for a single room. Nevertheless, howl-round is still the limiting factor for system gain in teleconferencing systems.

5.2.3.4 Background Noise

In the Unison integrated office, background noise (see section 3.5.1.2.5) from the workstation and video monitor can be picked up by the microphone.
5.2.3.5 Mono Reproduction

The human auditory mechanism is binaural; we can locate sounds with great accuracy due to the difference in phase and intensity of the sound arriving at each ear (Lim 83). The binaural mechanism also gives the power of discrimination (cocktail party effect). For speech that has had reverberation added to it, the hearing mechanism gives the ability to discriminate against the unwanted sound.

The mono reproduction of speech reduces our powers of discrimination, which means that the apparent reverberation is increased. The resulting speech consequently sounds hollow and blurred.

5.2.3.6 Human Behaviour Characteristics

The teleconference participant is likely to be untrained in the use of microphones; head movement, and variation in distance from the microphone can cause changes in the pickup.

5.2.4 Audio Equipment

5.2.4.1 Microphones

The use of microphones in other applications, such as broadcasting and public address systems, is well documented (Olson 57, Nisbett 93). Whilst both situations have certain similarities to teleconferencing, there are enough discrepancies to warrant a different solution.

Three types of microphone were considered as candidates for the Unison integrated office: a dynamic first-order cardioid, a lapel microphone, and a pressure zone microphone.

5.2.4.1.1 Dynamic First Order Cardioid

The dynamic first-order microphone is directional, and is the most common choice for public address systems.
A typical public address system layout is shown in figure 39.

![Figure 39: Public Address System Layout](image)

Important points to note about the system:

- The microphone and loudspeakers point in opposite directions to one another. The first-order cardioid microphone (which rejects sound from the rear) consequently picks up very little directly-radiated sound from the loudspeakers.
- The room is probably large, and acoustically engineered. This means that any first-order reflections will have sufficiently decayed in magnitude before they are picked up by the microphone.

This system provides good echo and howl-round immunity for the reasons identified above. The disadvantages of using this type of microphone are that the speaker must remain fairly fixed in distance from the microphone, and in head orientation. (This restriction is a consequence of the method of operation of the microphone, the pressure gradient method is more fully explained in Borwick 87). The movement restrictions are not a great problem in this situation, as the speaker will probably not want to move around very much. If movement is necessary, either a hand-held dynamic microphone (such as used by musicians) or a lapel microphone (such as used by lecturers) may be substituted, and still provide good howl-round protection.

5.2.4.1.2 Lapel Microphone

If movement by a person is required in a situation, then an omni-directional lapel microphone can be used (Nisbett 93). The omni-directional microphone uses a different method of operation to that of the dynamic cardioid microphone (the pressure method, which is more fully explained in Borwick 87).
While movement restrictions have now been eased, a certain amount of speaker discipline is now required; any clothing rustles will now be picked up as will heavy breathing. The lapel microphone is also only suitable for one person per site.

5.2.4.1.3 Pressure Zone Microphone (PZM)

The pressure zone microphone provides a hemispherical response, and is well suited to speech and music applications. The microphone consists of an electret microphone capsule that is permanently fixed a short distance away from a plate or boundary (Nisbett 93, Crown 93). Generally, microphones suffer from interference effects when placed near a reflective surface (see section 3.5.2.1.2), but as the diaphragm is moved closer to the surface, the resulting re-enforcement or cancellation of a particular frequency moves up in the spectrum. At half an inch away from the reflective surface, the effect is limited to the highest part of the human audio range. At closer distances the effect becomes inaudible. The PZM does not suffer from the proximity effect, since the microphone capsule is fixed a short distance away from the boundary.

Another feature of the PZM is that, since the input to the microphone is a combination of the direct and reflected sound from the plate, sound pressure is doubled. This improves the level of the original sound compared to the reverberation content in the pick-up (Crown 93).

The PZM places no movement restrictions on a speaker, either in terms of head incidence or in terms of discipline to stop clothing rustles or heavy breathing.
5.2.5 A Hands-free Teleconferencing System

A typical teleconferencing system layout can be seen in figure 40.

![Diagram of the Unison Teleconferencing Desk Layout](image)

Figure 40: The Unison Teleconferencing Desk Layout

Note that the diagram only shows the workstation and wide-band speech system. Other applications developed as part of the Unison Integrated Office are not shown for clarity.

There are important differences to note between this situation and those described previously:

- The microphone and loudspeaker both point in the same direction, and are close together. This means that unless a very directional microphone is used, some direct sound will be picked up by the microphone.

- The room is probably small and likely to be acoustically less than optimum; first-order reflections off the desk and walls (principally that behind the speaker) are likely to be of significant magnitude, and will be picked up by the microphone (even if a very directional microphone is used).

Use of a dynamic cardioid microphone for a teleconferencing system will not give howl-round immunity for the reasons detailed above. Interference from the desk in front of the speaker is likely to cause colourations in the input speech spectrum. The disadvantages associated with a directional dynamic microphone, those of a readily perceptible loss of volume for different distances from the microphone, and head orientation, are seen as unacceptable restrictions for a teleconference participant.

Use of a lapel microphone for a teleconferencing situation means that care must be taken by the speaker not to move into the output field of the loud speaker, as this will lead to howl-round. Unfortunately, due to the positioning of the person and the
loudspeaker, this is unavoidable. This microphone is therefore thought to be unsuitable for use in a teleconferencing situation.

Use of a suitably placed PZM will give rejection of first-order reflections, particularly those reflections from the desk top. Head movements and small changes in distance from the microphone will not affect the pick-up by the microphone. The comparatively low level of reverberation picked up by this microphone improves the subjective mono-aural response. This microphone is thought to be the best choice for a teleconference system.

5.2.5.1 Pre-Amplifier, Power Amplifier and Loudspeaker

For the Unison teleconferencing system, an ordinary high-fidelity power amplifier and loudspeaker was more than adequate. A simple pre-amplifier, constructed out of low-noise audio amplifiers, was sufficient for microphone amplification.

5.2.5.2 Improving the System

5.2.5.2.1 Reverberation

The use of a PZM did much to lessen the perception of reverberation. Reverberation produced from the wall behind the conference participant and other reflective surfaces was reduced further by having curtains on the walls, carpeting on the floor, etc.

5.2.5.2.2 Binaural/Mono Reproduction

The Unison teleconferencing system was restricted to mono reproduction because of network bandwidth constraints. However, the blurred/hollow response usually found in mono-aural systems was significantly reduced by the reverberation response outlined above.

5.2.5.2.3 Echoes

Echoes, due to the presence of a closed audio loop, presented significant problems to the Unison teleconferencing system. This problem is usually eased by providing echo cancellers or by attenuating the microphone input when the loudspeaker is active.

Echo cancellers produce an estimate of the echo from a filter and subtract it from the recorded speech. Apart from the fact that the wide-band teleconferencing system did not have access to a digital representation of the un-coded speech, the long echo path, and
its variation due to the variable delay across the network made the cost of such a filter prohibitive.

The method of reducing echoes by attenuating the microphone output when the loudspeaker is active was rejected, since the fading in and out of background noise can be irritating, and speech is always lost.

The perceptibility of echoes in the Unison teleconferencing system can be minimised by keeping the end-to-end delay as small as possible (see section 3.4.3.4.2). This approach was ultimately adopted.

5.2.5.2.4 Frequency Shifters

Frequency shifters are relatively cheap devices that can be used to increase the gain at which singing and howl-round become a problem (which is more fully explained in Jones 73).

The frequency response of any room is made up of a series of peaks and troughs, the location of which depends upon the dimensions of the room, furniture position etc. This colours speech, and at some locations in the spectrum leads to singing, where room resonances have been excited (see section 3.5.1.2.4).

The devices shift the whole spectrum up in frequency by 5 Hz (10 Hz if two are employed), which is imperceptible to the listener (Olson 57). Shifting the spectrum in this manner places what were peaks in one room onto troughs in the same room, when the speech travels round the closed audio loop.

For small rooms, the rise in gain before the onset of howl-round is not as good as for larger rooms, but a gain of approximately 2-3 dB can still be expected (Jones 73). Frequency shifters were used in the Unison wide-band speech system, and increased the maximum gain by nearly 3 dB.

5.2.5.2.5 Background Noise

Background noise (see section 3.5.1.2.5) in a closed audio loop is the sum of the background noise emanating from both rooms. A solution to this problem is by using a noise gate (Nisbett 93). This device attenuates sound that is deemed to be background noise. This method was rejected for the Unison wide band speech system, since its use will give problems similar to those found for silence detection (Ades 87), certain speech sounds at the end of words may be attenuated, and background noise may be heard cutting in and out.
5.2.6 Subjective Audio Performance

The audio system with the improvements increased intelligibility, and was subjectively assessed by the author as being 'natural'.

5.2.7 Conclusion

The audio system used in the prototype wide-band speech system consisted of the following: a directional microphone, simple pre-amplifier, power amplifier and a pair of loudspeakers. As part of the work done for this project, the system was modified as follows:

- replacement of the directional microphone with a PZM
- the inclusion of a better, two-stage pre-amplifier
- the inclusion of a pair of frequency shifters
- the provision of carpeting in one of the rooms used for the teleconferencing system

The audio system provided for the Unison 'hands-free' teleconferencing system achieved the performance requirements expected of it. The system was cheap, easy to set up, unobtrusive and most importantly was suitable for most different acoustic environments. No training for conference participators was necessary and the system provided excellent quality for both male and female participants. The system did not suffer from voice clipping or howl-round at reasonable gains. The resulting sound was subjectively assessed as being 'natural'.

A pre-requisite for other parts of the Unison wide-band teleconferencing system design is the minimisation of end-to-end delay, in order to reduce the perceptibility of echoes.
5.3 Voice Protocol Design

5.3.1 Requirements

The requirements of a voice protocol are that it should facilitate a system that will cope with all expected network loading conditions and mixtures of traffic.

5.3.2 Unison Network Performance Survey

A voice protocol design must be dictated by the environment in which it is to operate. Careful assessment of the error rates and delay expected can lead to savings in terms of delay budget, algorithm complexity (see section 3.4.4.4.1) and consequently processor power and memory. The assessment of network performance is based on the Ph.D. thesis of Mahboob Siddiqui (Siddiqui 89). The figures quoted are used as a guide for the design of the voice protocol, and no conclusions about the network performance should be drawn from this text.

5.3.2.1 Error Expectations

The Unison network performs Cyclic Redundancy Checks (CRC) and subsequent re-transmissions. Corrupted blocks are therefore almost never received. Consequently, the only error problem to be addressed is one of block loss.

Within the Unison network, there are three points at which block loss can occur. These are the Cambridge Fast Ring, the portal and the Ramp/ISDN inter-site connection.

5.3.2.1.1 Cambridge Fast Ring

There are two or more CFRs per site (see sections 2.4, 2.6.1). These are the client CFRs and the exchange CFR. Block loss occurs during ring re-frames, and also if the destination CFR station (portal or ramp) is backing off due to congestion. Ring re-frames occur only relatively infrequently, but during these conditions more than one consecutive block can be lost.

5.3.2.1.2 Portals

Results for block loss at portals indicate a loss of less than 10% for extremely heavy loading conditions (6, 600 packets per second) with no block loss for loads of less than 5, 500 packets per second. Packet loss characteristics for the portal under heavy loading conditions (above saturation) indicate that most packets are lost individually, rather than consecutively.
5.3.2.1.3 Ramp/ISDN Inter-site Connections

Results for this part of the Unison network indicate that this is the throughput bottleneck. No traffic loss occurs up to a throughput of 4950 packets per second (full-duplex). After this maximum, blocks begin to get thrown away.

5.3.2.1.4 Summary of Error Expectations

Block loss across the Unison network occurs during CFR ring re-frames, when a few consecutive blocks are lost, and when the maximum throughput of the ramps is exceeded. The provision of a priority mechanism within the Unison network has benefits in terms of delay for voice streams, but has disadvantages in terms of block loss, since voice blocks are preferentially thrown away from congested nodes.

5.3.2.2 Delay Expectations

Delay can be introduced across the Unison network by all the component parts of the system. The CFR nodes employ re-transmissions in the case of corrupted blocks, or receiver CFR chips backing off due to congestion. Re-transmissions introduce only a relatively small amount of jitter due to the high-speed of the CFR.

The major contributors to delay and jitter are the portals and the ramp/ISDN interconnection network.

5.3.2.2.1 Portals

Loading below the saturation point of the portal shows that most packets experience less than 0.5 ms delay (average 0.4 ms and jitter ±0.2 ms). Under heavy loading, most packets experience less than 1.2 ms (average 0.7 ms and jitter ±0.6 ms).

5.3.2.2.2 Ramp/ISDN Network

For light loading conditions, results indicate an average delay of 4 ms, with almost no jitter. For high loading conditions (below the saturation point), the average delay increased up to 9 ms, with a jitter of 5 ms (minimum 6 ms, maximum 10 ms).

5.3.2.2.3 Delay Performance Summary

The delay experienced across the Unison network has been decreased for delay-sensitive traffic by the introduction of a priority mechanism (see section 2.6.4.2). Voice packets are preferentially processed by the ramp and portals. For voice connections under light loading conditions, the average delay expected across the full Unison network is
7 ms, with almost no jitter expected. Under heavy loading conditions however, the average delay increases up to 14 ms, with a jitter of 3 ms (minimum 12 ms, maximum 15 ms).

5.3.3 Design

A voice protocol must be 'light-weight', and must minimise the perception of block loss and variations in delay (see section 3.4.4). There are a number of parameters that can be adjusted to achieve optimum voice quality: block length, reconstruction buffer size, voice reconstruction method and timing mechanism.

5.3.3.1 Block Length

Some considerations behind the choice of a suitable block length have already been put forward in section 3.4.4.5. The choice of block length is governed by the wish to reduce the perceptibility of lost blocks and the need to maintain small delay across the network for voice connections.

Delay across some networks may be reduced by assembling packets into blocks of more than one packet, since node processing overhead occurs on a block basis. Within the Unison network however, construction of individual packets into blocks of longer length is expected to be on an end-to-end basis, and the responsibility of the user. Since the network deals with traffic only on a packet basis, the construction of blocks of longer length does not affect the percentage of packets lost, nor does it reduce the amount of per block processing time expended by nodes. Consequently, this method of transmission is useful only for services that require error free transmission, and do not have tight delay requirements.

For networks where node processing is on a per packet basis, the delay experienced at the transmitter (by collecting enough samples for a packet) can be reduced by only partly filling the packet. This also produces benefits in terms of the perceptibility of lost packets on the resulting speech, especially where block loss is on an individual basis, as is the case with the Unison network. The disadvantage in only partly filling packets lies in increasing the number of packets across the network, which may lead to congestion, and increased block loss.

The length of Unison voice packets was chosen to be 3.5 ms worth of speech, or 1 CFR slot completely full. At this value, block losses are expected to be small, and the packetisation delay is also reasonably low.
5.3.3.2 Timing

Examination of the expected delay results (above), indicates that delay should range from a negligible delay on a local CFR, to inter-site values of 7 ms and almost no jitter under light loading conditions, to 14 ms with 3 ms of jitter under heavy loading conditions. One might expect therefore, that a Strict timing mechanism which took account of the differences in jitter on local links and inter-site connections would be the best approach; the maximum expected jitter on inter-site connections is as follows:

\[
\text{(maximum heavily loaded network delay)} - \text{(lightly loaded network delay)}
\]

\[
15 \text{ ms} - 7 \text{ ms} = 8 \text{ ms (see above)}
\]

Unfortunately, delay budgets are restricted by the lack of echo cancellors (echo perception increases with increasing delay) (see section 5.2.4.2.3). Consequently, the voice protocol should not impose more delay on a voice link than is absolutely necessary. Usage patterns of the Unison office mean that on a per call basis, it is a fair assumption that either the network is lightly loaded or it is heavily loaded; the video traffic is either present or it is not! This means that the jitter assessment can be revised to 3-4 ms for the purposes of the Unison project.

A strict timing protocol with a reconstruction delay of \((2\times3.5)=7 \text{ ms}\) will give a network heavily loaded delay of 25.5 ms (packetisation delay = 3.5 ms, delay across the network = 15 ms, reconstruction delay = 7 ms) and a network lightly loaded delay of 17.5 ms.

There are two problems with using a strict timing algorithm:

- There is no mechanism for adjusting the delay when network loading conditions ease, and the delay becomes 7 ms (the receive buffers will then always be full).
- The clocks at either end of a voice link are independent, will never be the same frequency, and will drift over the duration of the call. Clock drift (see section 3.4.3.5.4) will force one receive buffer to empty and the other to fill up.

Clock drift adjustment is usually made in one of three ways (Ades 87):

1. by slightly trimming the frequency of one clock to match the other
2. by compression or expansion of silence intervals
3. by adding a small amount of delay to the reconstruction delay in a Strict Timing voice protocol to account for clock drift over the expected duration of the call

Methods 1 and 2 would involve designing extra circuitry to that discussed in chapter 4 (section 4.2.3.1). Method 3 is not favourable, as it increases the end-to-end delay. The solution to the problems identified above was to use the delayed timing method.
In the delayed timing method (see section 3.4.4.4.2), extra blocks are added to an
habitually empty buffer and blocks are lost from a full buffer. This method would not
normally be considered in a PCM voice packet network, since the block length is
normally in the range 16-32 ms worth of speech samples (see section 3.4.4.5), and silence
detection facilities are normally available.

Within the Unison wide band speech system however, a number of factors make
the method of coping with block loss and clock drift possible:

- The block length is 3.5 ms. This fact in conjunction with a suitable voice
  reconstruction method should mean that perceptually the effects are limited (see
  section 3.4.4.5), providing adjustment is made over a period of time, where each
  instance of adding or losing blocks involves just one block.
- Adjustment is only required very infrequently. The clocks used are accurate crystal
  oscillators, and loading variations across the network are only significant if the
  video is switched on or off.

5.3.3.3 Voice Reconstruction

The effect of block loss and a survey of possible block reconstruction methods has
been discussed in chapter 3 (sections 3.4.4.5, 3.4.4.6). For a SB-ADPCM voice system
(see section 3.3.4.3), the choice of block reconstruction method is based on the need to
supply a suitable replacement for the lost block, whilst maintaining decoder tracking of
subsequent speech samples (see section 3.4.4.6).

The choice of block reconstruction technique used for the wide-band voice system
is restricted because access to the digital representation of the speech wave-form is not
possible. This excludes methods such as wave-form substitution, and many of the better,
but more complicated algorithms available, such as pitch replication (see section 3.4.4.6).

Possible reconstruction methods for SB-ADPCM, which do not require use of the
digital representation (see section 3.4.4.6) are as follows:

- Zero Amplitude stuffing
- Packet Repetition

Some other methods, such as Odd-Even sample interpolation (see section 3.4.4.6)
would be possible at the code-word level, but the increased delay (collecting twice as
many samples for transmission as for one packet) and its use of processor power would
make this method prohibitive.

The best method in this situation is packet repetition, which can be used to replace
periods of block loss of up to 16 ms worth of speech (see Table 1, section 3.4.4.5).
5.3.4 Conclusion

A delayed timing method will provide the best means available of coping with variations in network loading, and with clock drift.

