An Investigation of Measurement-Based Load and QoS Management in a Distributed Audio Environment

Mathew W. Kinghorn

Thesis submitted for the degree of

Masters of Science

in

Applied Mathematics

at

The University of Adelaide

(Faculty of Mathematical and Computer Sciences)

School of Mathematical Sciences

November 24, 2005

Contents

Signed Statement viii

Acknowledgements ix

Dedication x

Abstract xi

1 Introduction 1

1.1 A Brief Introduction to Voice over IP 5

1.2 Road Map 6

1.3 Contributions 7

2 Motivation 8

2.1 Current Technology 9

2.1.1 Real Time Strategies 10

2.1.2 First Person Shooters 11

2.1.3 Role Playing Games 12

2.1.4 Summary 14

2.2 The IACMMG and DICE Projects 15

2.2.1 How It Works 15

2.3 Issues 21

3 Discussion of Load Management **23**

3.1 Load Balancing Literature 23

3.2 Definitions 24

3.2.1 Static vs Dynamic 24

3.2.2 Homogeneous vs Heterogeneous 25

3.2.3 Sender-Initiated vs Receiver-Intiated 25

3.2.4 Centralised vs Distributed 27

3.2.5 Maintaining State Information 27

3.2.6 Costs 29

3.3 Example Methods 30

3.4 Load Management Scheme 33

3.4.1 Maintaining State Information 35

3.4.2 Costs 35

4 Load Management Schemes **37**

4.1 The Architecture 37

4.2 Load Balancing Algorithms 38

4.2.1 Minimum Offload 39

4.2.2 Maximum Offload 40

4.2.3 Minimum Redistribution 41

4.2.4 Simple Packing 41

4.3 Simulation Analysis 42

4.3.1 Simulation Architecture Assumptions and Parameters 42

4.3.2 Implementation Issue 46

4.3.3 Further Analysis 55

4.4 Summary 59

5 Discussion of QoS Management **61**

5.1 Goals and Definitions 61

5.2 Measuring Audio Quality 63

5.2.1 The ITU-T E-Model 63
5.2.2 Discussion of Measurement Approaches 70
5.3 Implementation Recommendations 74
5.3.1 Time 74
5.3.2 Packet Size and Inter-departure Time 75
5.3.3 Measurement Approaches 77
5.3.4 Loss 79
5.3.5 Codec Choice 80

6 QoS Measurement Schemes **82**
6.1 Proposal 82
6.1.1 Stage 1 85
6.1.2 Stage 2 87
6.1.3 Colour Coding 89
6.1.4 Check Confidence Interval 89
6.1.5 Comparison of Two Servers 90
6.1.6 Server Ranking 91
6.1.7 Parameters 91
6.2 Simulation Analysis 92
6.2.1 User Satisfaction 92
6.2.2 R-value Improvement 92
6.2.3 Simulation Architecture Assumptions and Parameters 92
6.3 Summary 97

7 Integrated QoS Management Schemes **99**
7.1 The Architecture 99
7.1.1 Minimum Offload 101
7.1.2 Maximum Offload, Minimum Redistribution & Simple Packing 101
7.2 Simulation Analysis 102
7.2.1 Simulation Architecture Assumptions and Parameters 102

7.3 Summary 113

8 Conclusion **114**

Bibliography **120**

List of Tables

4.3.1 Mean and Standard Deviation Pairs 46
4.3.2 Limited Information 51

6.2.1 Mean and Variance of the 8 delay distributions 94

List of Figures

2.2.1 Depiction of a four table Virtual Café. 16
2.2.2 Example of a grid overlay. 18
2.2.3 Users see a single server to which they connect 20
2.2.4 A network of servers allow for load sharing and maximising audio quality. 21

4.3.1 Excess Overload for the four load balancing algorithms. 45
4.3.2 Total Load for the four load balancing algorithms. 47
4.3.3 Average Excess for Minimum Offload using Forward Measurement Cycle. 50
4.3.4 Average Excess for Minimum Offload using Backward Measurement Cycle. 53
4.3.5 Average Excess for Minimum Offload using Random Measurement Cycle. 54
4.3.6 Average Excess for Minimum Offload using Forward Measurement Cycle. 56
4.3.7 Average Excess for Minimum Offload using Backward Measurement Cycle. 57
4.3.8 Average Excess for Minimum Offload using Random Measurement Cycle. 57

