NETWORKING AND ONLINE GAMES

UNDERSTANDING AND ENGINEERING MULTIPLAYER INTERNET GAMES

Grenville Armitage,
Swinburne University of Technology, Australia

Mark Claypool,
Worcester Polytechnic Institute, USA

Philip Branch,
Swinburne University of Technology, Australia

John Wiley & Sons, Ltd

NETWORKING AND ONLINE GAMES

NETWORKING AND ONLINE GAMES

UNDERSTANDING AND ENGINEERING MULTIPLAYER INTERNET GAMES

Grenville Armitage,
Swinburne University of Technology, Australia

Mark Claypool,
Worcester Polytechnic Institute, USA

Philip Branch,
Swinburne University of Technology, Australia

John Wiley & Sons, Ltd

Telephone (+44) 1243 779777

Email (for orders and customer service enquiries): cs-books@wiley.co.uk
Visit our Home Page on www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons Inc., 111 River Street, Hoboken, NJ 07030, USA

Jossey-Bass, 989 Market Street, San Francisco, CA 94103-1741, USA

Wiley-VCH Verlag GmbH, Boschstr. 12, D-69469 Weinheim, Germany

John Wiley & Sons Australia Ltd, 42 McDougall Street, Milton, Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01, Jin Xing Distripark, Singapore 129809

John Wiley & Sons Canada Ltd, 22 Worcester Road, Etobicoke, Ontario, Canada M9W 1L1

Wiley also publishes its books in a variety of electronic formats. Some content that appears
in print may not be available in electronic books.

Library of Congress Cataloging-in-Publication Data:

Armitage, Grenville.
Networking and online games : understanding and engineering multiplayer
Internet games / Grenville Armitage, Mark Claypool, Philip Branch.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-470-01857-6 (cloth : alk. paper)
ISBN-10: 0-470-01857-7 (cloth : alk. paper)
1. Computer games – Programming. 2. TCP/IP (Computer network protocol)
3. Internet games. I. Title: Understanding and engineering multiplayer
Internet games. II. Claypool, Mark. III. Branch, Philip. IV. Title.
QA76.76.C672A76 2006
794.8′1526 – dc22

2006001044

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN-13: 978-0-470-01857-6
ISBN-10: 0-470-01857-7

Typeset in 10/12pt Times by Laserwords Private Limited, Chennai, India
Printed and bound in Great Britain by Antony Rowe Ltd, Chippenham, Wiltshire
This book is printed on acid-free paper responsibly manufactured from sustainable forestry
in which at least two trees are planted for each one used for paper production.

Contents

Author Biographies

Grenville Armitage Editor and contributing author Grenville Armitage is Director of the Centre for Advanced Internet Architectures (CAIA) and Associate Professor of Telecommunications Engineering at Swinburne University of Technology, Melbourne, Australia. He received his Bachelor and PhD degrees in Electronic Engineering from the University of Melbourne, Australia in 1988 and 1994 respectively. He was a Senior Scientist in the Internetworking Research Group at Bellcore in New Jersey, USA (1994 to 1997) before moving to the High Speed Networks Research department at Bell Labs Research (Lucent Technologies, NJ, USA). During the 1990s he was involved in various Internet Engineering Task Force (IETF) working groups relating to IP Quality of Service (QoS). While looking for applications that might truly require IP QoS he became interested in multiplayer networked games after moving to Bell Labs Research Silicon Valley (Palo Alto, CA) in late 1999. Having lived in New Jersey and California he is now back in Australia – enjoying close proximity to family, and teaching students that data networking research should be fascinating, disruptive and fun. His parents deserve a lot of credit for helping his love of technology become a rather enjoyable career.

Mark Claypool Contributing author Mark Claypool is an Associate Professor in Computer Science at Worcester Polytechnic Institute in Massachusetts, USA. He is also the Director of the Interactive Media and Game Development major at WPI, a 4-year degree in the principles of interactive applications and computer-based game development. Dr. Claypool earned M.S. and Ph.D. degrees in Computer Science from the University of Minnesota in 1993 and 1997, respectively. His primary research interests include multimedia networking, congestion control, and network games. He and his wife have 2 kids, too many cats and dogs, and a bunch of computers and game consoles. He is into First Person Shooter games and Real-Time Strategy games on PCs, Beat-'em Up games on consoles, and Sports games on hand-helds.