The network gives priority processing at nodes to voice packets, which reduces the delay variations expected by voice packets. A lower jitter assessment consequently reduces the reconstruction buffer size needed. A smaller reconstruction buffer means a smaller end-to-end delay.

While more voice blocks than data blocks are thrown away at congested nodes, the behaviour is acceptable, as most are thrown away singularly, rather than in clusters. This last fact will have consequences for voice reconstruction, since even a relatively simple method will work well in these circumstances. A simpler reconstruction method means less processing power is required to produce optimum quality voice and less end-to-end delay can be expected.
5.4 Choice of Processor - the Transputer and Occam

5.4.1 Choice of Processor

The choice of processor for the Unison wide-band voice system is based on the fact that it must be capable of processing both-way real-time traffic. The processor does not require large data storage facilities, nor a complex user interface.

Most of the applications equipment and network gateways/managers will have similar requirements to that above. Examples of other such devices are the video equipment, and the ramps and portals. These can be contrasted with machines which process bursty data, such as the secretary and workstation. To avoid the duplication of software within the Unison project, it was decided to use only one or two different processors.

The requirements described above are satisfied by the transputer.

The workstation and secretary devices have different requirements to those identified above for voice equipment etc. A more conventional processor, the 32-bit 68020 was adopted for this task. This processor is not discussed further in this report, since it has no relevance to the wide-band speech system.

5.4.2 The Suitability of the Transputer as a Real-Time Embedded Controller

For process control type applications, the requirements made on both the hardware and software of the system are of a parallel nature. Consequently, the software should have a multiple process architecture, where each process can execute concurrently with other processes. This can be achieved by either sharing the CPU time of a single processor out amongst the processes on a time slice basis, (which results in quasi parallelism), or by employing more than one processor to give genuine parallelism.

5.4.2.1 Real-Time Hardware

The block diagram of a transputer can be seen in figure 41 (Inmos 88).
The main processor can either be of a 32-bit (T4 series) or 16-bit type (T2 series), with 4 Kbytes of fast internal on-chip RAM. The transputer executes a fast reduced instruction set (RISC), which inherently supports parallelism. The four links are connected to the main processor via four link interfaces. A link between two transputers is implemented by connecting a link interface on one transputer to a link interface on the other transputer. Communication occurs over two half-duplex paths, which operate at the standard speed of 10 Mbps, or alternatively at 5 or 20 Mbps. Transputers do not require peripheral hardware to communicate to each other, which means that adding another transputer to a system is straight-forward (provided external RAM is not required). The interfaces can manage communication to other devices (including direct memory access (DMA)) independently from the main processor.

A transputer can therefore simultaneously communicate on all four links AND execute an internal process.

This last feature is very important when considering the transputers suitability as a real-time embedded controller.

For quasi-parallel processing on one transputer,
the presence of a scheduler implemented in hardware means that context
switching is very fast; minimal time overhead is incurred by processing software as a
number of concurrent processes, rather than as a single sequential process.

5.4.2.2 Real-Time Software

5.4.2.2.1 The Occam Language

The Occam language is based on Communicating Sequential Processes, a fuller
explanation of which can be found in Hoare (85).

The Occam language (Inmos 88) is essentially constructed from three primitive
actions:

- assignment
- input from a channel
- output to a channel

The above can be combined by using SEQ, PAR and ALT constructs: sequential,
parallel and alternative execution. The sequential construct is deterministic, whereas the
parallel and alternative constructs are non-deterministic.

For the PAR and ALT constructs, some amount of determinate execution can be
introduced by the use of the PRI PAR and PRI ALT constructs, which assign priority to
the component processes in the case of the PRI PAR construct, and which assign priority
to the component channel inputs in the case of the PRI ALT construct.

In order to evaluate the transputer’s performance during quasi-parallel operation,
it is necessary to investigate transputer support for sequential and concurrent processes.
Scheduling, priority and channel communication are also important performance
considerations.

5.4.2.2.2 Sequential Processes

The transputer has six registers available for use within a sequential process
(figure 42) (Inmos 88).
A, B and C form the evaluation stack. Other registers (workspace, next instruction and operand) have the usual functions.

5.4.2.2.3 Concurrent Processes

A process on a transputer can be in one of three states:

- suspended
- executing
- executable

A process that is suspended is waiting for an input or output communication on a channel. Within each transputer, there is only one process being executed at any instant in time, all other processes that are capable of being executed are queued. Consequently, two other registers are required to implement quasi-parallelism (figure 42).

5.4.2.2.4 Scheduling

When an executing process is waiting for communication, the scheduler stores the contents of the next instruction register in the process workspace. When this happens, a new process is swapped in (the process from the front of the executable queue has its next instruction pointer restored to the corresponding register). When a suspended process becomes executable once more, it is appended to the queue.
From the above, it can be seen that a number of processes operating in parallel incur a time overhead, as queues must be manipulated, and registers initialised. Therefore, whilst this overhead is small, a series of independent assignment statements should be executed sequentially and not in parallel.

5.4.2.2.5 Priority

The transputer implements two levels of priority. There are two scheduling algorithms to implement priority:

- Low priority processes are scheduled using a round-robin algorithm (Burns 87). The process is executed until it either:
  - can no longer proceed
  - a high priority process becomes executable
  - it has been executing for one time-slice and has reached the end of a control structure

- A high priority process can pre-empt a low priority process and once it is executing, it continues until it can no longer proceed; it cannot be pre-empted. A high priority process can become executable if:
  - a delayed timer has expired
  - it was waiting to communicate with a low priority process that is now ready to communicate
  - it was waiting to communicate with a process on a different processor that is now ready
  - it was waiting for an interrupt on the event-in channel, and the interrupt has been triggered

Two important conclusions can be drawn from the above:

- The transputer's implementation of an interrupt is not really an interrupt in the true sense.
  This is because if there is more than one process executing at high priority, synchronisation with the interrupt will not occur until the high priority process servicing the interrupt is scheduled; when the current high priority process becomes suspended and the relevant interrupt servicing process is scheduled. This means that the reaction time is indeterminate.

- If a high priority process is waiting to communicate with a low priority process, it must wait until the low priority process is scheduled and it reaches the communication point.
  The high priority process has effectively been demoted in priority, while it waits to communicate with the low priority process. More preferable in this case would be a scheme whereby the priority of the low priority process was temporarily increased while it communicated with a high priority process, or more than two levels of priority existed.

5.4.2.2.6 Channel Communication

Channel communication can either take place between processes running on the same transputer or between processes running on different transputers (external channels).
The concepts of communication are the same internally and externally: a program developed on a single transputer can be easily re-configured to run on a network of transputers (Inmos 88).

a) Internal Channels

When a channel is defined, a single word of storage is allocated in memory, and initialised as empty (Burns 87). As only one process can be executing at any one instant, then one of the two processes attached to the channel will want to communicate first. The process accesses the channel word and finds that it is first and must wait. It stores its ‘ID’ in the channel word and the process becomes suspended.

Eventually, the other process reaches the point of communication and is scheduled. Using the ‘ID’, the second process initiates the transfer, and the message is copied across to the relevant process. The second process continues after the transfer.

The important point to note about channel communication is that the CPU expends time in copying the message across.

b) External Channels

If the channel is external, then the CPU delegates responsibility to the independent link interface. The link interface transfers the message and reschedules the process.

Consequently, no CPU time is expended in communications across hard links.

The protocol used to transfer data is based on the transmission of serial bytes.

This means that processors of different word length can communicate.

5.4.3 Conclusion

The key requirement for an embedded real-time processor within the Unison wide-band voice system, is that it must simultaneously process full-duplex traffic. This requirement is met by a transputer.

The suitability of the transputer as a real-time embedded controller can be described in terms of the following:
• No CPU time is expended in communicating externally.

• The scheduling of internal processes is very fast.

• Both hardware and software can be easily re-configured if more processing power is required.

• Processors of different word lengths can communicate; networks can be tailored to processing power needs, representing cost savings.

The transputer's weaknesses in this application can be summarised:

• There is a small overhead for a number of processes executing on a single transputer.

• The transputer implementation of an interrupt gives indeterminate execution if there are other processes executing at high priority.

• Communication between a high priority processes and a low priority process proceeds as if both were low priority processes.

The advantages of using the transputer as a real-time embedded controller outweigh the disadvantages in my opinion. Furthermore, the disadvantages of the transputer can be alleviated to a large extent by the judicious use of buffering techniques (see section 5.6.5).
5.5 Hardware Design and Implementation

5.5.1 Requirements

The embedded real-time controller for the wide band speech interface to the Unison network is the transputer (see section 5.4). Circuitry to interface the transputer to the CFR (see section 2.6.1) and the non-standard SB-ADPCM codec (see section 3.3.4.3) was implemented.

The CFR was built by Steve Temple of Caman Systems LTD, and consists of a CFR interface, memory mapped to a T414 transputer, with 1 Mbyte of associated DRAM.

The transputer interface to the BT 7 kHz audio codecs was originally developed as part of a final year project by (Chapman 87). However, as the interface did not work properly, and was significantly re-designed, a discussion is included here.

The requirements for this circuitry:

- reusable in the three party teleconferencing system
- cheap
- robust

5.5.2 Design Considerations

5.5.2.1 The Codec Interface and the X21 Standard

The X21 standard (CCITT X.21) was developed for serial synchronous full-duplex operation between data terminal equipment (DTE) and data circuit-terminating equipment (DCE). Call set-up and clear-down facilities are provided in the standard.

The 7 kHz audio codecs (Lee 84) do not use the control signals in the standard interface; the codecs operate in permanent 'call-in-progress' mode. The signal wires used are the transmit and receive, and the bit and byte clocks, with DCE ground used for the receivers in the codec.

The codec is configured as the DTE and should consequently receive its timing signals from the DCE. The byte clock is required by the DTE, so that 'in-band' framing signals do not need to be used.
5.5.2.2 Converting X21 Communication into the Transputer Protocol

A possible option for interfacing the transputer to the 7 kHz audio codec would be to memory map the data into the transputer's address space. This solution was immediately rejected however, since the data rate is not very high (64 Kbps) and this method is complex. The requirements are well within the capabilities of the transputer family link adapter (CO11), which is a much simpler solution.

The CO11 (Inmos 88) is supplied by Inmos Ltd., as a means of converting between transputer links and standard micro-processor architectures. In mode 1, the serial bi-directional data is converted into two 8-bit wide parallel data streams, which operate using hand-shakes.

The use of the CO11 meant that extra circuitry was required to convert two 8-bit parallel buses into an X21 interface.

5.5.2.3 Expected Processing Power Required

The transputer to CFR interface software was developed prior to this interface at Rutherford and Appleton Laboratories. Examination of the operating principles of the design revealed that communication to the Unison network secretary was done on a regular basis, and that during this period of time, the transputer would not be available to service the codec interface (process running at high priority). Since the X21 interface was synchronous, using the main T414 transputer to service both interfaces would be unacceptable. Consequently, an extra transputer was added to the wide band speech system interface to service the codec.

One of the requirements for the hardware developed for this system is that it is cheap. The transputer is available in several different word sizes and internal clock speeds. The speed of the voice data received from the audio codecs is 64 Kbps. Using the link adapter, as discussed above means that data will be transferred to and from the interface in 8-bit units. The data rate and expected data input size means that only a relatively slow 16-bit transputer needed to be used for this interface (T222) (Inmos 88).
5.5.3 Circuitry

The circuitry in this interface can be split into a number of sections:

- *Bit and Byte clock generation.*
- *Serial to Parallel Data Conversion Circuitry (and vice versa)*
- *CO11 Related Circuitry*
- *Transputer circuitry*

5.5.3.1 Bit and Byte Clock Generation Circuitry

These are derived on board from a high quality 4 MHz crystal module to give a 64 Kbps bit clock and a byte clock (figure 43).

5.5.3.2 Serial to Parallel Data Conversion Circuitry

The circuitry can be seen in figure 44. Serial to parallel and parallel to serial data conversion was achieved using 26LS31 and 26LS32 chips. Latch (74LS373) chips were used to interface to the CO11.
The signal which is fed to the parallel to serial converter (which indicates that new values on the parallel inputs should be read) is a shortened version of the inverted byte clock. This pulse was made narrower so that the serial output of this chip is continuous; if the inverted byte clock was used directly, there would be small gaps in the output serial data, where the bits were being placed in an internal register before being clocked out.

A further point to note about this circuitry is that the bit clock is inverted before being fed to the serial to parallel converter. This is due to the data circuit layout in the 7 kHz codec, where the A/D and D/A converters are arranged to operate out of phase, so that a single sequential processor can service both the transmit and receive paths.
5.5.3.3 C011 Related Circuitry

5.5.3.3.1 System Services

The system services required for the C011 are VCC, GND, CapMinus, ClockIn, Reset and SeparateIQ (Inmos 88).

Important points to note:

- **capacitor type**
  
The capacitor connected between VCC and CapMinus should be a 1 micro farad low inductance low leakage capacitor, preferably a ceramic. Failure to comply with this requirement means that the chips internal power supply and hence internal clock will drift and data communicated between the transputer and the C011 may be lost.

- **SeparateIQ**
  
  This selects the C011 operating mode and the transputer link communication rate. It would be desirable to select the 20 Mbps rate, but as the circuit board is expected to be of less than ideal quality, a data rate of 10 Mbps was selected.

- **ClockIn**
  
The transputer family uses a standard external 5 Mhz, preferably a crystal module since stability is important.

- **Reset**
  
  Reset initialises the C011 to a known state. LinkIn must be held low.

5.5.3.3.2 Handshake signals - Input

Data is transferred to the serial to parallel conversion circuitry (and vice versa) under the control of handshake signals. It is extremely important that all handshake signals are electrically clean, since spurious spikes on the signals also register as valid signals.

This handshake is composed of two signals: IVALID (generated by the circuit), which indicates that data present at the input of the link adapter is valid and IACK (generated by the link adapter), which indicates that data has been read and more can be supplied. An edge-triggered flip-flop was used to generate the IVALID signal from the byte clock, this being reset by the presence of the IACK from the link adapter.

5.5.3.3.3 Handshake Signals - Output

This handshake is composed of two signals; QVALID (from the link adapter), which indicates that the data present at the output of the link adapter is valid, and QACK (generated by the circuitry), which indicates that the output data has been read and more can be supplied. An edge triggered flip-flop was used to generate the QACK signal from the byte clock, the output being reset by the presence of QVALID from the link adapter.
Consequently, it can be seen that the processor must guarantee that it can both read from and write to the codec once every 125 microseconds (for a data rate of 64 Kbps). If this is not the case then data will be corrupted; the same value is loaded for a second time into the parallel to serial converter and clocked out and a value at the receiver latch input will be overwritten before it can be read. This situation produces performance degradation in SB-ADPCM, which persists after the error due to the decoder memory (Jayant 84). The perceptibility of the error for SB-ADPCM is lower than it would be for PCM, since a low amplitude error wave-form (lower due to the prediction process at the encoder) smeared over a long time is less annoying than a large, single instance error spike.

5.5.3.4 Transputer Related Circuitry

The system services are VCC, GND, CapPlus, CapMinus, Reset, Analyse, Error, BootFromRom and ClockIn (Inmos 88).

Important points to note:

- capacitor type
  as above
- ClockIn
  as above
- Reset
  Reset initialises the transputer to a known state. All LinkIn signals must be held low.
- Analyse
  If this signal is taken high, then the transputer will halt as soon as possible. The status of the transputer is preserved for subsequent analysis.

5.5.4 Circuitry Initial Conditions

The transputer either boots from a link, or from ROM. If the processor is configured to boot from a link, it will try to load bootable code from any of the four links. Therefore, the hand-shake signals to the CO11 must be inhibited during booting (since the CO11 is connected to a transputer link). A flip-flop was used as in figure 45 in conjunction with a NOR gate to inhibit the byte clock after a RESET.
The byte clock is enabled at the start of transmission by writing to the link adapter output (QVALID signal).

5.5.5 Servicing the Hardware

The hardware has to be serviced at regular intervals (every 125 microseconds for a 64 Kbps stream). The reasons for choosing the COII were discussed above.

At first sight, the polling method appears to tie the processor into wasting a lot of time checking to see if the interface is ready. It also appears to allow the possibility of data being lost if the servicing processes are not scheduled on-time. The provision of an independent link interface means that no time (or very little) is spent by the transputer in communicating data to the COII. The transputer’s scheduler means that processor time is not spent checking the interface to see if data can be written out in the first place. However, the problem of scheduling the correct process, once communication is deemed as being able to take place, will still be a problem. This problem must be allowed for when designing the software.
5.5.6 Final Circuit and PCB Layout

The full circuit can be seen in figure 46. There were two X21 interfaces and a T222 transputer on the board. The two X21 interfaces were provided to allow for possible acoustical three-party wide band speech experiments in the absence of delay (with mixing at the analogue level). PCB layout was designed using a (Computer Aided Design (CAD) system, and two double sided boards were populated and tested.

The transputer is a device with three rows of pins. The PCB production capability of the electrical engineering department only allowed for double-sided circuit boards with no through-hole plating. The boards required a great deal of effort in laying out the circuit to have most component connections on the soldering side of the board. Significant power decoupling (at the power entry to the board and at the power connection to each chip) capacitors were provided to improve the noise immunity of the circuit boards.
Figure 46: Full Circuit for the Transputer to X.21 Interface
5.5.7 X21 Interface Board Testing

The synchronous capabilities of the X21 board were tested by a program running on the transputer (T222) (figure 47).

The transmit and receive pins of the X21 interface were connected together. In order to verify correct synchronous operation, it is not sufficient to simply write a byte out to the circuitry and match it with the received byte. What is required is a program.
out to the circuitry and match it with the received byte. What is required is a program which continually sends out a stream of bytes (modulus 255), and checks to see that they are all received with none missing, no duplication, and all uncorrupted.

The program makes use of a monitor program to control the keyboard and screen. The basic control software (character in and character out, with echo to the screen) is based on the sample software supplied by Inmos. This interface was developed further to include a large buffer for messages from the application, since the screen is slow. Also included was software to interpret received messages, and to control the input to the screen process. This interface program was used throughout the project (see section 5.7.2.4.1).

The transputer is good at manipulating data equivalent to its word size, whereas byte manipulation is time-consuming. Data for transmitting to the hardware was therefore assembled into 32-bit integers, and received as such. The generator fills the integers with 4 bytes 0, 1, 2, 3 and then 4, 5, 6, 7 etc. (modulus 255). Integers received at the receiver process are checked off against a 64 integer array. It is important to note that when the integers are transmitted to the hardware by the link adapter, they are in byte form. Consequently received arrays may not contain 0, 1, 2, 3 etc. but may consist of 253, 254, 255, 0 or 254, 255, 0, 1 or 255, 0, 1, 2. It is for this reason that 4x64 integer arrays are stored.

When the receiver process starts up, a period for synchronisation occurs; the first received integer is checked against the first integer in each array. If no match is found, 63 integers are thrown away and the 64th checked. This is necessary in order to prevent buffer overflow while a match is being sought. After synchronisation, the receiver process goes to the checking state, cycling through the chosen array checking off the received integers.

After 256x256 bytes have been received correctly, a message ‘good’ can be sent to the screen (1 per second for correct operation). If an integer is received in error, a message ‘bad’ is sent to the screen and the process is returned to the ‘searching for a match’ state. The ‘searching for a match’ state is also indicated by a message printed out at the same frequency.