5.2.1 E-model rating R with G.711, G.729 and G.723.1 with 0% loss and all other parameters set at their default values (T=Ta=Tr/2). 67

5.2.2 E-model rating R with G.711, G.729 and G.723.1 with 0.5% loss and all other parameters set at their default values (T=Ta=Tr/2). 68
5.2.3 E-model rating R with G.711, G.729 and G.723.1 with 1% loss and all other parameters set at their default values (T=Ta=Tr/2). 68
5.2.4 E-model rating R with G.711, G.729 and G.723.1 with 2% loss and all other parameters set at their default values (T=Ta=Tr/2). 69
5.2.5 E-model rating R with G.711, G.729 and G.723.1 with 5% loss and all other parameters set at their default values (T=Ta=Tr/2). 69

6.1.1 E-model rating R using G.729 & 0%-2% loss. 84
6.2.1 Average User Satisfaction 96
6.2.2 Average User Satisfaction 97

7.2.1 Average Dissatisfaction with Network at 25% loading. 106
7.2.2 Average Dissatisfaction with Network at 50% loading and Server Inter-Measurement Time = 16s 107
7.2.3 Average Dissatisfaction with Network at 75% loading and Server Inter-Measurement Time = 16s 108
7.2.4 Average Dissatisfaction with Network at 100% loading and Server Inter-Measurement Time = 16s 109
7.2.5 Comparison of Average Excess Overload With and Without QoS Algorithm. 111
7.2.6 Comparison of User Satisfaction With and Without QoS Algorithm. . 112

Signed Statement

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

SIGNED: ..... DATE: .....

Acknowledgements

Foremost I would like to thank my supervisors, Professor Nigel Bean, Dr David Green and Professor Michael Rumsewicz, for their strong encouragement and high level of support throughout the entire period of my studies.

I greatly appreciate the assistance offered by Alison Jobling, who gave me much advice as to proper programming design and structure, for which I would be lost without. I would like to thank Justin Viiret for his programming assistance and googling skill.

I would like to acknowledge the financial support provided by the Smart Internet Technologies Cooperative Research Centre (SITCRC). In addition, I wish to thank the staff of TRC Mathematical Modelling for making me feel welcome for the duration of my studies.

Finally, I would like to thank my family and any friends not already acknowledged for their support and understanding throughout the entire period of my studies.

Dedication

This thesis is dedicated to Karen, who always knows how to make me smile.

Abstract

Recently, Massively Multiplayer Online Games (MMOGs) have received significant popularity. Made up of populations of physically distributed players, MMOGs provide user immersion into the playing environment through realistic graphics and sound. However, many lack real-time communication between users, relying upon text-based and simple VoIP systems. The Immersive Audio Communications for Massively Multiplayer Games (IACMMG) and Dense Interactive Communications Environment (DICE) projects, currently being developed by the Smart Internet Technologies CRC, aim to change this by providing realistic two dimensional voice communications. As MMOGs cater for a global audience, the projects are designed to utilise a network of audio processing servers to which users are connected for the purposes of communication.

These projects face a number of challenges, two of which we address in this thesis. The first is concerned with the finite resources of each server and is thus a question of how to effectively balance user load over the available audio servers. The second problem is concerned with ensuring that users receive acceptable audio Quality of Service (QoS).

In order to address the first problem, a load balancing architecture is proposed. Four load balancing algorithms are designed and implemented and their respective strengths and weaknesses explored.

To address the second problem, we explore how QoS should be defined in an MMOG context. The IACMMG and DICE projects are VoIP based, thus QoS is primarily dependent upon packet delay and loss. We explore techniques for information gathering from packet traffic and use it for equating perceived QoS. This leads to the development of a QoS algorithm designed to measure the QoS of conversations within the virtual environment of the MMOG, reacting to the perceived level of QoS, and moving users to servers capable of providing higher levels of QoS should this be necessary.

The load balancing and QoS aspects of the IACMMG and DICE projects are dependent upon one another, and thus we are left with a problem of two potentially conflicting objectives. To address this we integrate the two architectures, modifying each to make it compatible with the other. The culmination is a complete architecture capable of both load and QoS management.