Philip Branch Contributing author Philip Branch is Senior Lecturer in Telecommunications Engineering within the Faculty of Information and Communication Technologies at Swinburne University of Technology. Before joining Swinburne he was a Development Manager with Ericsson AsiaPacific Laboratories and before that, a Research Fellow at Monash University where he conducted research into multimedia over access networks. He was awarded his PhD from Monash University in 2000. He enjoys bushwalking with his young family and playing very old computer games.

Acknowledgements

We would like to acknowledge the permissions of, and give special thanks to, a number of copyright owners for the use of their images in this book.

Figures 2.3, 2.4 and 2.8–Photos reproduced by permission of William Hunter, "The Dot Eaters: Videogame History 101", http://www.thedoteaters.com

Figure 3.10 Warcraft™ provided courtesy of Blizzard Entertainment, Inc.

Figure 3.20 is reproduced by permission of Tiffany Wolf

Figure 3.21 is reproduced by permission of Konami

Figures 3.22 and 3.23 are reproduced by permission of Adrian Cheok

Figures 3.24(a) and (b) are reproduced by permission of Wayne Piekarski

Figures 3.6, 3.9, 3.11, 3.14, and 3.15 are reproduced by permission of Electronic Arts

Figures 2.11, 3.4, 3.5, 7.5, 7.6, and 7.7 are reproduced by permission of Id Software, Inc.

Figure 3.12 Pole Position and Figure 3.13 Ridge Racer™ provided courtesy of Namco®

We would also like to acknowledge the work of Warren Harrop and Lawrence Stewart in constructing a large collection of client-side cheat scenarios from which Figures 7.5, 7.6 and 7.7 were selected.

1

Introduction

A lot has happened since 1958 when William A. Hinginbotham used an oscilloscope to simulate a virtual game of tennis. Computing technology has made staggering leaps forward in power, miniaturisation and sophistication. High speed international data networks are part of modern, everyday life in what we call 'the Internet'. Our peculiarly human desire for entertainment and fun has pushed the fusion and evolution of both computing and networking technologies. Today, computer games are sold to an increasingly significant market whose annual revenues already exceed that of the Hollywood movie industry. Multi-player games are making greater use of the Internet and the driving demand for 'better than dial-up' access services in the consumer space. Yet many networking engineers are unfamiliar with the games that utilise their networks, as game designers are often unsure of how the Internet really behaves.

Regardless of whether you are a network engineer, technical expert, game developer, or student with interests across these fields, this book will be a valuable addition to your library. We bring together knowledge and insights into the ways multi-party/multi-player games utilise the Internet and influence traffic patterns on the Internet. Multi-player games impose loads on Internet Service Providers (ISPs) quite unlike the loads generated by email, web surfing or streaming content. People's demand for realistic interactivity creates somewhat unique demands at the network level for highly reliable and timely exchange of data across the Internet – something the Internet rarely offers because of its origins as a 'best effort' service. Game designers have developed fascinating techniques to maintain a game's illusion of shared experiences even when the underlying network is losing data and generally misbehaving.

For those with a background in data networking, we begin with two chapters by Mark Claypool, 'Early Online and Multi-player Games' and 'Recent Online and Multi-player Games', covering the history of computer games and the various ways in which game-related technology has branched out. From the earliest single-player electronic games, through multi-user dungeons and first-person shooters, to today's emerging augmented-reality games and simulation systems, we have come a long way in 40 years. We cover the definition of multi-player networked games and discuss the meaning of peer-to-peer and client–server communication models in the context of game systems. For those readers with a background in game design and development, our next chapter, 'Basic Internet Architecture', provides a refresher and short introduction to the basics of Internet Protocol

(IP) networking. We review the concept of 'best effort' service, IP addressing and the role of transport protocols such as TCP (Transmission Control Protocol) and User Datagram Protocol (UDP) as they pertain to game developers. When you complete this chapter, you will have an understanding of the differences between routing and forwarding, addresses and domain names. You will learn why Network Address Translation (NAT) exists and how it impacts on network connectivity between game players.