The T222 processor merely acts to smooth out transmission to and from the X21 hardware, and the software consisted of two elastic buffers. Decoupling processes are provided on both sides of every interface (see section 5.6.5.1).

The program is able to distinguish between no data received and corrupted data. It can be used to check the correct operation of a transputer and its links, by the addition of
a process to control the transmission of bytes between the buffer processes. This test program was used extensively in this project and was extremely useful in identifying timing difficulties and unclean wave-forms. It was also useful in suggesting suitable software structures for writing data smoothly to the X21 hardware.

5.5.8 Conclusion

The circuitry developed for the two-party wide-band voice system is based on a T222 processor, to guarantee service to the codec. The interface uses the CO11 peripheral chip and associated parallel to serial (and vice versa) conversion circuitry to communicate with the codec.

The circuitry is cheap (low price transputer) and relatively simple, yet capable of communicating at the desired rate. Care was taken to make the circuitry resistant to noise and robust to unclean handshake signals. Attention was paid to the initial conditions, placing the interface in a known state after boot-up. The interface is enabled by writing a byte to the hardware.

Servicing the hardware does not consume processing power, since it is managed by an independent link adapter. Therefore, the requirement for the software design is only to guarantee that the correct servicing process is scheduled on time.

The interface is re-usable for a three party teleconferencing system, since the interface is simple and software driven. The processing power of the interface can be expanded (if necessary) to suit any method of teleconferencing, by the easy addition of more transputers (see section 5.4.2.1).

The interface was thoroughly tested by a program designed to test both the transputer links and the X21 circuitry.
5.6 Concurrent Real-Time Software on the Transputer

5.6.1 Introduction

There are extra stresses placed on the Unison wide-band voice software, compared to other applications. The software must operate in real-time and, more importantly, service the inputs and outputs within stringent timing requirements.

A single transputer has to execute a number of independent parallel processes which need to communicate. Process interaction is based on the principles of communicating sequential processes (see section 5.4.2.2.1), where two processes that wish to converse with one another must wait until both are ready. Within the transputer as a whole, the strict timing requirements may not be met if the processes are not scheduled in the correct order, and the data flow between them is not managed.

5.6.2 Real-Time Concurrent Software Problems

In order to assure reliable real-time concurrent software, there are a number of problems which need to be considered:

- dead-lock
- starvation
- process latency

5.6.2.1 Dead-Lock

Dead-lock is said to occur in a situation where there is no possible recovery, but the concept can be extended to other instances where the system may only be dead-locked on a temporary basis.

5.6.2.1.1 Permanent Dead-Lock

This situation occurs when two communicating processes wish to transfer information, and they both want to output at the same time. Both processes become de-scheduled, and will never be re-scheduled, because they are both waiting for the other process to receive. The standard way of avoiding this situation is to provide buffering on the output of each link, the amount of which depends on the number of outputs that must take place before an input to the same process occurs.
5.6.2.1 Temporary Dead-Lock

This situation occurs when a number of processes are connected to a reluctant end process. Examples of reluctant end processes are a temporarily unavailable separate transputer (as can be the case with the system in question, see section 5.5.2.3), or an end process which is expected to service a number of inputs (multiplexor). The standard solution to this problem is again one of buffering.

5.6.2.2 Starvation

Starvation can occur in multiplexors; where a process has a number of inputs that are always ready, and the selection system is not 'fair'; one input to the multiplexor is always serviced in preference to another. In the Occam language, servicing a number of inputs is achieved by using an ALT or PRI ALT construct. The problem with both the ALT and PRI ALT construct, is that neither of them are fair (Tempelman 91) (see section 5.6.3). The ALT construct was designed to be 'fair', (but is not), and the PRI ALT construct was designed to favour the input which is textually first.

5.6.2.3 Process Latency

Process latency is very important when dealing with synchronous real-time software. Careful attention to program structure and to the performance of constructs used within the program can greatly improve performance. In particular, attention to the operation of multiplexing processes (use of the ALT and PRI ALT construct), and the processes required within the software, can yield time savings.

5.6.3 The ALT Construct

The function of the ALT construct is to multiplex a number of input channels onto one output channel. For each execution of ALT, one of the alternative inputs is selected and its related sub-process is executed. The ALT process will be suspended if none of the inputs are ready, and will only proceed when an input is available. If more than one input is ready, an arbitrary choice should be made as to which one is selected.

Performance data (Inmos 88) does not make it very clear as to the overheads that can be involved in using an ALT construct, so, for clarification purposes, the reference (Tempelman 91) is included here.
Consider:

```plaintext
WHILE TRUE
    ALT i=0 FOR N
    input[i] ? x
    out ! x
```

This is compiled as follows:

<table>
<thead>
<tr>
<th>Clk cycles</th>
<th>Micro-code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ALT</td>
</tr>
<tr>
<td></td>
<td>--enable channels</td>
</tr>
<tr>
<td>5/17</td>
<td>ALTWT</td>
</tr>
<tr>
<td></td>
<td>--disable channels</td>
</tr>
<tr>
<td>4</td>
<td>ALTEND</td>
</tr>
<tr>
<td>4</td>
<td>j again</td>
</tr>
</tbody>
</table>

The body cost of an ALT can either incur 5 cycles if an input is ready, or 17 cycles if none is ready. If none is ready, then another process is scheduled (19 clock cycles).

Consequently, the total body cost of an ALT is as follows:

- **15 clock cycles if a guard is ready**
- **46 clock cycles if no guard is ready**

A guard is an expression which equates to ‘true’ or ‘false’, and it can either allow or disallow an input.

For each input, there is a need for an enable/disable pair:

### ENABLE:

<table>
<thead>
<tr>
<th>Cycles</th>
<th>Micro code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/3</td>
<td>idl channel</td>
</tr>
<tr>
<td>1</td>
<td>ldc 1 --boolean TRUE</td>
</tr>
<tr>
<td>5/7</td>
<td>ENBC</td>
</tr>
</tbody>
</table>

### DISABLE:

<table>
<thead>
<tr>
<th>Cycles</th>
<th>Micro code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/3</td>
<td>dl channel</td>
</tr>
<tr>
<td>1</td>
<td>ldc 1 --boolean TRUE</td>
</tr>
<tr>
<td>2</td>
<td>ldc process.offset</td>
</tr>
<tr>
<td>8</td>
<td>DISC</td>
</tr>
</tbody>
</table>
(The variations depend on a guard being ready and the offset to channel and process). Consequently, for N channels (guards), the ALT cost is as follows:

- $15 + 22N$ under high load
- $46 + 22N$ under low load

The following disadvantages of using an ALT can be derived from the above information:

- The number of clock cycles increases proportionally to the number of alternatives.
- An ALT is not 'fair'.
- For real-time applications, the programmer should try to minimise the number of alternatives, but for flexibility, the number of inputs should be large.

The fact that an ALT is not 'fair' could have serious consequences if the transmitting processes are of a critical nature. Many solutions to these problems have been suggested, ranging from micro-coded quicker versions (Tempelman 91) to the separation of a large ALT into layers of ALTS (for number of input channels > 8) (Shallow 91).

The question of fairness can be addressed in certain situations by judicious use of the PRI ALT construct; but, short of involving the programmer in complicated microcode, the time penalties of using an ALT with a large number of inputs can only be minimised by reducing the number of inputs.

5.6.4 PRI ALT Construct

The PRI ALT construct behaves in the same way as the ALT construct, except when there is more than one guard ready. In this case, the one that is textually first is chosen. However, if there are no guards ready the first time the construct is executed, and both guards then become simultaneously ready, it cannot be guaranteed that the textually first guard will be picked.

5.6.5 Buffer Structure

5.6.5.1 Internal Process De-coupling

Internal channel communication has already been discussed (see section 5.4.2.2.6). When two communicating processes are ready to communicate, the processor copies the message across to the relevant process. Channel communication time therefore increases with message length, and also with the number of processes waiting to be scheduled. Consequently, messages should be kept as small as possible, and should be transferred between a minimum number of processes.
A truly parallel buffering solution was suggested in (May 91), where a buffer is implemented as a pipe-line of parallel buffers of capacity one (figure 48).

![Figure 48: A Truly Parallel Buffering System](image)

The passage of a data packet through such a buffer requires a large number of scheduling instances, and also incurs a large overhead in channel communication. This method of buffering should be avoided in real-time software.

A better solution, using a pool of buffers and a regulator (figure 49) greatly reduces the data latency time through the buffer (Shallow 91). This reduction is achieved at the cost of destroying the parallel nature of the buffer; the buffer can either read in or write out a packet, but it cannot do both in parallel.

![Figure 49: Buffer pool and Regulator](image)

Buffering techniques to minimise data latency and retain true parallelism are extensively discussed in (Shallow 91). In Shallow's investigations further reduction in buffer latency is achieved using shared buffer memory and index passing between processes. The results show significant time savings only when inter-process communication employs messages of greater than 32 bytes.

5.6.5.2 Inter-Transputer Communication De-Coupling

External channel operation has already been discussed (see section 5.4.2.2.6). The communication proceeds under the control of independent link adapters, and does not involve the CPU. The sending and receiving processes in each transputer are descheduled until the communication is complete. The effect on the internal processes of
each transputer must be reduced by providing simple de-coupling processes of one message capacity.

5.6.6 Conclusion

For concurrent software on the transputer to operate successfully within stringent time requirements, a number of problems need to be considered in addition to those normally associated with concurrent software.

Dead-lock can be eliminated by the judicious use of buffering; a pool of buffers and an associated regulator is more favourable than a parallel buffer when operating in real-time on one transputer, but the mechanism is then no longer parallel. Dead-lock occurring between multiple transputers can be alleviated by using simple de-coupling processes of one message capacity.

Starvation can be eliminated using a fairer construct than the transputer's ALT (see section 7.4.3.4.1).

Process latency and consequently the overall performance of a real-time concurrent transputer program benefits from the careful use of Occam constructs.
5.7 Voice Protocol Implementation in Occam - A Parallel Approach

5.7.1 Requirements

The voice protocol software for the Unison wide-band system is executed mainly on the T222 transputer and associated interface circuitry (see section 5.5).

The voice protocol design is based on the delayed-timing method, with voice reconstruction being achieved by packet repetition (see section 5.3).

The software must fulfil the following requirements:
- be reusable in the three-party teleconferencing system
- meet stringent timing requirements
- provide optimum voice quality
- cope with all variations in network loading

5.7.2 Specification

The method used to specify the software design and operation of the Unison wide-band voice system protocol is based on MASCOT 3 (Axford 89).

The following stages are identified by MASCOT:
- external requirements and constraints
- design proposal
- network decomposition
- element decomposition
- program definition

5.7.2.1 External Requirements and Constraints

This section can be further sub-divided into hardware control requirements, software requirements, and system test requirements.

5.7.2.1.1 Hardware Requirements

(see section 5.5)
Wide-band Speech Teleconferencing over an Integrated Network

- The X21 interface has to be initialised.
- The X21 interface has to be read at the correct rate and written to at the correct rate.
- There is a limited amount of 'on-chip' memory available (4 Kbytes) (Inmos 88).
- The interface is initialised by writing one byte out to the hardware.

5.7.2.1.2 Software Constraints

- Inter-Transputer Communication must be decoupled (see section 5.6.5.2).

Due to the servicing of an 'event.in' pin, the large RPC package executing on the T414 CFR interface board must run at high priority. Consequently, there are some 125 second intervals during which the T414 is not available to read data from the T222. This effect must be decoupled from the software used to service the codec. The decoupling can be achieved by providing buffering on the transmit path as well as the receive path.

- The processes that service the codec must be correctly scheduled (see section 5.5.5).

Data transfer to and from the X21 interface is via transputer links. Once a data transfer is initiated, it proceeds under the control of an autonomous link interface, and the sending and receiving processes are descheduled. Consequently, performance considerations are that if data is transmitted to and from the X21 interface in bytes, the sending processes have to be scheduled every 125 seconds (for a codec data rate of 64 Kbps).

If data transfer is achieved using larger data structures (such as the data portion of the CFR slot or 28 bytes), the sending process only has to be scheduled once every 3.5 ms, which is a useful time-saving method.

The disadvantage of this method occurs when the reconstruction buffer is just running empty. In this situation, the hardware requires the block just before it arrives in the buffer, and it is supplied with a dummy one. When the block arrives, it has to be completely thrown away, even though part of its contents have arrived 'in-time' to be played back. Compared with the situation when data transfer is on a per sample basis, the resulting performance degradation is less marked.

The performance quality of the resulting speech when transferring information on a block basis is not expected to be affected by any noticeable amount, since the block size is small enough not to provide much degradation (see sections 3.4.3.1.3, 3.4.4.5), providing this state does not persist for any significant length of time.

The potential processor time savings make this approach more desirable.
5.7.2.1.3 System Test Requirements

The time-critical nature of this system means that careful consideration must be given to the type and amount of information gathered:

- Too much data from a particular process, and channel communication may significantly impact the performance of the critical parts of the software; the transmit and receive buffers and associated pipe-lines.

- Data gathered from too many sources, results in a wealth of processes supporting information exchange, which could well begin to impact the scheduling of more important processes.

The protocol must be able to cope with all variations in network loading. The delayed timing mechanism (see section 5.3.3.2) operates on the receive buffer, to adjust the timing of play-back. Information about the number of blocks in the buffers is therefore required to assess the performance of the protocol.

Heavy network loading is characterised for voice streams by the loss of blocks (see section 5.3.2.1). The number of lost blocks is also essential information.

The gathering of statistics was initially designed to be in response to a command from the keyboard, with limited information being returned.
5.7.2.2 Design Proposal

The normal functions of the voice protocol software identified in the design and software constraints sections are as follows:

- transmit buffer - to allow for the temporary unavailability of the T414 (CFR interface processor)
- sequence number tagging
- sequence number checking
- receive buffer - to smooth out CFR packet arrival times before play-back at a constant fixed rate. This involves an initial delay in play-out.

Abnormal events which require special action are as follows:

- Lost blocks
- repeated blocks (CFR transmitter chip bug!)
- abnormally delayed blocks
- significant changes in the average delay across the network
- clock drift

The above list shows that these elements considered jointly as block loss by Ades (87) can be split into three distinct events:

- actual block loss in the network (as detected by the block sequencing mechanism)
- blocks which are abnormally delayed and will not cause any permanent upset to the operation of the receive buffer
- situations which do cause a permanent upset to the receive buffer play-out mechanism

This design of a parallel implementation of the voice protocol maintains these distinctions, and keeps the three events separate.

The hardware constraints of the system (the requirement to service the codec every 3.5 ms and the fact that there is only 4 Kbytes of memory available for program code and data storage) mean that as many features identified above as possible should be performed by the T414 processor.

The normal functions identified above can be classed according to their nature: functions which need to happen at regular fixed intervals are classed as synchronous, whereas operations which happen or can happen at irregular non-fixed intervals are classed as asynchronous.
The classification is as follows:
- Transmit buffer - synchronous
- Sequence number tagging - asynchronous
- Sequence number checking - asynchronous
- Receive buffer - synchronous

The above list immediately suggests a parallel process structure and system partitioning on the two processors (figure 50).

![Diagram](image-url)

Figure 50: Voice Protocol Design Proposal

The special actions required for abnormal events can now be assigned to processes depending upon how they can be detected.
- Lost blocks and repeated blocks are detected by sequence number tagging at the receiver.

Special action for these events must be taken within the sequence number checking mechanism. (N.B. For the most part, if sequential block loss is limited to one or two blocks, then the receive buffer will proceed as normal. Should more than two blocks be lost, however, the receive buffer will under-run. The buffer should recover as soon as a new block is received. The sequence number mechanism quickly adds the number of lost blocks and the receive buffer once more operates as normal.)
• Abnormally delayed blocks (those arriving outside of the jitter estimate) may cause the receive buffer to under-run, but only temporarily.

No special action is required here.

• Significant changes in the delay across the network and clock drift cause the receive buffer to over-run or under-run permanently.

This can only be detected by the state of the receive buffer and so must be handled by the receive buffer itself (or by nearby parallel processes). Action required in these cases will either involve throwing away excess blocks (if the buffer has filled up) or adding extra blocks if the buffer has permanently under-run.

The approach of separating the actions of detecting lost blocks and detecting buffer under-run due to changing network loading enables what would otherwise be a complex program to be separated into simpler processes. This also means that some of the voice protocol software can reside on the T414 processor.

5.7.2.3 Network Decomposition

The network decomposition is a more detailed design of the software as a picture of all the concurrent processes and their interconnections. The voice protocol network decomposition can be seen in figure 51.

Communication across hard-links (though not under processor control) takes more time than communication across software channels (Inmos 88). As stated previously, once communication is initiated between two processors, the controlling processes are descheduled and the communication takes place under the control of a separate interface controller on the transputer chip. The individual transputer’s controlling process can then only be rescheduled once the interaction has been completed.

For the T222/T414 interface, if the T222 synchronous processes were not isolated from the effects of hard-link communication, the interaction could seriously disrupt codec servicing; the time taken to read or write a block is the sum of the T222 block communication time plus the time for the T414 controlling process to be scheduled to begin the interaction in the first place. Decoupling from the effects of communication can be achieved by the use of single buffers.

The priority mechanism available in Occam needs to be utilised in the T222 concurrent processes. Hard-link communication single buffers should be executed at top priority; as soon as they need to communicate, they should be able to. The other processes within the T222 main pipe-line of processes should execute at low priority to ensure processor time is evenly shared between them.
The gathering of statistics from the voice protocol processes could be achieved by the use of variant tag protocols (Inmos 88) on the hard links between the T222 and T414. This approach, however, increases the communication time for each packet across the T222 to T414 interface boundary. A far less obtrusive solution is to make use of a separate hard link between the T414 and T222, since one is available. Results are collected and relayed to the monitor interface, developed as part of the hardware test program (see section 5.5.7).
The use of buffers on the information outputs of all the statistics-reporting main pipe-line processes ensures that the scheduling difficulties of the T414 and any possible starvation (see section 5.6.2.2) introduced because of multiplexing in the statistics sub-system (using the ALT construct) does not 'back-up' the results to the main processes (temporary dead-lock, see section 5.6.2.1.2); statistics messages are removed from the main pipe-lines as soon as possible.

The above mechanism ensures that the monitoring system infringes as little as possible upon the critical timing of the processes in the main data thoroughfare.

5.7.2.4 Element Decomposition

5.7.2.4.1 Statistics Gathering Sub-system

The statistics gathering sub-system is designed to respond to an inquiry from the keyboard, although this can easily be altered to some other method, such as a timer. The response to the inquiry is in the form of two byte messages, the first byte identifying which parameter is being reported, and the second byte the actual value of the parameter. A two byte message was chosen in order to minimise the processor time required to transfer the message between processes, and also to reduce the amount of storage required. The best times to check for an inquiry and reply back are as soon as a block has been read into the transmit buffer from the codec, and as soon as a block has been transferred to the codec from the receive buffer. This policy ensures that the generation of statistics infringes as little as possible upon the time-critical nature of the transmit and receive buffers, and that the true state of the buffers is identifiable.

