Introduction

A lot has happened since 1958 when William A. Higinbotham 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 gaming-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.
Networking and Online Games: Understanding and Engineering Multiplayer Internet Games

(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)