Our next chapter, 'Network Latency, Jitter and Loss', should be of interest to all readers. Here we look in detail at how modern IP networks fail to provide consistent and reliable packet transport service – by losing packets or by taking unpredictable time to transmit packets. We discuss how much of this network behaviour is unavoidable and how much can be controlled with suitable network-level technology and knowledge of game traffic characteristics. This leads naturally to Mark Claypool's next chapter, 'Latency Compensation Techniques', where we look at the various techniques invented by game developers to cope with, and compensate for, the Internet's latency and packet loss characteristics. A fundamental issue faced by multi-player online games is that the latency experienced by each player is rarely equal or constant. And yet, to maintain a fair and realistic immersive experience, games must adapt to, predict and adjust to these varying latencies. We look at client-side techniques such as client prediction and opponent prediction, and server-side techniques such as time warping. Compression of packets over the network is introduced as a means to reduce network-induced latency.

Our next chapter, 'Playability versus network conditions and cheats', takes a different perspective. We look at how two separate issues of network conditions and cheating influence player satisfaction with their game experience. First, we look at the importance of knowing the tolerance your players have of latency for any particular game genre. Such knowledge helps game hosting companies to estimate which area on the planet their satisfied customers will come from (and where to place new servers to cover new markets). We discuss existing research in this area and issues to consider when trying to establish this knowledge yourself. Next we look at communication models, cheats and cheat mitigation. Cheating is prevalent in online games because such games combine competitiveness with a sense of anonymity – and the anonymity leads to a lessened sense of responsibility for one's actions. We look at examples of server-side, client-side and network-based cheating that may be attempted against your game, and discuss techniques of detecting and discouraging cheating.

In 'Broadband Access Networks', Philip Branch takes us through a discussion of the various broadband access technologies likely to influence your game player's experiences in the near future. Access networks are typically the congestion point in a modern ISP service; they come in a variety of technologies allowing fixed and wireless connectivity, and have unique latency and loss characteristics. From a high level, we review the architectures of cable modems, Asymmetrical Digital Subscriber Line (ADSL) links, 802.11 wireless Local Area Networks (LANs), cellular systems and Bluetooth.

We then move in an entirely different direction with the chapter 'Where do players come from and when?'. One of the key questions facing game hosting companies is determining where their market exists, who their players are, and where they reside. This has an impact on the time zones over which your help desk needs to operate and the ebb and flow of game-play traffic in and out of your servers. Taking a very practical direction, we first discuss how you can monitor and measure traffic patterns yourself with freely available open-source operating systems and packet sniffing tools. Then we look at existing research on daily and weekly player usage trends, trends in server-discovery probe traffic that hit your server whether people play or not, and note some techniques for mapping from IP addresses to geographical location.

At the other end of the spectrum is the packet-by-packet patterns hidden in packet size distributions and inter-packet arrival times. In ‘Online Game Traffic Patterns’, we look at how to measure traffic patterns at millisecond timescales, and show how these patterns come about in First-Person Shooter (FPS) games – the most demanding interactive games available. It is at this level that network operators need to carefully understand the load being put on their network in order to properly configure routers and links for minimal packet loss and jitter. We review how typical FPS packet size distributions are quite different in the client-to-server and server-to-client directions, and how server-to-client packet transmissions are structured as a function of the number of clients. Overall this chapter provides great insight into the burstiness that your network must support if you wish to avoid skewing the latency and jitter experienced by every player.

Then in ‘Future Directions’, Mark Claypool provides general thoughts on some topics relating to the future of online multi-player games. We particularly focus on the use of wireless technologies, automatic configuration of Quality of Service without player intervention, hybrid client–server architectures, cheaters, augmented reality, massively multi-player games, time-shifting games (where you can start and stop at anytime) and new approaches to server discovery.