5.7.2.4.2 Sequence Number Tagging and Checking Processes

The sequence number (16 bits) is added into the CFR packet header (see section 2.6.1). The sequence number checking mechanism detects repeated and lost blocks. Repeated blocks are thrown away in this process. Lost blocks are added in and sent onwards down the receive pipe-line. Up to four lost blocks are replaced by the previously received block, and any more are replaced by blocks full of zero amplitude samples (see section 5.3.3.3).

If more than one block has been lost, the receive buffer will under-run and will replace blocks (as in the sequence number checking mechanism). Assessment of the performance of the Unison network, however, shows that more than one block being lost is unusual (see section 5.3.2.1.4).
5.7.2.4.3 Transmit Buffer

The function of this buffer is solely to overcome the difficulties of communicating with the T414; it is not normally needed in voice protocol software. The structure of the buffer is one of a bounded buffer in conjunction with a regulator (see section 5.6.5.1).

For the voice protocol software, shared buffer memory and index passing (Shallow 91) as a means of reducing process latency (see section 5.6.2.3) was thought to be unnecessary. This decision is based on the observation that time-savings for this type of buffer structure are only significant for messages of length greater than 32 bytes, and the voice protocol software always uses messages and data blocks of less than 32 bytes (2 bytes and 28 bytes respectively).

Other buffering structures that introduce true parallelism into the buffer operation are also not required here; what is needed is a bias of servicing preference towards the synchronous end of the system (the X21 interface), both in the transmit and receive buffers and in the message buffers.

5.7.2.4.4 The Receive Buffer

The receive buffer is again a bounded buffer in conjunction with a regulator, with servicing preference towards the codec. As with the transmit buffer, statistics are only considered after communication with the codec has occurred.

As identified in the design considerations, the jitter assessment is set at 1 CFR packet (reconstruction delay set at 2 CFR packets), with abnormally delayed blocks being replaced by the previous packet (see sections 5.3.3.2, 5.3.3.3).

For conditions where the receive buffer under-runs by up to 14 ms, the last packet to be received and played out within its correct time interval is repeated. This limit is seen as the maximum acceptable speech loss before speech characteristics change significantly (see section 5.3.3.3). Further under-run of the buffer causes replacement by zero amplitude packets.

5.7.2.4.5 Recovery Procedures for Abnormal Events occurring in the Receive Buffer

The actions detailed above describe the normal situation.
Some form of recovery procedure needs to be executed in the following abnormal cases:

- if the transit delay across the network significantly alters to that experienced by the very first received packet
- if the local clocks drift significantly over the duration of the call

As discussed in design considerations (see section 5.3.3.2), adjustment needs to be made only very infrequently. This is because of very high stability crystal oscillators, and also because transit delays across the network are not expected to vary greatly. Since adjustment is only required infrequently, buffer alteration can be achieved in a relatively clumsy way (by simply adding or losing blocks, rather than by expanding or decreasing silent intervals, as described in Ades (87), and the temporary reduction in quality will go unnoticed.

The abnormal events detailed above may result in one of two persistent states:

- **Full Buffer**
  This situation is easy to detect and easy to remedy. A full buffer can only be the result of either clock drift adjustment or of a large change in the conditions of the network. Consequently, as soon as this condition is detected, it is safe to throw away every other block until the buffer is half-full once more. Every other block is thrown away to minimise the effect of the adjustment on the speech quality (see section 3.4.4.5).

- **Empty Buffer**
  This condition is far more difficult to detect than a full buffer, since there are instances when the buffer can empty (and even significantly under-run) for quite large intervals of time and still recover.

The persistent states occur in the following situations:

- if a large number of blocks is lost across the network
  This is the responsibility of the sequence number tracking process. If, for example, ten blocks are lost, then the buffer will significantly under-run and play silence. Upon receipt of a new block of data, the sequence number manager will discover the loss and transfer ten dummy blocks to the receive buffer followed by the new block.

- if a block or a few blocks are delayed by an abnormally long time
  This can occur if a CFR interface hangs, or if communication with a network management machine is requested (see 2.6.4.4). When this happens, the CFR interface becomes ‘busy’ and the delay momentarily increases. As above, no action is required.

The solution is to start a monitor upon detection of an empty buffer. After a reasonable amount of time the buffer is examined again and, if it is still empty, recovery procedures can be initiated: the buffer is reset and pre-filled to half-full before being allowed to free-run again.

The definition of a ‘reasonable amount of time’ depends upon the maximum significant number of blocks that can be lost and the maximum amount of jitter expected.
(this is different to the initial jitter estimate, which is an average). The value must not be too short, as this could lead to action being taken for one of the cases identified above, and empty-buffer recovery action being followed by full-buffer recovery action. The value must also not be too long, or significant portions of speech may be lost before action is initiated.

For transmission across the Unison WAN network, a ‘reasonable amount of time’ can be set at 40 ms, since it is possible for 10 blocks to be lost consecutively during this period (Siddiqui 89).

The empty buffer monitoring function is best implemented as a separate process.

5.7.3 System Testing

The testing of the voice protocol software can be split into two stages:

- operation verification
- software latency

5.7.3.1 Operation Verification

Each of the processes in isolation and the process pairs (transmit and receive) was tested to ensure basic program operation. The voice protocol software (without the sequence number processes) was then coupled with the X21 hardware to test normal and abnormal conditions. The software for voice protocol testing of abnormal conditions can be seen in figure 52.
The testing process could perform the following functions:

- **delay blocks to test buffer under-run, test correct voice reconstruction and test correct system recovery, without the empty monitor process resetting the receive buffer**
- **lose blocks to test the empty monitor**
- **repeat blocks to test over-run (at 'number of blocks = 4', throw away until number of blocks = 1)**

The system was first tested with the buffers operating without the sequence number tagging and checking mechanisms. The purpose of this test was to show how the receive buffer behaved when the network loading characteristics changed: the delay between the transmitter and receiver increases (or decreases) and remains at this new level. In this situation, this effect can be created by losing blocks.
Statistics regarding the number of 'repeated' blocks, the number of repeated zero amplitude blocks (or 'silence' blocks), and the number of buffer resets were recorded. The inputs to the system include both valid inputs (0, 1, and 2 blocks lost, or 0, 3.5 and 7 ms worth of voice) and invalid inputs (10, 30 and 60 blocks lost, or 35, 105, 210 ms worth of voice). The invalid inputs will never occur in the Unison network, since the maximum variation in delay for voice connections is 8 ms (see section 5.3.2.2.3).

<table>
<thead>
<tr>
<th>Number of Blocks Lost at one instance</th>
<th>Total Number of Blocks Lost</th>
<th>Receive Buffer Length+Block in Regulator</th>
<th>Number of Repeated Blocks</th>
<th>Number of 'Silence' Blocks</th>
<th>Number of Buffer Resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>44</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>104</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Results for the Receive Buffer and an Increase in Delay

The results show that the buffer can take a delay variation of 3.5 ms, with no resulting action. An increase in delay of 7 ms causes the adaptive timing mechanism to adjust the timing of the buffer (the buffer is reset). The empty monitor in this example waits for 24 ms to evaluate the state of the receive buffer. This value was chosen to investigate the pattern of repeated as opposed to 'silence' blocks. The results show that 4 blocks were repeated, and 3 blocks were filled with zero amplitude samples. This means that the voice reconstruction design (repeat up to 4 blocks and then fill with zero amplitude samples) (see section 5.3.3.3) has been correctly implemented. The timing adjustment could be heard occasionally as a single 'click'. In the Unison network, the empty monitor timing period can be adjusted to 16 ms or less, since the maximum variation in delay as a result of network loading changes is 8 ms. The perception of adjustment in this situation is almost imperceptible.

The figures in table 2 show no difference in the repeated and silence numbers for block loss values of 2 or more. This is because a reset causes the buffer to wait for a new block to be received before it starts counting again.
The value of blocks in the receive buffer is a snap-shot just after a block has been written out to the codec; a value of zero in the receive buffer does not mean that the codec is playing out receive blocks. The start-up mechanism in the buffer before starting initial play-back (after the first block has been received, wait 3.5 ms to facilitate a 7 ms reconstruction delay) means that in most cases a block has also just been received by the buffer.

The second test was devised to investigate how the buffer behaved when the delay reduced in size. This resulted in the buffer filling up.

<table>
<thead>
<tr>
<th>Number of Blocks Added at One Instance</th>
<th>Total Number of Blocks Added</th>
<th>Receive Buffer Length +Block in Regulator</th>
<th>Number of Repeated Blocks</th>
<th>Number of ‘Silence’ Blocks</th>
<th>Number of Buffer Resets</th>
<th>Number of Blocks Thrown Away</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3: Results for the Receive Buffer, and a Decrease in Delay

The results in table 3 show that the buffer can take a delay decrease of 3.5 ms with no resulting action. A delay decrease of 7 ms causes the extra two blocks to be thrown away. This alteration was imperceptible.

The sequence number processes were then added back in, and further tests performed.
Table 4: Behaviour of the Voice Protocol Software

<table>
<thead>
<tr>
<th>Number Blocks Added (a), Lost (l) or Delayed (d)</th>
<th>Receive Buffer Length+</th>
<th>Number of Repeated Blocks</th>
<th>Number of ‘Silence’ Blocks</th>
<th>Number of Buffer Resets</th>
<th>Lost Blocks identified (Sequence Manager)</th>
<th>Repeated Blocks identified (Sequence Manager)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a 1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>a10</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>d 10</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 10</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>1 11</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>0</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>1 12</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>1 14</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>0</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>1 15</td>
<td>2</td>
<td>4</td>
<td>11</td>
<td>1</td>
<td>15</td>
<td>0</td>
</tr>
</tbody>
</table>

The results in table 4 were collected when the empty monitor evaluation time was set at 44 ms, in order to investigate the behaviour of the sequence manager and the receive buffer combined. The results show the following behaviour:

- abnormally delayed blocks do not cause any action by either the sequence manager or the receive buffer, as long as less than 44 ms of delay is experienced
- lost blocks are correctly detected by the sequence manager and replaced, as long as less than 44 ms worth of speech has been lost
- blocks that have been repeated (see section 5.7.2.2) are correctly identified by the sequence manager and removed, with no effect on the receive buffer

The handshake signals (see sections 5.5.3.2, 5.5.3.3) were used to determine correct operation within the time constraints. The statistics sub-system did not alter the duration of the handshake signals. This indicates the absence of any temporary dead-lock, and shows that the statistics reporting mechanism did not affect the real-time performance of the system.

5.7.3.2 Process Delay

The delay produced by a process is difficult to measure exactly. The most important part of the voice protocol software is the receive buffer. The transmit buffer will have a smaller value for delay, and so is not measured. The receive buffer delay was measured under normal operating conditions; the empty buffer process will not be activated, and consequently will not contribute to the results. The software structure can be seen in figure 53.
The results-multiplexing process will not contribute to the value, since it does not execute during the measurement. The only extra software that is executed with the process under test is the process that times writing out and receiving 500 blocks. The test process executes at high priority, so that the accurate 1 microsecond timer can be used. The process produced delay was found to be 97.8 microseconds per block; results in Shallow (91) for a 32-bit transputer give 50 microseconds per block. The process delay found in this test is comparable to that evaluated by Shallow (91), given that a 16-bit transputer was used, and the software was far more complex.

These results (handshake signals and process delay measurement) indicate that the software did not suffer from temporary dead-lock, or starvation.

5.7.4 Conclusion

The design was well structured. Problems such as starvation and dead-lock (both temporary and permanent), were eliminated by the judicious use of buffering and careful consideration of multiplexing techniques. The statistics sub-system was structured to impinge as little as possible upon the main pipe-line processes.

The protocol was tested during both normal and abnormal operating conditions, and found to cope with all variations in network loading. The software met the stringent timing requirements expected of it, and exhibited roughly the desired process delay (only slightly higher value was measured, despite much extra functionality).
5.8 Integrating the Wide-band Speech System into the Unison Office

5.8.1 Requirements

The requirements of the whole Unison integrated office (see section 2.2.2) are comprised of the following:

- **cost effectiveness and the potential for growth**
  The Unison office is a loosely-coupled system, where component control is essentially distributed (see section 2.2.3.2). This makes the office cost-effective, since resources can be shared. Distributed, loosely-coupled systems have a large potential for growth, since system expansion involves simply adding more application equipment to the LAN.

- **the capability of orchestrating the office in a simple and friendly manner**
  The hub of control for the user is the workstation. All applications equipment can be set-up easily through a simple and friendly menu-driven user interface.

- **reliability**
  The desire to control all the applications equipment from the workstation must be off-set by the need for reliability. It is necessary for the wide-band speech system to maintain the capability of independent operation. Consequently, the wide-band speech system must have two points of control: at the workstation and at the external keypad, which is directly connected to the wide-band speech system.

5.8.2 Distributed Control Implementation

The basis of the Unison integrated office distributed control is the RPC mechanism (see section 2.6.2.3). This was developed at RAL for all transputer-based applications equipment. The LCD keypad and associated controlling software was also developed at RAL.

Workstation control was implemented as a result of messaging directly from the workstation to the wide-band speech system. Workstation software modifications were done by LUT (Murphy 90). RPC and secretary process alterations were done by RAL. The state machine and LCD message display software modifications were done by the author.

5.8.2.1 Call Set-Up

Call set-up is achieved by the workstation sending an RPC message directly to the local wide-band speech system. The RPC message contains a function code and the ‘dialled number’ in string form. The wide-band speech connection is then initiated in the same way as for LCD pad control; by the application itself, without any further help from the workstation.
5.8.2.2 Call Clear-Down

To clear the call down, the workstation again sends an RPC message directly to the local wide-band speech system. The message contains a function code which means 'stop'. The remote system is then passed an 'in-band' function code in a CFR slot to terminate the call.

5.8.2.3 Ringing Control

The state machine in each wide-band speech system needs to be modified when the call is being set-up under workstation control. The modified state machine can be seen in figure 54.

A parameter signifying 'ringing control' is passed to the local wide-band speech system and from there to the remote wide-band speech system. All tone-related states are by-passed; instead, messages are returned to the workstation via the RPC mechanism. Abnormal situations (e.g. engaged) are also indicated to the workstation via RPC messages. At the end of call clear-down, the state of the local and remote wide-band speech systems changes directly to the 'on-hook' state without waiting for a keypad command to go 'on-hook'. This is also true for the local wide-band speech system in abnormal situations.
5.8.3 Office Integration

The integration of the wide-band speech into the Unison office was complemented by workstation video control (Lu 90). Workstation facilities such as the office manager (directory service) and user mobility were implemented at LUT (Murphy 90).

5.8.4 Conclusion

Integrating the wide-band speech system into the Unison office is cost effective and provides the potential for growth, due to the distributed nature of the office implementation and control. System growth by the addition of more components is substantially eased by the dynamic binding of office components at the start of the day and office manager control of such components.

Office orchestration can be achieved entirely by using the workstation as the 'hub' of control; complex services, such as multi-media teleconferences can be managed in a simple way (Murphy 90).

The addition of a separate point of control in the Unison wide-band speech system provides a means of communication should the workstation fail, and builds the necessary amount of reliability into the office.
5.9 Conclusion

This chapter has described the work undertaken to implement the two-party wide-band speech teleconferencing system in the Unison project.

The aims of the two-party wide-band speech teleconferencing system have all been successfully met:

- a robust system within the Unison network
- optimum voice performance at a reasonable cost

The audio system was designed to be easy to set-up, and easy to use. No speaker training is necessary to successfully use this teleconferencing system. Excellent speech quality was produced by the system, and no echoes were perceptible. The good echo performance was achieved without voice switching, which is commonly used in teleconferencing systems today.

The cost of the audio system was kept to a minimal level. Frequency shifters were included in the implementation, but the cost is relatively low with substantial performance benefits being gained.

The voice protocol design was fitted to the Unison network performance characteristics. The design was successful in minimising the end-to-end delay experienced by the system, which reduces the perception of echoes. The system copes with all expected mixtures of traffic in the Unison network.

The hardware designed and built for this system will be re-usable in the three party teleconferencing system. The careful design and testing of the hardware resulted in circuitry that was robust to spurious signals. Use of a 16-bit version of the transputer and only ‘on-chip’ memory resulted in a low cost interface.

The transputer and the Occam language were investigated as a basis for embedded system control. The pit-falls of concurrent real-time software on the transputer were identified and suitable solutions proposed.

The voice protocol implementation was well structured and carefully designed to maximise the strengths of the transputer. The end-to-end delay of the system and optimum voice quality requirements were of primary importance in the implementation. The design was thoroughly tested from a behavioural point of view and the implementation was evaluated as a contributor to the end-to-end delay of the system. The required determinism in servicing the codec was successfully introduced. A comprehensive statistical information reporting system was implemented which did not
affect the state of the system that was being measured. The reporting mechanism proved invaluable in assessing the performance of the system.

Integration into the Unison office was successfully achieved in conjunction with other office components. Call control was shown to be easy to implement. This was a direct result of the 'intelligent' services offered by the Unison network.

The overall performance of the Unison two-party wide-band speech system proved that the Unison network was truly integrated; it could carry voice and data on the same network, while providing the characteristics necessary for both types of traffic.
Chapter 6
Stream Mixing for SB-ADPCM 7 kHz Voice

6.1 Introduction

This chapter describes the basis of the three-party teleconferencing system; the principles and implementation of a system that will mix three streams of digitised voice in real-time.

The aims of this chapter are to investigate the basis of a teleconferencing system that will fulfil the following requirements:

- *basis of a three-party teleconferencing system for the Unison integrated office*
- *not over burden the network, in terms of bandwidth requirements*
- *provide optimum speech quality*
- *be robust with respect to block loss*
- *not result in excessive end-to-end delay*

The first part of this chapter investigates the possible options for providing a multi-party teleconferencing system using SB-ADPCM. Existing research is based on PCM. The possible methods identified, and their applicability to SB-ADPCM systems are discussed.

The second part of this chapter describes the possible codecs available at this stage of the project. The advantages and disadvantages of each are investigated. The choice of processors within the multi-party system is also considered.

The third part of this chapter details the equipment that was built to verify correct system operation.

The fourth part of this chapter gives some background to the principles of finite-length arithmetic in digital signal processing implementations.

The fifth part of this chapter describes the existing codec's implementation of finite length arithmetic, and the methods used to match the performance of the two micro-processors. The operation of the individual parts of the SB-ADPCM algorithm (see section 3.3.4.3) are investigated briefly. This work gives some idea of what to expect in the finished system operation.

The sixth part of this chapter looks at the transputer implementation of the software. The techniques involved in mixing three streams of speech are described.
6.2 Three-party Teleconferencing Systems and SB-ADPCM

6.2.1 Requirements

This section looks at the possible methods of mixing voice streams. To date, research for systems that operate across local area networks is based on PCM. These methods are described, together with each mechanism's applicability to SB-ADPCM.

A mechanism that will provide three party teleconferencing across the Unison network should meet the following requirements:

- not over-burden the network
- provide optimum voice quality
- be enhancable, to improve the mixing quality for example
- be available at a reasonable cost

6.2.2 Stream Mixing

The traditional method of combining voice streams is by mixing at the analogue stage. Here, each wide-band speech system would require one codec encoders and \(n-1\) codec decoders for an \(n\) way teleconference (figure 55).

![Figure 55: Network Bandwidth Required for Analogue Mixing](image)
Analogue mixing has a number of problems associated with it:

- It places an excessive burden on the network interface: $2(n-1)$.
- It places an excessive burden on the network: $n/2(n-1)$ streams.
- It is expensive (resources cannot be shared).
- It is difficult to extend.

Digital mixing alleviates the problems:

- It places less of a burden on the network.
- It is cheaper.
- It is easier to extend.

6.2.3 Digital Teleconferencing Systems Using PCM Coded Speech

Mixing PCM voice streams is reasonably straightforward, since the operations involved are those of addition and subtraction using look-up tables (Ades 87).

PCM teleconferencing systems that operate across LANs can consequently be implemented in a number of different ways:

- multi-cast
- voice packet shuttling
- bridging

6.2.3.1 Multi-Cast

This method relies on a multi-cast service being available on the network (Weiss 88). The participating users transmit voice packets to the multi-cast service, which generates $n-1$ replicas of each stream (where $n$ is the number of conference participants). Each station receives a voice stream for every other participant, and can combine them in the desired way.

The multi-cast method places quite a heavy burden on the network in terms of required bandwidth (figure 56).
A further burden is placed upon each participating interface, since each interface (while it only has to transmit one stream) must be capable of receiving and combining \( n-1 \) streams. This burden necessarily involves extra specialised equipment (or software) for teleconference situations, and means that the system is not easy to extend. This is in contrast to the requirements for an integrated office (see section 2.2.2), where the considerations of cost effectiveness and potential growth cite that multi-media equipment should be kept as simple as possible.

### 6.2.3.2 Voice Packet Shuttling

In this method, a single packet containing a packet worth of PCM speech contributions from all participating users is shuttled round the network (Weiss 88). As a station receives the shuttle packet, it removes its own previous contribution and uses the resulting packet for play-back. The station then adds a new contribution to the shuttle packet and sends it on its way to the next user.

Implicit in this design is the restriction that the shuttle packet must be relayed to all conference participants within the play-back duration of one packet. This design can be extended to multi-LAN networks, although it involves processing at gateways (Weiss 88). The software of the participants also has to be upgraded to operate in teleconferencing applications, which again contrasts to the requirements of cost-effectiveness and potential growth (see section 2.2.2).
This method also has a further disadvantage: conference set-up is not simple; each station must know the address of the next station on the LAN. Conference participants who wish to drop out of a conference, or those who wish to join an existing conference, will destroy the initial set-up plans, and a new 'next station' address will have to be transmitted to each participator by the controlling station (Weiss 88). This method is therefore rather complex to orchestrate.

The advantage of the voice packet shuttling method is that the burden on the network is very light, since only one packet is circulating around the network.

6.2.3.3 Bridging

This is the method generally used for teleconferencing. It involves the provision of a separate service (usually on a separate machine), whose sole purpose is to receive voice packets from every participant and to combine them in the appropriate manner. Each recipient receives a combination packet representing the contributions from the other \(n-I\) participants.

This method does not place an unnecessary burden on the network \(2n\) streams (figure 57), gives cost effectiveness and the potential for growth.

Figure 57: Network Bandwidth Requirements for Bridge Teleconferencing
Optimum voice performance is achieved by the bridging mechanism, since the increase in end-to-end delay is not excessive compared to the voice shuttling method. The set-up and orchestration of this method of teleconferencing is easy; each participant receives the address of the bridge station to transmit to and receive packets from, and conference participants that wish to join into or drop out of the conference do not have any effect on other participants (the bridge deals with this).

6.2.4 Digital Teleconferencing Systems Using ADPCM Encoded Speech

The mixing of streams of ADPCM and SB-ADPCM encoded speech is more complex than that required for PCM; each stream has to be decoded before mixing can take place.

The multi-cast method (above) requires the provision of n-1 decoders, which is a significant extra burden on the controlling processor (it will require more than one processor to accomplish this task). The multi-cast method is therefore not favoured for ADPCM and SB-ADPCM encoded speech.

The voice packet shuttling method is completely unsuitable for more complex coding algorithms, since the extra delay incurred in decoding the stream would mean that the shuttle packet can not reach every participator during the play-back of one sample.

The bridging method is the mechanism which satisfies the requirements for Unison teleconferencing involving more complex speech coding algorithms.

6.2.5 Conclusion

The multi-cast and the voice packet shuttling methods are not suitable for SB-ADPCM. The requirements set out at the start of this section are best met by the bridging method: not over-burdening a network, optimum voice quality, reasonable cost, and enhancement capability.
6.3 Equipment Available and Processor Choice

6.3.1 Requirements

This section investigates the 7 kHz audio codecs available for a three-party teleconferencing system. The choice of a suitable processor on which to implement the speech coding algorithm is also described.

The three-party teleconferencing system algorithm, and the processor(s) on which it runs, must meet the following requirements:

- robust in the presence of block loss
- cheap to develop

6.3.2 Equipment Available

There were two similar codecs available at this stage of the project:

- 7 kHz Prototype Codec
- G.722 Standard Codec

6.3.2.1 7 kHz Prototype Codec

The 7 kHz prototype codec was used for the two-party teleconferencing system (chapter 5). The main features of this coding algorithm have been described earlier (see section 3.3.4.3).

The non-standard SB-ADPCM algorithm is robust to block loss (see section 3.4.4.6). The disadvantage of using this type of codec was that it was merely a prototype; no more were obtainable. Fortunately, there were already three on loan to the Unison project.

The algorithm used by this codec was not a CCITT standard, and so exact details of the speech coding mechanism were not available in the manuals (Lee 84, CCITT SG XVIII).

6.3.2.2 G.722 Standard Codec

This codec conformed to the standard CCITT specification G.722 (CCITT G.722). The main features of this coding algorithm are described in section 3.3.4.3.
6.3.3 Choice of Codec

The G.722 standard algorithm has been exhaustively documented, and so would be easier to implement on a new processor. Because of its use of an adaptive predictor, the G.722 coding algorithm gives better quality speech than the prototype. The G.722 standard coding algorithm is more robust to bit errors than the prototype codec, because the standard assigns high-density code words (large proportion of ‘ones’) to low amplitudes (CCITT G.722). The standard, however, is not as robust to block loss as the non-standard codec (Choi 89); the voice reconstruction mechanism used for the two-party system (packet repetition) will not work as well with the standard codec, due to the adaptive predictor used in this algorithm. The more complex coding algorithm requires more processing power than the non-standard codec.

The non-standard codec was chosen for use in the Unison three-party teleconferencing system for the following reasons:

- less complex coding algorithm

  The non-standard coding algorithm is likely to consume less processor power (this algorithm required only one processor for the encoder, whereas the standard was expected to require two processors). The choice of the non-standard codec meant a cheaper teleconferencing bridge.

- better performance during block loss possible

  Packet repetition does not work as well with the G.722 standard codec as it does with the non-standard coding algorithm (Choi 89).

6.3.4 Processor Choice

The SB-ADPCM algorithm (see section 3.3.4.3) is inherently sequential, which suggests implementation on a specialised DSP chip, similar to that used in the non-standard codecs. This idea was rejected, however, as extra hardware would be necessary to connect a 'standard' processor into the existing transputer based CFR interface.

The multi-processor architecture expected in the teleconferencing bridge suggests a parallel processor, such as the transputer. The high-level language for the transputer (Occam, see section 5.4.2.2.1) means that implementation of the algorithm is likely to be easier than the assembly language required for Digital Signal Processing (DSP) chip programming.

A bridge system implemented on a network of transputers is easily expandable. This means that not only can the system be readily extended to cater for more than three parties engaged in a teleconference, but that there is scope for other features, such as digital audio processing techniques to be incorporated into the design.
The overview of the transputer (see section 5.4) highlighted that the transputer family includes different bit-size processors, all of which can easily communicate with one another. This means that individual transputers within a network can be 'tailored' to suit the processor power needs of different parts of the bridge; more processor intensive parts can be executed on high speed 32-bit processors (the codec encoder and decoder), while less processor-intensive functionality can be run on slower, 16-bit processors (see section 7.3.2). This capability represents savings in bridge hardware cost.

6.3.5 Conclusion

The non-standard codecs were chosen for the Unison three-party teleconferencing system. While exact details of the algorithm were not available, development was expected to be easier due to the simpler structure of the mechanism. (This turned out to be very short-sighted!)

The non-standard algorithm is robust against block loss, and was expected to be cheaper to implement than the G.722 standard.

The transputer was chosen as the basis for the teleconferencing bridge implementation for the following reasons:

- ease of connection into the existing applications CFR interface
- scope to incorporate other features into the design
- the possible cost savings within a network of transputers
6.4 System Development Considerations

6.4.1 Requirements

This section investigates the equipment used to aid the implementation of the non-standard algorithm on the transputer. The text also describes the circuitry that was built to verify correct system implementation and to investigate possible methods of voice mixing.

6.4.1.1 System Requirements

A teleconferencing bridge has the following requirements:

- **low cost**
  Within a SB-ADPCM teleconferencing bridge, the most processor-intensive parts of the software are the encoder and decoder parts, one pair for each party. In order to keep the system costs down the algorithm should operate on 1 or 2 processors.

- **real-time implementation**
  The speed of the processor(s) and the program design should be such that the system can operate in real-time. The crucial parts of the software are again the encoder and decoder.

- **optimum voice quality**
  The behaviour of the coding algorithm must exactly match that of the non-standard codecs.

6.4.1.2 Development Requirements

The development approach required the following criteria:

- **ease of development**
  Developing an implementation of the algorithm that exactly matches the behaviour of the non-standard codecs is not trivial. Easy to use graphics and debugging facilities are essential.

- **verification of correct algorithm operation**
  The algorithm must exactly match that of the non-standard codecs at all frequencies and amplitudes. This leads to a large number of possible test cases. It is therefore necessary to design a verification test plan that adequately covers system operation using minimal test cases.

6.4.2 Development

6.4.2.1 Simulation

Since access to the un-coded digital version of the input wave-form of the codec (or to any intermediate versions) is not possible, development had to proceed on a simulation basis. The transputer, while being the eventual vehicle for the algorithm, does not have a very user-friendly interface, and a graphics interface is not available without
substantial development. The initial part of the development was therefore accomplished by simulation in a high-level language on an IBM PC (or compatible machine).

6.4.2.2 Algorithm Verification

Simulation of an encoder/decoder pair is not sufficient to ensure correct algorithm operation; whatever the coding action, so long as the decoder is the exact inverse of the encoder, the input and output wave-forms will match. Correct implementation of the algorithm means that the implemented decoder must be the inverse of the non-standard codec encoder and vice versa. This was achieved by feeding an analogue sine-wave into the codec, and recording the output code words for a reasonable interval. The frequency and the amplitude were varied (figure 58).

![Figure 58: Algorithm Verification](image)

Algorithm verification is normally achieved by using a program which exercises the codec over the poles of the predictor (CCITT G.722). The testing program is usually detailed within the specification. As such a program was not available for the non-standard codecs, and was deemed to involve a lot of investigation/development for little benefit, this was not done. (This omission is reasonable, since access to a 'similar' version of the non-standard encoder and decoder programs was available).

6.4.2.3 Real-Time Verification

Real-time verification was achieved by timing the transputer implementation of the encoder (since this is the more processor intensive of the two), and inserting the transputer implementations between the non-standard encoder and decoder (figure 59).
6.4.2.4 Mixing Verification

This was accomplished by simulation and by building a circuit solely for mixing verification (figures 60, 61).

Figure 60: Mixing Test Circuitry

The circuit inputs two streams of code-words and decodes them. The mixing transputer combines the two inputs to form a composite stream, which is then re-encoded.
6.4.3 Conclusion

The development requirements (ease of algorithm development, and verification of correct algorithm operation) are met by the plan described in this section.

Simulation on a PC, to identify programming errors, is chosen for the first stage in the implementation of the non-standard 7 kHz coding algorithm. The coding scheme implementation in Occam is then to be verified using the non-standard codecs to identify any logical errors. The implementation’s operation in real-time is then to be checked by timing the encoder.

The simulation studies are used to investigate efficient algorithm implementation in order to reduce costs. The plan also incorporates scope to verify the real-time operation. There are no plans to measure the SNR of the implementation, as this will be the same as that of the non-standard codecs.

The development methods described above are less than ideal, due to the lack of access to the internal operation of the non-standard codecs. Consequently, development is likely to be difficult and time-consuming. The development plan chosen is the best possible within the circumstances.
6.5 The Effect of Finite Word Length Arithmetic in Digital Systems

6.5.1 Introduction

Practical implementations of digital systems are limited by the accuracy with which they can represent a value. This is due to the word length of the digital signal processor (DSP). A study of the effects of finite word length arithmetic is necessary when designing and implementing digital systems, as it is important to know the lowest possible number of bits required to accurately represent a quantity without introducing unacceptable errors.

Following on from this, when a second system (based on a different processor) is being designed to perform exactly as the first, information about the type of errors introduced by finite word length arithmetic in the first system is required (Le Tourneur 86).

6.5.2 Number Representations

The three most common binary number representations, in systems which do not have floating point capabilities, are as follows:

- sign and magnitude
- one's complement
- two's complement

Arithmetic performed using any of the above representations gives the same answers except where quantization and overflow operations are involved.

6.5.2.1 Quantization

This is the mechanism where a value \( x \) is replaced by another value \( x_q \), which is nearly equal to \( x \) but can assume fewer different values when replacing a real number by the nearest integer.
There are three different quantization characteristics available (figures 62, 63, 64):

- rounding
- value truncation
- magnitude truncation

Figure 62: The Rounding Quantization Characteristic

Figure 63: The Value Truncation Characteristic

Figure 64: The Magnitude Truncation Characteristic
Different number representations have different quantization characteristics. Care must be taken when implementing a system on a new microprocessor that must be compatible with an existing system. The quantization characteristic used must be the same for both systems. This rationale prevents the occurrence of cumulative errors within the composite system.

6.5.2.2 Overflow

Overflow is another non-linearity that can occur in digital systems. It occurs when a variable tries to assume a value outside the maximum range describable by the word length of the system. This gives rise to three possible overflow characteristics (figures 65, 66, 67):

- saturation
- zeroing
- 'saw-tooth' overflow

![Figure 65: The Saturation Overflow Characteristic (Van Den Enden 89)]

![Figure 66: The Zeroing Overflow Characteristic (Van Den Enden 89)]
Microprocessors usually use the 'saw-tooth' method of overflow, since this is the easiest to implement, and it preserves the differences. High-level language implementations detect overflow when it happens, and produce a system error.

As with quantization (above), when implementing a digital system on a new processor that has to be compatible with the existing system, care must be taken to ensure the overflow characteristic used is the same.

6.5.3 Quantization and Overflow Effects in Digital Filters

This section forms a basis for determining what effect finite number length arithmetic has on a SB-ADPCM system. The text is only brief, as a greater understanding of these effects is required only when designing a new implementation of a digital system. The background material was useful in this research when interpreting errors caused by differences in the implementation of the SB-ADPCM system on a transputer network of processors, compared to the implementation of the non-standard codecs on a DSP chip.

6.5.3.1 Quantization of Filter Coefficients

Design methods for determining the filter coefficients of a digital filter yield very accurate results. The implementation of a digital filter using finite length arithmetic means that the filter coefficients have to be quantized. Unfortunately, this results in a change of the frequency response of the filter; the positions of the poles and zeros are altered. Depending on the structure of the filter, these changes can be quite considerable, sometimes resulting in a stable filter changing into an unstable one (Van Den Enden 89). Consequently, when implementing an existing algorithm on a new processor, the quantization level of the filter coefficients must be maintained.
6.5.3.2 Limitation of the Word Length of Intermediate Results

The consequences of limiting the word length of intermediate results in a digital system can be both complicated and disastrous. Intermediate results are formed as the outcome of addition or multiplication somewhere inside the implementation of a digital system.

In general, the addition or multiplication of two B-bit words leads to an increase in the number of bits required to store the result:

- Addition gives B+1 bits.
- Multiplication gives 2B-1 bits.

In digital filters, the individual coefficients are always less than 1, so the result of an intermediate multiplication (of a coefficient by a previous input to the filter) has not increased in magnitude, but has increased in terms of the number of places after the decimal point. Word length limitation by quantization is therefore required. (This is especially true for recursive filters, where the word length would increase without bound.)

The result of an addition of two B-bit numbers need not be quantized, but has the possible problem of overflow. Consider a second-order recursive filter (figure 68).

Figure 68: Limiting the Word-Length in a Recursive Digital Filter, with Overflow Control (P) after every Addition and Quantization (Q) after every Multiplication. T represents a unit delay, where a signal is delayed by one sampling interval (Van Den Enden 89)
The word-length limiters have been included in two different ways. Unfortunately, the filter is now non-linear, and a number of specific effects (such as oscillations in recursive filters) can arise as a direct result. Certain kinds of input stimulus, such as $x[n]=0$ or a sinusoidal signal are most likely to cause these problems (Van Den Enden 89). In addition, the non-linearity due to quantization causes problems different to those associated with overflow prevention.

The adverse effects depend upon the following:

- *the filter structure (some forms are more prone to trouble than others)*
- *the positions of word-length limiters*
- *the characteristics of $Q$ and $P$ (i.e. rounding or magnitude truncation, saturation or 'sawtooth')*

### 6.5.3.2.1 Overflow of Intermediate Results

The overflow of intermediate results can have many effects, from oscillation to sub-harmonics of the input signal (Van Den Enden 89). The problems of overflow can be avoided by scaling the input signal, so that overflow does not normally occur, if at all. In general, the best way to handle overflow if it occurs is by employing the saturation characteristic (figure 65), since this introduces fewer undesirable effects.

### 6.5.3.3 Quantization of Intermediate Results

The most troublesome effect caused by the quantization of intermediate results is that of parasitic limit cycles (Van Den Enden 89). Limit cycles most readily occur in recursive filters, when the quantization characteristic is one of rounding, and the input
error (limit cycle). The easiest solution to this problem is to use magnitude truncation as
the quantization characteristic, although this introduces more quantization noise than
rounding for near zero amplitudes.

Limit cycles generally have a small amplitude (unlike overflow oscillations). By
reducing the quantization step of all the intermediate result quantifiers in a given filter,
the limit cycle amplitude is also reduced (Van Den Enden 89). If the output of the filter is
then quantized to a greater level than that of the intermediate results, the filter becomes
'apparently' free of limit cycles. This indicates why a circuit like that shown in figure 69
is preferable to placing the word-length limiters as shown in figure 68.

6.5.4 Conclusion

The effects of finite length arithmetic in digital systems can be severe. The
quantization of filter coefficients leads to a change in the frequency response of the filter.
Improper quantization of the intermediate results in a digital filter can lead to oscillations
(parasitic limit cycles) and badly managed overflow can lead to both oscillations and sub-
harmonics being produced in the output signal.

Great care must be taken when designing a new system that involves recursive
digital filters, and when implementing an existing algorithm on a new processor.

For a new system, it is important to ensure that the level of quantization at every
point is the minimum possible. For a second implementation, the level of quantization at
any point must match that used in the original system.