Finally, in ‘Setting up online FPS game servers’, we wrap up this book with a practical introduction to installing and starting your own FPS game servers on free, open-source platforms. In particular, we look at the basics of downloading, installing and starting both Wolfenstein Enemy Territory (a completely free team-play FPS game) and Valve’s Half-Life 2 (a commercial FPS). In both cases, we discuss the use of Linux-based dedicated game servers, and provide some thoughts on running them under FreeBSD (both Linux and FreeBSD are free, open-source UNIX-like operating systems available for standard PC hardware).

We hope you will find this book a source of interesting information and new ideas, whether you are a networking engineer interested in games or a game developer interested in gaining a better understanding of your game’s interactions with the Internet.

Grenville Armitage (author and editor)

2

Early Online and Multiplayer Games

In this chapter, we cover some of the history of early online and multiplayer games. Like most computer systems and computer applications, online games evolved as the capabilities of hardware changed (and became cheaper) and user expectations from those games grew to demand more from the hardware.

Besides being interesting in their own right, examining early online and multiplayer game history can help us understand the context of modern network games. We will deal with the following:

- Introduce important early multiplayer games that set the tone for the networking multiplayer games that would follow.
- Describe early network games that often had a centralised architecture, suitable for the mainframe era in which they were developed.
- Provide details on turn-based games that were popular before low latency network connections were widespread.
- End with popular network games that made use of widespread Local Area Network (LAN) technology.

2.1 Defining Networked and Multiplayer Games

By its very definition, a network game must involve a network, meaning a digital connection between two or more computers. Multiplayer games are often network games in that the game players are physically separated and the machines, whether PCs or consoles or handhelds, are connected via a network. However, many multiplayer games, especially early ones were not network games. Typically, such multiplayer games would have users take turns playing on the same physical machine. For example, one player would take turns fighting alien ships while the second player watched. Once the first player was destroyed or when he/she completed the level, the second player would have a turn. Scores for each player were kept separately. For simultaneous multiplayer play, either cooperatively or head-to-head, each player would see their avatar on the same screen or the screen would be 'split' into separate regions for each player. For example, a multiplayer sports game

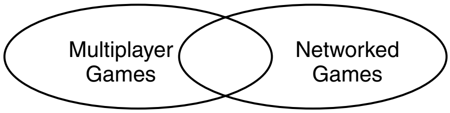

Figure 2.1 The sets of multiplayer games and network games are overlapping, but not subsets or supersets of each other

(a)

- 1960s
  - Era of early multiplayer games
- 1970s and 1980s
  - Era of arcade multiplayer games
- 1990s and beyond
  - Era of on-line, multiplayer games

(b)

1958	Tennis for Two
1961	Space War
1970	Galaxy War
1972	Pong
1978	Atari Football
1993	Doom

Figure 2.2 Timeline overview of early online and multiplayer games. (a) Lists approximate game eras. (b) Lists the release of milestone games mentioned in this chapter

may have each player working one member of opposing teams. The game field could either be entirely seen by both players or the screen would be physically split into the part of the field viewable by each player. Thus, the area of multiplayer games includes some games that are not network games.

On the flip side, some network games are not multiplayer games. A game can use a network to connect the player's machine to a remote server that controls various game-play aspects. The game itself, however, can be entirely a single-player game where there is no direct interaction with other players or their avatars. Early games, in particular, were networked because a player logged into a mainframe server and played the game remotely over a network via a terminal. Even with today's modern computer systems, players can run a game locally on a PC and connect to a server for map content or to interact with Artificial Intelligence (AI) units controlled by a server.

Thus, multiplayer and network games overlap, as depicted in Figure 2.1, but neither fully contains the other.

This sets the stage for discussing the evolution of computer games, starting with early multiplayer games, early networked games and progressing to early, multiplayer networked games (Figure 2.2)

2.2 Early Multiplayer Games

In 1958, William A. Higinbotham, working at the Brookhaven National Laboratory, used an oscilloscope to simulate a virtual game of tennis. This crude creation utilised an overhead view, allowing two players to compete against each other in an attempt to sneak the ball past the paddle of their opponent. Higinbotham called this game *Tennis*

Figure 2.3 William Hinginbotham invented the multiplayer game *Tennis for Two* using an oscilloscope. Reproduced by permission of William Hunter.