In digital filters, scaling must be used to reduce the probability of overflow prior
to any addition operations. Overflow, when it happens, should use the saturation
characteristic. As with quantization, when implementing an algorithm on a new micro
processor, the level of scaling and the overflow characteristic used should be the same as
that of the existing codec.
6.6 Non-Standard SB-ADPCM Algorithm Development

6.6.1 Introduction

The prototype codec used a pair of NEC μPD7720s DSP chips, one for the encoder and one for the decoder (Lee 84). The first part of the section gives a very brief description of this DSP, in order to identify the word size required in a transputer.

A brief description of the algorithm, its component parts and the salient implementation details are included. This description is intended for those interested in following on from this work.

The section finishes with a description of the process structure of the transputer implementation of the algorithm.

6.6.2 Information Available

The following information was available:

- two non-standard codec manuals
  a non-embedded algorithm version and an embedded algorithm version
- program listing for a μPD7720 implementation
  The listing was for the embedded version.
- G.722 standard CCITT manual
  This was used to get some idea of the detail necessary to implement an algorithm.

It was not known at the outset which algorithm had been implemented in the non-standard codecs. This made the task of implementing the system extremely difficult. A considerable amount of extra work was therefore necessary in order to correctly implement the algorithm.

6.6.3 Comparison of Processor Behaviour

6.6.3.1 The Non-Standard Codec Processor (μPD7720)

The processor generally uses 16 bits to store numbers in two's complement form and has an overflow bit. There is a 16x16-bit multiplier, which produces a 31-bit result ('2B-1'-bit result from two B-bit numbers). The multiplier makes the μPD7720 particularly suitable for implementing recursive digital filters, where intermediate multiplication results must not be quantized to the processor's usual register length of 16 bits (see section 6.5.3.4).
For recursive filters, scaling is usually employed to prevent the occurrence of overflow (Van Den Enden 89) and the μPD7720 uses scaling at the input to the predictor. Despite this precaution, a complete absence of overflow cannot be guaranteed, so a suitable overflow characteristic must be chosen. Saturation introduces fewer problems than either zeroing or ‘saw-tooth’ overflow control (Van Den Enden 89). The non-standard SB-ADPCM algorithm uses the saturation overflow characteristic throughout.

6.6.3.2 Implementing the Non-Standard SB-ADPCM Algorithm on the Transputer

In order to store a 31-bit multiplication value, a 32-bit version of the transputer is required. Quantization to 16 bits from a 31-bit result will be required at the output of the recursive filter, which is part of the predictor (see section 3.3.4.3). Rather than dividing by $2^{15}$, this can be achieved in a similar way to that in the μPD7720. By placing the 31-bit output of the recursive filter in an array, and taking the top 16-bit integer as the result, division by $2^{15}$ is achieved. The ‘division’ operation using an array is faster than standard division (Inmos 88). Overflow protection is required after every addition (see section 6.5.2.2), particularly since an overflow on the transputer will cause a system halt (Inmos 88). As discussed previously, a saturation type of overflow characteristic should be used.

6.6.4 SB-ADPCM Algorithm Overview

A detailed description of the principles behind the component parts of the algorithm has already been given (see section 3.3). The text that follows is not meant to replace that provided by the manuals (Lee 84, CCITT SG XVIII), but is intended as a supplement. The details presented here do not include the Quadrature mirror filter (see section 3.3.4.3) or the noise filter operation (see section 3.3.4.2.1), since these will not be required in the transputer implementation (see section 6.7). The two SB-ADPCM algorithms (references Lee 84, CCITT SG XVIII) are very similar, but the embedded version uses a less precise estimate in the feed-back loop than the other. A block diagram of each algorithm (one band) can be seen in figures 70 and 71.
The embedded version allows the possibility of operating the codec at a number of different rates in order to transmit a low-rate data channel alongside the speech, or to transmit speech at a lower bit-rate across packet networks. The possibility of reducing the number of bits transmitted across the Unison network was never pursued, due to the excessive amount of bit manipulation required.

Both algorithms use the same accuracy for results (excluding the predictor's intermediate products): 12 bits * $2^4$ (which equals 16 bits).
6.6.5 SB-ADPCM Component Operation

6.6.5.1 The One-Word Memory Adaptive Quantizer

The quantizer used in both algorithms is the one-word memory adaptive quantizer (see section 3.3.4.1.2), that was developed for use over packet networks by Goodman and Wilkinson (Goodman 75). The quantizer is time-varying, in that its input/output is controlled by the output code sequence. The design goal is to maintain a constant loading factor, and the system may be thought of as a fixed quantizer with an automatic gain mechanism employed in the feed-back loop. This method of quantization is vulnerable to transmission errors, since it employs feed-back. The automatic gain mechanism, however, has the capability of dissipating transmission errors, with a 5 ms time constant, since a leakage factor is used in the equation which generates the step-size (Goodman 75), and see section 6.6.5.1.2.

6.6.5.1.1 The Fixed Quantizer

This is of the mid-riser type (figure 72).

![Figure 72: A 3-Bit Uniform Mid-riser Quantizer](image)

The number of levels is a power of 2, which is convenient for a binary coding scheme. The quantizer has an equal number of positive and negative levels, which are symmetrically placed about the origin.
\[(i+1) = i^k \cdot M(c(i))\] where:

\[k = 63/64\] (leakage parameter)

\[(i+1) = \text{new step-size}\]

\[i = \text{previous step-size}\]

\[M(c(i)) = \text{multiplier based on previous code-word}\]

The new step-size can be seen to depend on the previous step-size and a multiplier which depends on the previous code word. In practice, logs (to base e) of both sides are taken:

\[\log_e(i+1) = k \log_e i + \log_e[M(C(i))]\]

The step-size is found from a table of inverse logs, which stores values between 1 and 1022. This corresponds to a dynamic range of approximately 60 dB and a step-size resolution of 0.43 dB.

The multipliers have been chosen to maximise the SNR. It is important that multipliers which correspond to the inner levels are less than but close to unity. This ensures that the rate at which the step-size increases is greater than its rate of decrease; this minimises the occurrence of serious overload errors.

### 6.6.5.2 Implementation Details

#### 6.6.5.2.1 The Quantizer

The uniform quantization characteristic means that processing here is minimal. Quantization occurs after multiplication by the inverse step-size (i.e. division by the step-size). Value truncation (see section 6.5.2.1) is used to distinguish between those values quantized to +0 and those values quantized to -0 (represented as -1). For the low band, the range of values is -16 to -1, and 0 to +15. The bit representations for the low-band code words have the top three bits replaced by the high-band contribution, and the result is the output code word.

#### 6.6.5.2.2 The Inverse Quantizer

Inverse Quantization is achieved:

- by subtracting code words in the range 16 to 31 from 32 and negating the result
- then by multiplying the result up by the desired power of two and adding 1/2

#### 6.6.5.2.3 Step-Size Adaption

The adaption rule is implemented in practice as follows (multiplied by 64\(^2\)):
\[ 64 \times 64 \times \log_e(i+1) = 63 \times 64 \times \log_e i + 64 \times 64 \times \log_e(M(c(i))) \]

The multiplier code words are stored in a table as \(64 \times 64 \times \log_e(M(c(i)))\).

The initial value of \(\log_e i\) is stored as \(64 \times \log_e i\).

The manuals (Lee 84, CCITT SG XVIII) indicate an inverse log look-up table storing 153 values between 32 and 65400 (multiplied by \(2^8\) bits) but, in this version of the program, only 140 are used. Consequently, the value \(64 \times \log_e(i+1)\) is limited to the range \(-8250\) to \(19866\) (-2.08 to 4.85), although it should have been -2.77 to 4.85.

This range of numbers (-8250 to 19866) must be converted to a suitable index for use in the inverse log look-up table. This conversion is achieved by multiplying the log step-size by the log increment of the table and converting the range to all the possible values (13 to 153).

A quantity representing the inverse step-size (multiplied up by a power of 2) is required in the feed-back loop of the encoder. As above, a table of values is used to give the appropriate step-size.

### 6.6.5.3 The Linear Predictor

The linear predictors (see section 3.3.4.1.3) are fixed, all-pole filters. The predictor for the lower band is a four-tap filter, while that for the upper band is a single-tap filter. The predictor can be viewed as a linear recursive filter with feed-back. The input to the predictor is the summation of the input to the all-pole filter and the previous output (figure 73).

![Figure 73: A Fourth Order All-Pole Linear Predictor with Feed-back](image-url)
6.6.5.4 Implementation Details

The predictor coefficients have 13 bits of accuracy, and are stored multiplied by 16384 ($2^{14}$) to give the required resolution. Intermediate results are stored in 32-bit variables and the resulting sum of the intermediate results is quantized to 16 bits, using magnitude truncation (see section 6.5.2.1).

While figure 73 shows that the input to the predictor is delayed by one sample, in practice, because the estimate is evaluated after the output of the system has been calculated, this first delay must be omitted.

6.6.6 Simulation Studies and Algorithm Verification

The algorithm was developed using simulation and algorithm verification facilities (see section 6.4.2). It was discovered that the non-standard codecs used the embedded version of the algorithm.

6.6.7 Transputer Implementation

The speech coding algorithm is inherently sequential; there is no benefit to be gained from trying to make the implementation parallel. The implementation is also very processor power-intensive, which means that the algorithm should be implemented in a sequential fashion, which saves processor time (Inmos 88). The transputer processor used to implement the algorithm is the 25 MHz T425. The transputer implementation of the encoder and decoder can be seen in figure 74.

Figure 74: Encoder and Decoder Process Structure
Two high-priority decoupling processes are included for communication to other transputers (see section 5.6.5.2). The actual encoder or decoder process runs at low priority. No effort is made to report statistics from the encoder and decoder processes, since there is no spare time available.

6.6.7.1 Real-Time Operation

The encoder was checked for real-time operation, and was found to take $2.9084 \times 10^{-3}$ seconds per 28 byte packet. This is adequately fast, since each packet contains $3.5 \times 10^{-3}$ seconds worth of speech. The fact that the algorithm works in real-time was also ascertained by examining the hand-shake signals in the X21 interface circuitry (see sections 5.5.3.3.2, 5.5.3.3.3). No buffering in the T222 was present in the algorithm verification set-up (see section 6.4.2.2).

6.6.8 Conclusion

The non-standard embedded SB-ADPCM algorithm was implemented on two 32-bit 25 MHz T425 processors. Algorithm instantaneous values were quantized to 16-bits. The intermediate results in the predictor were stored as 31-bit values. Outside of the predictor, magnitude truncation was used after multiplication operations, and saturation was used after addition operations.

The algorithm was developed using simulation, and verified for correct operation using a non-standard codec. Real-time operation was verified.
6.7 SB-ADPCM Mixing

6.7.1 Requirements

In the teleconferencing bridge, each stream will be decoded and sent to the mixing process. The mixing operation will combine the streams in the appropriate manner. The streams will then be encoded and sent to the recipients. Each listener must receive a combination of the other two speakers' outputs.

The mixing process should not add distortion or extra quantization noise.

6.7.2 SB-ADPCM Mixing Options

Mixing should be accomplished by separately spanning the sub-bands (CCITT G.722, Taka 88, Maitre 88, Mermelstein 88), since this eliminates the need for Quadrature Mirror Filters.

The following advantages are associated with the elimination of the QMFs in the bridge:

- The resulting signal quality is maximised.
- The large delay associated with this filtering is not incurred. This is very advantageous in the Unison wide-band speech system, since minimal end-to-end delay is a priority.
- Adaptive echo cancellation is easier at the sub-band level (Gilloire 87)

Mixing at the sub-band level is achieved simply by adding the high- and low-band components separately. As was discussed in section 6.5.3.2, addition of integers can lead to overflow. Consequently, scaling and saturation overflow control might be used.

Scaling could take the form of division by two, but then the resulting volume would be halved, which means more quantization noise per encoding/decoding pair.

Another solution could be one of limiting, based on an average magnitude measure, although this leads to amplitude distortion (see section 3.5.1.1.2).

Teleconferencing systems currently in use incorporate either directional microphones (see section 3.5.2.1.1) or digital voice switching (echo suppression) where input stream activity attenuates the output to the same party (Murano 90); however, digital voice switching leads to voice clipping (see section 3.4.3.1.3).

Recent efforts in the field of acoustic echo control, propose echo cancellation instead of voice switching (Murano 90, Yasukawa 89, Gilloire 87). The echo cancellors
proposed, however, are complex and processor power-intensive, (because the echo path is long for satellite connections), and time-varying (because objects and people may move around during the conference, Gilloire 87). Echo cancellation in wide-band speech systems at the sub-band level shows advantages over echo cancellation in the full band (Gilloire 87).

The simple method of mixing (scaling and saturation overflow control) was used in the prototype, while maintaining the pre-requisite that more processors might be included for system enhancement.

A statistical information interface was also developed.

6.7.3 Conclusion

It is evident from the preceding discussion, that the mixing of three voice streams is not trivial, and that there is a large scope for digital audio algorithms to improve the resulting speech quality. Therefore, while the system should be designed with only a reasonable amount of processing power available for mixing, the option for easy system enhancement to include more processor power is desirable.
the methods used to match the performance of the two processors was also included. The transputer implementation of the non-standard algorithm was described, and real-time operation was verified by timing execution of the encoder. The implementation did not result in excessive end-to-end delay, as the bridge added approximately 6 ms of delay per packet (3.5 ms worth of speech). This text was included for those who wish to follow on from this work.

The techniques involved in mixing three streams of speech were described, and a prototype method of scaling, with overflow control, was implemented.
Chapter 7
The Unison Three-party Wide-band Speech Teleconferencing System

7.1 Introduction

This chapter describes the work done on the Unison three-party teleconferencing system.

The aims of this work are to provide the following:

- *cost minimisation of hardware for the bridge*
- *a configuration that re-uses as much of the previously described system as possible*
- *a system that will cope with variations in network loading, and the mixture of types of traffic that is necessary for an integrated office*
- *optimum voice quality*
- *a system that is easy to set-up and orchestrate*
- *a system that can be easily enhanced*

The first part of this chapter discusses the design considerations: the existing components developed earlier that can be re-used in this system, and the development effort required.

The second part of this chapter considers the hardware that was built for the bridge, and how correct operation was verified.

The third part of this chapter discusses the parallel approach taken to the software design. This section also describes the statistics reporting mechanisms used that were developed from those of the two-party system.
7.2 Design Considerations

7.2.1 Introduction

The three-party teleconferencing wide-band speech system was initially developed for use over a single CFR. This section aims to identify the design effort required to supplement the two-party system (see chapter 5), so that it will cope with three parties.

The three-party system will be implemented using the bridging method (see section 6.2.3.3). The necessary transputer software to implement the encoding and decoding operations has already been described (see section 6.6).

7.2.2 Existing Components

The existing components that will be used in the three-party system are as follows:

- two-party voice protocol software (see section 5.7)
- two-party audio system (see section 5.2)
- software for encoding and decoding non-standard embedded SB-ADPCM (see section 6.6)
- CFR interface software

The CFR interface software was developed by RAL for point-to-point communication across the Unison network. Only part of the software interface will be used, as the three-party system will be developed to operate only over a single CFR.

7.2.3 Expected System Layout and Extra Equipment Needed

The expected system layout can be seen in figure 75.
The system requires the provision of a new application interface: codec, CFR interface, X21 interface, audio system, LCD pad. New hardware is also required for the bridge.

7.2.4 Extra Software Required

7.2.4.1 Voice Protocol Software

Within the bridge, packets collected from each participant must be synchronised, and delivered to the decoding/mixing/encoding system at a regular rate. The two-party voice protocol software can be used for each voice stream in the bridge, to interface between the asynchronous system (CFR) and the synchronous system (mixing). In the two-party system, the voice protocol software operates at the rate dictated by a local crystal. In the bridge, the same method can be employed (provisioning a local crystal), or the clocking required can be implemented in software (using the transputer’s internal clock).

The provision of a local hardware clock requires extra circuitry (link adapter). In order to keep costs down, the software approach was chosen.

7.2.4.2 CFR Interface Software

The Cambridge Fast Ring interface was originally designed to cope with one duplex stream. Since this system is being developed only for use over a single CFR (no
RPC mechanism being executed), some of the software is not required for this prototype system. However, the interface will need modification to handle three streams simultaneously.

### 7.2.4.3 Voice Mixing Software

The prototype mixing software (see section 6.7), can be easily expanded to cope with three streams.

### 7.2.4.4 Statistical Reporting Software

A coherent statistical information reporting mechanism is required for the three-party system:

- The individual participants report statistical information from the voice protocol software (see section 5.7.2.4.1). The same reporting mechanisms will be available for each stream in the bridge.
- The method used to gather information during mixing development (see section 6.7.2) can be extended.
- No reporting from the encoder and decoder processes is required.
- The CFR interface reporting mechanism needs to be extended to give information on the enhancement to cope with three streams.

Results from all the possible reporting points can either be collected simultaneously by a system which is permanently connected to each point, or can be collected separately by moving a collecting system around each reporting point.

The latter approach was chosen since, in a real environment, simultaneous access to all reporting points is not possible. The collecting mechanism developed earlier (see section 5.5.7) will again be used.

### 7.2.5 Conclusion

Modification of some parts of the two-party system is required to expand to a three-party system:

- production of a software 'clocking' mechanism in the voice protocol software
- alteration of the CFR interface to simultaneously handle three voice streams
- extension of the prototype mixing software to cope with three streams
- extension of the existing statistical information reporting mechanisms to provide comprehensive information from around the system
7.3 Bridge Hardware

7.3.1 Requirements

The CFR interface used earlier (see section 2.6.1), will be used for bridge connection to the LAN. In addition, new hardware for the bridge needs to be developed.

The bridge hardware development has the following requirements:

- **low cost**
  
  This can be achieved by efficiently using transputers, and by tailoring the transputer type to the required function. The voice protocol software can run on 16-bit transputers (see section 5.5.2.3), while encoding and decoding functions must run on 32-bit transputers (see section 6.3.4).

- **easy enhancement capabilities**
  
  For ease of enhancement, the system must be improved to provide better mixing performance, which may require extra processing power (see section 6.7).

- **provision of adequate reporting points for system evaluation**
  
  The preferred method is to allocate a separate link from the main speech thoroughfare for this purpose (see section 5.7.2.3). The expected reporting points are the voice protocol instances, and the mixing process.

7.3.2 Processors Required for the Bridge

7.3.2.1 Voice Protocol Instances

Each implementation of the voice protocol requires a T222 processor, since the voice protocol software uses almost all the available on-chip memory. Consequently, three T222 transputers are required to implement the voice protocol software in the bridge.

7.3.2.2 Encoding/Decoding

Each instance of SB-ADPCM encoding and decoding requires a 25 MHz T425 transputer to operate in real-time. Consequently, six T425 transputers are required to implement the encoding/decoding functions within the bridge.

7.3.2.3 Mixing

One processor provides sufficient power for mixing; however, because each transputer only has 4 hardware links, at least one other transputer is required to interface to six encoding/decoding transputers.

The option for system improvement (better mixing performance, which may require extra processing power), and the provision of a reporting point are required.
further transputer is therefore necessary, because if only two transputers are used for connection to the six encoding/decoding processors, no extra hard links are available; consequently, three transputers (two T222s and a T425) are required for the mixing function.

7.3.2.4 Total Processor Requirements for the Bridge

The bridge (without CFR interface) requires a total of twelve transputers (five T222s and seven T425s).

Tailoring the transputer type to the function means that the minimal cost requirement can be achieved. The requirement for adequate reporting points means that apart from the single link to and from the CFR interface transputer, an extra five statistical information reporting links are necessary.

Note that it is not possible to multiplex the voice protocol reporting point links together within the CFR interface (as was done for the two-party system (see section 5.7.2.3), since there are not enough available on the CFR interface processor to provide a statistical information link, and connect to both the bridge board and the collecting point.

The bridge board layout can be seen in figure 76.

Figure 76: Bridge System Processor Diagram
7.3.3 Bridge System Clocking

The bridge system will be regulated via a software 'clock' (see section 7.2.4.1). This will take the form of three independently-running software regulators executing on the T222 voice protocol transputers. In order to ensure that the 'clocks' all run at the same frequency, all the bridge transputers are tied to the same 5MHz crystal. This means that, averaged out over a period of time, the software 'clocks' should all operate at the same rate, instantaneous discrepancies being due only to scheduling differences (see section 5.4.2.2.4).

As was decided upon in the two-party implementation, data will be transferred around the system on a CFR slot basis (see section 5.7.2.1.4). Note that after each packet has been decoded, the size of the packet becomes 56 sixteen-bit words (high- and low-band contributions). While ensuring that communication between transputers is more efficient, transferring information around the system on a 3.5 ms worth of speech basis also means that a lower 'clock' resolution is required.

7.3.4 Hardware Implementation

The system was wire-wrapped onto a single power plane board. The system services (5 MHz clock, reset and analyse) were distributed amongst the processors via an octal tri-state non-inverting buffer.

Initially, the system was designed to operate with all on-board links running at 20 Mbps. The fully operational board has on-board links running at 10 Mbps, and a high power version of the system services driver (74 ALS244). A modification to the link speed was necessary, as the system would not work with links operating at 20 Mbps, because of noise interference. A high-power version of the system services driver was needed due to the large number of transputers being fed from it.

The error outputs from the processors were connected as shown in figure 77.
The T425s have the capability of daisy-chaining the error signals (to cause complete system halt on error), whereas the T222s do not (Inmos 88).

7.3.5 Hardware Testing

Processor and link validation was performed using the program developed for the two-party system hardware (see section 5.5.7). This software highlighted problems associated with an incorrect transputer capacitor type being used (inter-transputer link communication suffered protocol errors), and the effect of noise interference when operating the links at 20 Mbps.

7.3.6 Conclusion

Twelve transputers are required for the bridge. The network of twelve transputers was wire-wrapped onto a single power plane board. All the transputers are fed from the same 5 MHz crystal, to facilitate more accurate software 'clocking' instances.

The requirement to minimise the cost is achieved by tailoring the individual transputer type to its function.

The option of mixing enhancement has been catered for by the addition of a third transputer. This is in addition to the two necessary to implement the prototype mixing functionality.

The requirement to provide adequate reporting points has been built into the bridge hardware by providing five extra link connections for statistical information reporting.
7.4 Bridge System Software - A Parallel Approach

7.4.1 Requirements

The real-time concurrent software for the bridge is described using MASCOT 3 (Axford 89). The details of some of the component parts of the software have been described elsewhere, and so will not be repeated. The software for the CFR interface is taken in part from the RPC software (developed at RAL), and where appropriate this will be stated.

The requirements for the bridge software are as follows:

- **robust, 'fair' CFR interface for 3 streams**
  A 'fair' interface should not starve any of its inputs.

- **must cope with all variations in network loading**
  This is as described in section 5.7, the system must be able to cope with lost blocks and changes in end-to-end delay.

- **accurate software 'clocking' mechanism**
  This is described in section 7.3.3.

- **suitable system reporting**
  As described in section 7.2.4.4, coherent statistics are required to assess correct system operation.

7.4.2 Software Required

The software that must be developed was discussed in section 7.2.4:

- **software 'clocking' mechanism**
- **robust three-party CFR interface**
- **mixing sub-system**
- **coherent statistics reporting sub-system**

7.4.3 Software Specification

7.4.3.1 External Requirements and Constraints

7.4.3.1.1 Hardware Requirements

There are a number of bugs associated with the CFR interface logic:

- **The transmitter may occasionally re-transmit packets when it should not.**

  This problem was rectified in the two-party system voice protocol software, by using the sequence number checking mechanism (see section 5.7.2.4.2). Since the same voice protocol software that was developed for the two-party system is being incorporated into the bridge design, this problem will not require extra software development.
• The receiver may occasionally hang.

This error condition requires a chip disable/enable. This problem was rectified by the original CFR interface software that was developed at RAL.

7.4.3.1.2 Software Constraints

One of the requirements for this software is that the data latency must be kept as low as possible. As with the two-party system, data transfer around the system will be in arrays of 3.5 ms worth of speech (see section 7.3.3). This eases scheduling difficulties and saves processor time (see section 5.7.2.1.4).

7.4.3.1.3 System Test Requirements

As with the two-party system, software operation must be assessed by the collection of statistics. Statistical information is generated at reporting points in response to a command. The collecting point will be the monitor system developed for the two-party system (see section 5.5.7).

7.4.3.2 Design Proposal

The following functions of the bridge system software were identified in the design considerations and software constraints:

• robust three-stream CFR interface
• voice protocol software to 'synchronise' streams
• non-standard embedded SB-ADPCM encoding and decoding
• stream mixing

The top level design proposal can be seen in figure 78.
The voice protocol software for the three-party system is nearly the same as that developed for the two-party system (see section 5.7). An additional piece of software will be added to implement a software 'clocking' mechanism.

7.4.3.3 Network Decomposition

The effects of hard-link communication need to be decoupled from an individual transputer's main processes (see section 5.6.5.2). Consequently, at every such interface high-priority buffers with a capacity of one block were inserted at either end of the link.

7.4.3.3.1 CFR Interface Sub-System

The CFR interface is presented to the host as a series of registers (Hopper 86). Occam code to use this interface and to provide for the marshalling and un-marshalling of arguments was developed at RAL, as part of the generic RPC interface that was used for all transputer-based applications. Since the bridge system was initially developed for use over a single CFR, many of the processes incorporated in the RPC were redundant in this instance. The processes making up the CFR interface can be seen in figure 79. The transmitter, receiver and 'chip-fix' processes were part of the original RPC code. The rest of the processes in the diagram were coded specifically for the bridge system.
The priorities of the individual processes can be seen in figure 79. As discussed previously, the processes connected to hard links all run at high priority. The CFR transmitter and receiver processes necessarily must include high resolution timers, which are used to control the CFR interface. The CFR receiver process is also responsible for reacting to interrupts generated as a result of any incoming CFR mini-packet, and therefore must be executed at high priority. The rest of the processes in the three-stream version of the CFR interface execute at low priority.

The function of the 'chip-fix' process is to arrange for the station to transmit a packet to itself periodically. This is to detect the lockout situation, where the receive part of the station chip has hung. If this fault has occurred, disabling and re-enabling the station chip makes the system operational again.

The set-up process is responsible for instructing the 'transmit packet' and 'receive packet' processes to allow packets to be passed onto other processes later on in the pipe-
line, rather than throwing packets away. A further function of the set-up process is to allow for re-synchronisation in the sequence number managers.

7.4.3.3.2 Voice Protocol Sub-System

The voice protocol sub-system was described in section 5.7, and can be seen in figure 80.

![Figure 80: Voice Protocol Sub-system - Network Decomposition](image)

The software 'clock' resides in the high-priority, hard-link de-coupling process on the receive side of the protocol software. It needs to run at high priority because access to the more accurate high-priority transputer clock (1s) is required.

7.4.3.3.3 Encoding/Decoding Sub-System

The process structure for the encoding/decoding sub-system was described in section 6.6.7.

7.4.3.3.4 Mixing Sub-System

The mixing process structure can be seen in figure 81.
A variant tag protocol (Inmos 88) has been used on channels which carry speech packets from different sources. The decoupling processes all run at high priority, while the other processes run at low priority.

### 7.4.3.3.5 Statistical Reporting Sub-System

The statistical reporting sub-system is composed of a number of totally separate reporting points (see section 7.2.4.4). This is illustrated in figure 82.
Statistics may be obtained from the following positions in the bridge:

* the three stream CFR interface

The reporting facility from the CFR interface was initially developed at RAL as part of the RPC package. It was modified by the author to interact with the collecting point interface. The modifications are the addition of processes 'Message' and two instances of 'Screen Buffer Manager' (figure 83). Statistics reporting to and from the sequence tagging and management processes (that were present in the two-party system) have been removed, due to lack of storage memory (only 4 Kbytes of on-chip RAM available).

* the mixing sub-system

The collecting point interfaces directly with the 'Mixproc' process.

* the voice protocol sub-system

The statistics reporting software is the same as that developed for the two-party system (see section 5.7.2.4.1). While for the two-party system, access to the reporting point was via the CFR interface processor, here access is direct. Consequently, the collecting point software has been changed for this interface (the communication messages are in the form of a two-byte array). It is not possible to change the voice protocol software itself, due to lack of storage memory. (It is very easy to change the collecting point software at run-time, since this program will easily terminate).
7.4.3.4 Element Decomposition

7.4.3.4.1 Three-stream CFR Interface

The transmit packet process (txpac) accepts inputs from three sources and sends the output to the transmitter process that communicates with the CFR interface itself. The function of the process is to operate a ‘fair’ selection procedure.

The problems associated with the ALT construct in Occam have been discussed earlier (see section 5.6.3). A normal ALT construct was initially tried in this process; starvation in some of the inputs was seen for a T414 based CFR interface, whilst not for a T800 based CFR interface. Note that the T414 and T800 are pin-compatible 32-bit transputers, the T800 supports floating point arithmetic, whilst the T414 does not (Inmos 88). The difference in behaviour is due to the different way that a T414 implements the ALT construct compared to a T800!
To prevent starvation, a mechanism based on boolean flags and an 'IF' construct was used:

```
IF
  last.1 -- boolean flag
  PRI ALT
  (got.1 OR got.2 OR got.3) & c.fetch ? any --regulator not busy
  {output the packet and set the relevant 'got' flag to false}
  input 2 ready?
  {read in input 2 and set last.1=false, last.2=true, last.3=false, got.2=true}
  input 3 ready?
  {read in input 3 and set last.1=false, last.2=false, last.3=true, got.3=true}
  input 1 ready?
  {read in input 1 and set last.1=true, last.2=false, last.3=false, got.1=true}
last.2
  {set up another PRI ALT construct, but this time favour input 3, and
  then input 1 and finally input 2}
last.3
  {set up another PRI ALT construct, but this time favour input 1, and
  then input 2 and finally input 3}
```

The point with the above solution to the problem of a 'fair' ALT is that while the above solution still uses a costly PRI ALT construct (see section 5.6.4), the time penalties are not going to be excessive. This is because of a high probability that more than one input to the process will be ready (otherwise starvation would not have occurred), and the number of input choices for the PRI ALT is not great. The 'fair' selection method above did not produce starvation of any inputs.

The 'txpac' process is decoupled from the CFR interface transmitter process by the regulator.

### 7.4.3.4.2 Voice Protocol Sub-System - Software Clock

At first sight, the code to implement a software clock might appear to be relatively simple; introduce a delay into the output of the high priority process. What has been omitted in this solution however, is that the scheduling of the process (after the timer has expired), may add to the measured delay. This is due to the presence of other high-priority processes on the same transputer that decouple the effect of hard-link communication from the main processes (see section 5.6.5.2).

A more accurate solution to the problem is to measure the actual delay that occurred, and to alter the delay in the next interval to account for the delay variation.
The value of the delay was chosen by measuring the frequency of the three crystal oscillators (on the participant's X21 interface circuitry (see section 5.5.3.1)), and taking the average.

The solution above was implemented in the bridge, and while successful in maintaining a constant rate, the solution led to problems due to insufficient transputer timer resolution (1s). The symptoms were that the participants' buffers under-ran, and the bridge's voice protocol buffers filled up (leading to buffer resets for the participators, and blocks being thrown away in the bridge). The degradation in speech quality was a barely noticeable click at regular intervals.

The accuracy of this solution can be improved by varying the measured delay in the software clock slightly over a number of intervals. This enables the frequency of the software 'clock' to better match that of the local crystal oscillators.

7.4.3.4.3 Mixing

The mixing process in the bridge was very simple. Each stream was scaled, before being combined in the appropriate manner; each listener receives a combinational stream from the other two participants (see section 6.7). While this approach causes more quantization noise, very good quality speech resulted, with no disturbances.

7.4.3.4.4 Statistics Gathering

After booting, the channel connected to the bridge can be moved around the reporting points to collect statistics as required. The collecting point (or monitor) software can be easily changed to communicate with the voice protocol software instances.

Immediately after booting, messages are received un-prompted from the CFR interface. Once these subside, all communication is on reply to a prompt from the terminal. This method of communication means that the collecting point can be connected at will to different reporting points. If communication were un-prompted, communication on the collecting point's hard-link would be prone to crashing. If the hard-link is removed during communication, a protocol error occurs, leading to BOO4 system halt.
7.4.3.5 System Testing

The three-party system has been built from software sub-systems that have been rigorously tested prior to their inclusion. Apart from minimal subjective tests, the system was soak tested over a CFR. The system was run over an 8 minute interval.

After the 8 minute interval, some reports from the voice protocol sections were collected in order to provide a history of the events that took place.

<table>
<thead>
<tr>
<th>Processor Number (B=Bridge, P=Participant)</th>
<th>Blocks Repeated</th>
<th>Blocks of Silence</th>
<th>Number of Resets</th>
<th>Blocks Thrown Away</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>B2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>B3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>P1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>P3</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The results above actually illustrate that the software clock is too slow; blocks are thrown away in the bridge voice protocol receive buffers, and a very slight under-run in the participant's receive buffers occurs. A more accurate software 'clocking' mechanism should produce even better results.

7.4.3.6 Conclusion

This section has described the software used in the three-party wide-band speech teleconferencing system bridge. The system was composed to a large extent from software sub-systems that had been produced and rigorously tested individually. This policy undoubtedly made system integration easier.

The requirements set out at the beginning of this section were largely met in this work:

- A robust, 'fair' CFR interface for three streams is required.

The interface was carefully designed without the 'unfair' ALT construct. It used boolean flags and an 'IF' construct to force 'fair' selection. The mechanism did not produce starvation of any inputs.
- **The system must cope with all variations in network loading.**

This requirement was met by using the voice protocol software developed for the two-party system.

- **An accurate software 'clocking' mechanism is required.**

The solution delivered an un-varying delay, despite the likely in-determinism inherent in parallel software systems. The rate of the 'clock' however, was found to be too slow. The resulting degradation over an eight minute interval was nevertheless shown to be very minimal.

- **The system must give optimum voice quality.**

The occurrence of block loss was found to be almost imperceptible. This is as a result of the voice protocol software developed for the two-party system. The mixing function, while extremely simple, did not distort the speech or suffer from background noise speech masking.

- **Suitable system reporting is required.**

A great deal of effort was made throughout the development of this software to facilitate unobtrusive statistics reports. While these were not as extensive as initially desired, they were adequate to investigate the system fully.

The system (with a small amount of extra work to improve the software 'clocking' mechanism), provides a suitable three-party teleconferencing system for use over a CFR.

The system can be easily enhanced (by replacing the RPC package) to integrate fully into the Unison office.
7.5 Conclusion

This chapter has described the work done on the Unison three-party teleconferencing system.

The aims of this work were to provide the following:

- *minimised hardware costs in the bridge*
- *a system that re-uses as much of the existing software as possible*
- *a system that will cope with all variations in network loading and the mixture and types of traffic that is necessary for an integrated office*
- *optimum voice quality*
- *a system that is easy to set-up and orchestrate*
- *a system that can be easily enhanced*

The requirement to minimise the hardware costs in the bridge was achieved by tailoring the transputer type to its function; less processor power intensive parts of the system were executed on 16-bit (lower cost) transputers.

A great deal of effort was expended throughout the development work to design software that would either be easy to integrate into the three-party system, or would form the basis for later work.

In re-using large parts of the previously developed software, such as the voice protocol sub-system developed for the two-party work, a system was produced that inherited the capability of handling all variations in network loading and mixture of traffic expected in an integrated office.

The requirement of producing a system that gives optimum voice quality has been of paramount importance throughout this work. Careful attention to the choice of mechanisms and software structures has produced a system which has almost imperceptible echoes, minimal degradation in speech quality due to block loss, and negligible distortion from the mixing method used.

Unfortunately, the three-party wide-band speech teleconferencing system was not fully integrated into the Unison network; however, system integration is believed to be straight-forward (see further work, section 8.2).

The capability of enhancing the system to give better mixing capabilities has been maintained in the system developed (see further work, section 8.2).
Chapter 8
Conclusions and Further Work

8.1 Conclusions

8.1.1 Introduction

This thesis has described the provision of a two- and three-party wide-band speech teleconferencing system in an integrated office, and over an ATM-like network. It discusses the practical aspects of designing, implementing, and using the system, in the prototype office and network developed for Project Unison. The value of this research is primarily that of being the most significant real-time service developed within Unison, and therefore the best measure of the behaviour of a truly integrated network.

The marketplace is currently seeing development in multi-media terminals. Aimed at the ISDN subscriber, these terminals offer some multi-media capability, such as the voice and slow-scan video terminal, and the G.722 wide-band codec and associated low-rate data channel, both designed to have a bit-rate of 64 Kbps, or multiples thereof. Constrained by the unit of bandwidth offered by the circuit-switched nature of the current PSTN networks, these terminals will not provide a basis for true office integration.

Packet networks are currently not seen as being suitable vehicles for voice. This attitude will doubtless change as packet networks, particularly ATM, begin to be able to guarantee the kind of delay behaviour that voice services need. Multi-media terminals which give the subscriber the flexibility to tailor and orchestrate communication in an individual and cost-effective way, and which offer a greater wealth of methods of communication, will encourage subscribers to demand improvement in the service offered by the network. True integration of media is only possible in a network which is capable of providing network level integration, not merely integration at the user level.

The prospect of a European rather than a Britain-wide market will also hasten the onset of truly integrated networks. Companies that can communicate effectively and quickly will be more successful than those which rely on sending staff to other parts of Europe. Effective communication between people who do not speak primarily the same language is greatly aided by using a range of communication media, not just relying on voice services.

The Unison project has investigated a network that provides a far greater level of integration than that currently offered by the commercially available ISDN service. The intelligent network features researched in Unison, such as user migration and dynamic
directory services, greatly increase the scope for effective, integrated communication. The applications developed for Unison use the intelligent network ‘services’ offered by the network, and are capable of providing better quality and cost-effective communication.

Voice will always be central to communication in an office environment. Better quality voice, such as that offered by 7 kHz SB-ADPCM, greatly improves the articulation possible between two people whose first language is not the same. A flexible voice service which can operate in an integrated office, such as that developed for Unison, will increase the subscribers’ expectation of communication, and encourage greater subscriber commitment to integrated multi-media use.

8.1.2 The Local Area Network

The Unison integrated office was based on a LAN. LANs have certain characteristics which are suitable for carrying real-time traffic (low error rates and low delay); however, LANs generally do not have the ability to guarantee minimum bandwidth to individual applications.