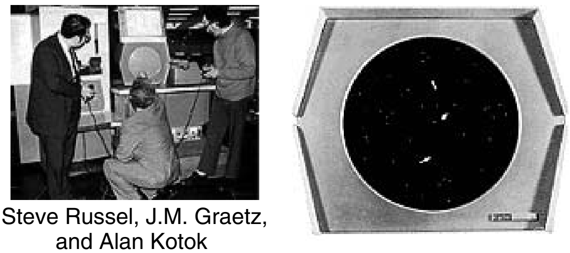

Figure 2.4 *Spacewar* was the first real computer game, and featured a multiplayer duel of rocket ships. Reproduced by permission of William Hunter.

for Two [PONG] and it was perhaps the first documented multiplayer electronic game (Figure 2.3).

However, while definitely a multiplayer game Tennis for Two used hard-wired circuitry and not a computer for the game play. The honour of the first real computer game goes to *Spacewar*, which was designed in 1961 to demonstrate a new PDP-1 computer that was being installed at MIT (Figure 2.4). In Spacewar, two players duelled with rocket ships, firing torpedos at one another. Spacewar had no sound effects or particle effects, but illustrated just how addictive compelling game play could be even without fancy graphics. It even showed sophisticated AI was not needed since real intelligence, in the form of a human opponent, could enhance game play in both competitive and cooperative modes.

Soon after its creation, Spacewar programmers were discovering the tradeoffs between realism and playability, adding gravity, star maps and hyperspace. Although the price of the PDP-1 (then over $100 000) made it impossible for Spacewar to be a commercial success, it had lasting influence on the games that followed, including subsequent multiplayer and networked games.

A version of Spacewar that was a commercial success was *Galaxy War*, appearing on campuses in Stanford in the early 1970s (Figure 2.5). It may have been up and running even before the far more popular *Pong* by Atari.

Figure 2.5 *Galaxy War*, early 1970s. Reproduced by permission of Id Software, Inc.

2.2.1 PLATO

Perhaps the first online network community was *PLATO* (which initially was supposedly not an acronym for anything, but later became an acronym for Programming Logic for Automatic Teaching Operations) that had users log into mainframe servers and interact from their terminals [PLATO]. PLATO included various communication mechanisms such as email and split-screen chat and, of course, online games. Two popular PLATO games were *Empire*, a multiship space simulation game and *Airfight*, what may have been the precursor to Microsoft flight simulator. There was even a version of Spacewar written for PLATO. These early online games were networked only in the sense that a terminal was connected to a mainframe, much like other interactive applications (such as a remote login shell or an email client) of the day. Thus, the game architecture featured a 'thin' game client (the terminal) with all the computation and communication between avatars taking place on the server.

The network performance of early systems was thus determined by the terminal communication with the mainframe server via the protocol used by the *Telnet* program [RFC854]. A Telnet connection uses the Transmission Control Protocol (TCP) connection to transmit the data users type with control information. Typically, the Telnet client will send characters entered by keystrokes and wait for the acknowledgment (echo) to display them on the screen. From the user perspective, a typical measure of performance is the *echo delay*, the time it takes for a segment sent by the source to be acknowledged. Having characters echoed across a TCP connection in this manner can sometimes lead to unpredictable response times to user input.

2.2.2 MultiUser Dungeons

MultiUser Dungeons (MUDs) rose to popularity shortly after PLATO, providing a virtual environment for users to interact with the world and with each other with some game-play elements. MUDs are effectively online chat sessions with game-play elements and structure; they have multiple places for players to move to and interact in like an adventure game, and may include elements such as combat and traps, as well as puzzles, spells and even simple economics. Early MUDs had text-based interfaces that allowed players to type in basic commands, such as 'go east' or 'open door' (Figure 2.6). Typically, characters