The CFR has the inherent capability of sharing bandwidth out among the connected stations. This characteristic, peculiar to only some LANs, is of paramount importance for carrying multi-media traffic. The CFR, running at 50 Mbps, was able to carry a reasonable amount of traffic, a stringently real-time service such as voice, slow-scan video, and other distributed computing traffic simultaneously.

8.1.3 The Wide Area Network

The Unison network architecture consisted of a number of client LANs interconnected by an ATM-like network. The long-haul segments of the network were based on prototype ISDN links with an overlay strategy to simulate an ATM-like network. The Unison network architecture rationale was to provide the real-time favourable CFR characteristics of low delay and low error rates on a wider area than normally covered by LANs.

The inter-site links use the same packet size as the LAN, a single 32-byte CFR slot, with only a very light-weight protocol to manage transmission; no error recovery or flow control mechanisms were implemented. This rationale makes the wide area network suitable for carrying real-time traffic, such as voice. The WAN, however, was still able to carry data-like traffic, which implemented error recovery and flow control on an end-to-end basis. The Unison network consequently is quite similar to the emerging ATM
network; ATM networks will transmit data in small fixed size 'cells', and provide only light weight protocol support in the network.

The Unison WAN proved to be a suitable medium for multi-media traffic, voice, slow-scan video, and computer data (Murphy 90). Particular characteristics of the WAN which made it suitable for voice services: reasonable delay (7 ms unloaded), and the fact that packet loss across the network usually occurred individually, rather than in clusters. The loss of packets individually meant that block loss was rendered imperceptible to the user.

8.1.4 Unison Integrated Office

The Unison integrated office has a loosely-coupled, distributed architecture. This has benefits in terms of performance, cost-effectiveness, potential for growth and reliability. Particularly for the wide-band voice system, this meant that the host processor could be tailored to meet the performance requirements of an embedded real-time controller. The architecture was cost-effective and had the potential for growth, since the two-party system could be expanded to a three-party system without altering the two-party hardware or software; a conference bridge was added to the network to mix three streams of voice. The configuration was reliable, since the failure of any application did not prevent the other applications from operating. The wide-band voice system also had an increased level of reliability, since it was provided with an alternative means of call set-up, rather than solely relying on the workstation; a key-pad was added to each instance of the wide-band voice system to provide the capability of independent call set-up.

A disadvantage of the distributed loosely-coupled architecture used in the Unison office is that stream synchronisation of voice and video is made more difficult than it would be for a tightly-coupled architecture. Such a stream synchronisation facility was never implemented in Unison.

8.1.5 Managing Voice Traffic Across an ATM-like Network

A lot of the research into managing voice traffic in the Unison network is directly relevant to ATM networks (Kitawaki 90). The voice protocol with its adaptive buffer control is exactly what will be required for voice services over an ATM network. Techniques discussed for voice reconstruction will also be required in the event of block loss.
8.1.6 Acoustical Aspects of Hands-Free Teleconferencing

For 'hands-free' teleconferencing, voice switching is recommended (Taka 88). Voice switching heavily attenuates the signal coming from the microphone if the loudspeaker is active for two-party connections, and does not mix inactive channels at the bridge. The voice switching scheme has benefits, however, in that it prevents echoes and feedback. The disadvantage is that some voice will always be lost, and the cutting in and out of background noise can be disturbing to listeners.

In the Unison wide-band voice system, voice switching was not used. The mixing mechanism was kept as simple as possible to avoid distortion. Feedback problems were alleviated by the use of frequency shifters, and every effort was made to reduce the perceptibility of echoes. This was achieved by minimising the end-to-end delay across the network, and by careful placement of the microphone. The resulting quality was very good, echoes were almost imperceptible, and feedback did not occur for reasonable volume levels.

The microphones used were of the pressure zone type (Crown 93). These microphones were of extremely good quality, and had many benefits. The microphones were omni-directional, with high sensitivity. Consequently, the amplitude of the resulting speech signal did not vary greatly with the distance from the microphone. The omnidirectionality and high sensitivity of the PZM meant that the system was easy to use: no volume changes due to head movements occurred.

8.1.7 The Suitability of the Transputer as a Real-Time Embedded Controller

Software for the two- and three-party systems was written exclusively on transputers. While there is reluctance in the market to adopt parallel processing techniques, it is the author's belief that the many advantages to be gained from using parallel processing far out-weigh the disadvantages.
8.1.7.1 Advantages

The advantages can be summarised as follows:

- **The software can be split into very easily managed sections.**
  This has many advantages in terms of readability, verification, and functionality decoupling. A current need in the software development community is for better quality software. It is the author's opinion that this should not be tested in, but designed in. Current research would like to develop software into an engineering discipline, and move away from the 'black art' techniques used in industry today (Hoare 85). This can only be achieved by developing techniques to prove software mathematically, which is greatly eased by splitting software into small independent sections, such as autonomous processes.

While it might be argued that non-concurrent software can also be split into small independent sections by the use of procedures, a structured software approach is not encouraged in non-concurrent software, as there are fewer benefits to be gained. Within the transputer environment, greater parallelism produces more flexibility in terms of software mapping onto processors, and produces well-defined interfaces to units of software.

- **Communication to other processors/peripherals is managed by an independent link interface.**
  The provision of an independent link interface means that communication to other processors/peripherals does not use CPU time. This is extremely valuable when using the transputer as a real-time embedded controller, as not only does the mechanism save processing power, but CPU involvement in communication (scheduling the output process in the first place) can also be reduced by communicating on an array basis, rather than on a byte basis.

- **System expansion to include more processing power is easy.**
  However well a system is engineered to have adequate processing facilities, system upgrades are likely to require more processing power. Within the transputer environment, another processor is easy to add, and the software does not need re-writing to make efficient use of the increase in processing power. This makes system upgrades far easier and less costly. Supporting old hardware configurations is also less costly, since no additional expert knowledge is required.

- **Cost reduction can be accomplished by tailoring the transputer type.**
  Communication is via a standard link interface which is compatible for all transputers. This means that, by choosing the type of transputer suited to the task, costs can be minimised.

8.1.7.2 Disadvantages

There are a number of problems with the transputer family which were highlighted by this research:

- **The ALT construct as implemented on the transputer is not 'fair', and can be time costly.**
  The 'unfair' ALT mechanism required a comparatively large amount of specifically written software to force a fair selection procedure, and to minimise time penalties.
Two levels of processor priority are insufficient. This fact has particular relevance when a high-priority process wishes to communicate with a low-priority process. The high-priority process has to wait until the low-priority process is scheduled; effectively reducing the priority of the high-priority process. What is required to overcome this is either a greater number of priorities available, or the capability to temporarily elevate the status of a low-priority process when a high-priority process wishes to communicate with it.

8.1.8 Three-party Teleconferencing

The implementation of the three-party teleconferencing system using a bridge (requiring no upgrades to the two-party implementation) is in keeping with the philosophy used throughout the Unison project: the architecture and intelligence is distributed wherever possible. The benefits associated with this decision are those put forward for the integrated office (see section 8.1.4).

The software composition of the bridge again highlights the suitability of the transputer as an embedded real-time controller. The inherent modularity of the software designed for the two-party system meant that large portions of it could be re-used in an unaltered form.

The software implemented to perform the non-standard embedded coding scheme was not really suited to the transputer; the algorithm is inherently sequential and was executed as a single process. A further point to support the unsuitability of the transputer is that most of the software operated with 16-bit resolution, with only the internal operation of the predictor requiring 31-bit resolution. The transputer range includes a cheaper 16-bit version which could not be used because of the required predictor resolution. A special DSP version of the transputer, maintaining the benefits associated with the link interfaces, would have been more suitable for this task.
8.2 Possible Future Extensions of this Research

8.2.1 Introduction

The development of the two- and three-party wide-band speech system for the Unison project identified some areas for further work.

8.2.2 Full Integration of the Bridge into the Unison Environment

The full integration of the bridge into the Unison environment would have lead to interesting research on the subject of call set-up and clear-down within the loosely-coupled architecture of the Unison office. It is the author's opinion that call set-up and clear-down would have been relatively easy to implement. Also, should a participant have decided to leave the conference, reverting to a two-party call would have been simple to accomplish, given the distributed nature of the call set-up facilities available within the Unison environment.

The full integration of the bridge within the Unison environment may have required more than one CFR interface for the bridge (given the high processing requirements of interaction with the secretary on a regular basis). The occurrence of a number of different station addresses for connection to the bridge, could have been dealt with using the concept of ports. The secretary on being asked for a <station> <port> pair, could have replied with the next free station address associated with the same port.

8.2.3 System Clocking

The clocking mechanism used for the Unison three-party wide-band speech system in the bridge could have been improved not only by increasing the resolution of the software clocking mechanism, but also by using information from the status of the receive buffers to trim the frequency. Also, rather than having a fixed crystal oscillator operating independently at each two-party interface, a trimmable voltage controlled oscillator (VCO) would have eliminated block loss due to clock drift.

An alternative approach to the problem of providing a single clocking rate for the system could have been to use the Rugby clock receivers developed by Siddiqui (89), using the transputer's event pin to read the pulses in. (The Rugby clock receivers use the radio-transmitted clocks to provide a universal fixed-rate clock.)
8.2.4 The CFR Interface

The heavy processing requirements of interaction with the secretary by each application led to problems in the two-party system. This problem was resolved by the subsequent inclusion of an extra transputer to service the codec. The idea of reducing the complexity of secretary interaction put forward by Murphy (90) is worthy of research.

An alternative to the approach put forward by Murphy (90) would be to decompose the functions of the single CFR interface transputer to execute on a pipe-line or network of transputers. This idea comes from the fact that the CFR interface consists of 8-bit registers, and 32-bit resolution offered by the CFR interface transputer is not conceivably required. Also, the occurrence of two separate paths in the low-level part of the interface (one for transmit and one for receive) does not need to be located on the same processor. The use of cheaper 16-bit transputers could lead to cost savings in the interface, as well as simpler software to fix the CFR interface bugs encountered during the project.

8.2.5 Extension of the Wide-band Speech System to More than Three Parties

This would involve directly extending the bridge to mix more than three streams, and/or using tandeming bridge units. Using extra bridge units would be relatively simple to implement, given that the embedded SB-ADPCM codec was designed to cope with up to three transcodings.

8.2.6 Alteration of Three-party Wide-band Speech System to Use the G.722 Standard

Alteration of the Unison wide-band speech system to use the G.722 standard for Embedded SB-ADPCM would be easy to implement. Transition to other future standards would also be straight-forward.

8.2.7 Graceful Bit-Rate Reduction Using Embedded Coding

A feature of the embedded coding scheme, used in both the non-standard and the G.722 standard SB-ADPCM codecs, is the ability to reduce the bit-rate of the stream. This can be altered from 64 Kbps to either 56 Kbps or 48 Kbps, by deleting one or two bits out of each 8-bit sample. This means of reducing the bit-rate was never developed for the wide-band speech system, because the capability of changing the clock frequency in each two-party interface was not available, and because the bit manipulations necessary
would require a lot of processing power. Also, within the context of the Unison project, the ability to reduce the bit-rate would not have been of much practical use.

Within the context of ATM networks, however, this capability is of extreme importance, since reducing the bit-rate of a large number of voice streams in response to network loading could lead to tremendous savings (Kitawaki 90). The means of assessing network loading in the Unison ATM-like network is available within the voice protocol. (The status of the receive buffer over a significant interval is used to determine whether the timing needs adjusting, due to changes in network loading.)

Subscriber billing in ATM networks is likely to be based on how many packets were transited by the network, rather than the customer being charged a flat rate. Giving a subscriber the option of choosing how much they spend per second on each call, (better quality means more packets sent per second, and therefore greater cost), is likely to find favour.

The above system would be easy to manage in either the Unison or an ATM network: the quality of service could be indicated in the packet header.

8.2.8 Voice Mixing

All current 'hands-free' audio teleconferencing systems use voice switching as a means of controlling echoes and feedback. The disadvantages of voice switching are that some speech will be lost, and background noise cutting in and out can be disturbing to the listener. Voice switching was not investigated in conjunction with the wide-band speech system. An alternative to voice switching might be the combination of a number of techniques, which would produce less quality degradation than is found with voice switching.

One such technique might be feedback control using a comb-filter (a description of a comb-filter can be found in Van Den Enden 89). The beginning of howl-round (singing) in 'hands-free' teleconferencing usually starts at one particular frequency, dictated by the dimensions of the participants' rooms. The idea is to have a system that swept along the frequency band to locate the frequency where the feedback was occurring, and to use a very narrow-band comb filter to attenuate that frequency. This system would extend the dynamic range of the system before feedback occurred.

The above technique would have to be used in conjunction with echo cancellation (preferably at the sub-band level), which is unfortunately more difficult with a variable network delay path than it is with a fixed delay (Gilloire 87).
Background noise would also be a problem in a system which did not use voice switching. Rather than not bridging any channels which were deemed 'inactive', the method of expansion might be used instead (Nisbett 93). This would reduce the disturbing effect of background noise cutting in and out.

8.2.9 Voice Imaging

Within a teleconferencing situation, an idea might be to use stereo/quadraphonic techniques to position a sound in space; the conference participants sound as if they are talking from different locations in the same room.
References

Adams 87
'A Simple Secretary Service'
C.J. Adams
Unison Working Paper UR039.

Ades 87
'An Architecture for Integrated Services on the Local Area Network'
S. Ades

Axford 89
'Concurrent Programming-Fundamental Techniques for Real-Time and Parallel Software Design'
T. Axford

Borwick 87
'Sound Recording Practice'
J. Borwick (editor)

Burns 87
'Programming in Occam 2'
A. Burns

Burren 87
'Priority in the UNISON Network'
J.W. Burren and C.S. Cooper

Burren 89
'Project UNIVERSE - An Experiment in High Speed Computer Networks'
J.W. Burren and C.S. Cooper

CCITT G.722
'7 kHz Audio Coding within 64 Kbps.'
CCITT SG XVIII
'Algorithm Disclosure for a Wide-band Speech Coder'
Rapportours Group on Wide-band Coding within 64 Kbps
British Telecom Document No. 57.

CCITT X.21
'Interface Between Data Terminal Equipment (DTE) and Data Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Network 8'
CCITT, Yellow Book, Vol. VIII, Fascimile VIII.2.

Chapman 87
'Wide-band Speech for the CR via a Transputer'
J. Chapman
Final year project, Department of Electronic and Electrical Engineering,

Choi 89
'Effects of Packet Loss on 3 Toll-quality Speech Coders'
A.W. Choi and A.G. Constantinides
IEE Conference on Telecommunications, April 1989.

Crochiere 76
'Digital Coding of Speech in Sub-bands'
Bell Systems Technical Journal
Vol. 55 No. 8, October 1976.

Crown 93
'The Crown Boundary Microphone Application Guide'
Crown International Inc., PO Box 1000, Elkhart, Indiana 46515 - 100, USA.

Eargle 86
'Handbook of Recording Engineering'
J. Eargle

Emling 63
'The Effects of Time Delay and Echoes on Telephone Conversations'
J.W. Emling and D. Mitchell
Wide-band Speech Teleconferencing over an Integrated Network

Flanagan 79
'Speech Coding'

Gilloire 87
'Experiments with Sub-band Acoustic Echo Cancellors for Teleconferencing'
A. Gilloire

Goodman 75
'A Robust Adaptive Quantizer'
D.J. Goodman and R.M. Wilkinson
IEEE Transactions on Communications, November 1975.

Gruber 83
'Performance Requirements for Integrated Voice/Data Networks'
J.G. Gruber and N.H. Le
Journal on Selected Areas in Communications, Vol. SAC-1 No. 6, December 1983.

Hoare 85
'Communicating Sequential Processes'
C.A.R. Hoare
Prentice Hall series in Computer Science.

Hopper 86
'Local Area Network Design'
A. Hopper, S. Temple, and R. Williamson

Inmos 88
'The Transputer Databook'
Inmos LTD, Ref. No. 72 TRN 20300.

Jayant 81
'Effects of Packet Losses in Waveform Coded Speech and Improvements due to an Odd-Even Sample Interpolation Procedure'
N.S. Jayant and S.W. Christensen
Wide-band Speech Teleconferencing over an Integrated Network

Jayant 84
'Digital Coding of Waveforms, Principles & Applications to Speech and Video'
N.S. Jayant and P. Noll

Johnston 80
'A Filter Designed for use in Quadrature Mirror Filter Banks'
J.D. Johnston

Jones 73
'Frequency Shifter for "Howl" Suppression'
M. Hartley Jones

Kitawaki 90
'Speech Coding Technology for ATM Networks'
N. Kitawaki, H. Nagabuchi, M. Taka, and K. Takahashi
IEEE Communications magazine, January 1990.

Le Tourneur 86
'Implementation of the 7KHz Audio Codec and its Transmission Characteristics'

Lee 84
'A 7KHz Bandwidth Speech Codec'
R.18.2.3 Memorandum No. 182

Lim 83
'Speech Enhancement'
J. Lim

Lu 89
'Graphics and Video over an Integrated Services Network'
G. Lu
Maitre 88

'7KHz Audio Coding within 64 Kbps'
X. Maitre

Mermelstein 88

'G.722 A New CCITT Coding Standard for Digital Transmission of Wide-band Audio Signals'
P. Mermelstein

Murano 90

'Echo Cancellation and Applications'
K. Murano, S. Unagami, and F. Amano
IEEE Communications Magazine, January 1990.

Murphy 90

'Multi-media Services in a Distributed Office'
B.J. Murphy

Nisbett 93

'The Sound Studio, 5th edition'
A. Nisbett

Olson 57

'Acoustical Engineering'
H.F. Olson

Rabiner 78

'Digital Processing of Speech Signals'
L.R. Rabiner and R.W. Schafer

Richards 73

'Telecommunication by Speech'
D.L. Richards
Shallow 91
'Maximising the Performance of Data Throughput in a Saturated Routing System'
P.A. Shallow

Siddiqui 89
'Performance Measurement Methodology for Integrated Services Networks'
M. Siddiqui

Stallings 84
'Local Networks - An Introduction'
W. Stallings

Taka 88
'Multimedia, Multipoint Teleconference System using the 7 Khz Audio Coding Standard at 64 Kbps'
M. Taka, S. Shimada, and T. Aoyama

Templeman 91
'An Alternative to ALT'
H. Templeman and D. Millot
Institut National des Telecommunications and Universite Paris-Sud, France

Tennenhouse 86
'The UNISON Data-Link Protocol Specification'
D. Tennenhouse

Tennenhouse 87
'Exploiting Wide-band ISDN : The UNISON Exchange'
Tennenhouse et Al
Unison 90
'The Final Report of the UNISON Project'
Loughborough University of Technology, Acorn, Logica, RAL, and Cambridge University, 1990.

Van Den Enden 89
'Discrete Signal Processing - An Introduction'
A.W.M. Van Den Enden, and N.A.M. Verhoeckx

Weiss 88
'Packet Switched Voice Conferencing Across Interconnected Networks'
G. Weiss and C. Ziegler

Westall 85
'Simultaneous Transmission of 7 KHz Bandwidth Speech and Data at 64 Kbps'
F.A. Westall and D.S.K. Lee

Yasukawa 89
'Acoustic Echo Control for High Quality Audio Teleconferencing'
H. Yasukawa, I. Furukawa, and Y. Ishiyama

Yin 86
'Wide-band Speech Interface for Cambridge Ring'
M.T. Yin