UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Mirko Sužnjević

MODELLING OF NETWORK TRAFFIC FOR MULTIPLAYER ROLE PLAYING GAMES BASED ON USER BEHAVIOUR

DOCTORAL THESIS

Zagreb, 2012.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Mirko Sužnjević

MODELIRANJE MREŽNOG PROMETA VIŠEKORISNIČKIH IGARA S PREUZIMANJEM ULOGA TEMELJENO NA KORISNIKOVOM PONAŠANJU

DOKTORSKI RAD

Zagreb, 2012.

The doctoral thesis was completed at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Telecommunications.

Mentor: Professor Maja Matijašević, Ph.D.
University of Zagreb,
Faculty of Electrical Engineering and Computing

The thesis has 194 pages.

Thesis number:

The dissertation evaluation and defence committee:

1. Professor Igor Pandžić, Ph.D.
   University of Zagreb Faculty of Electrical Engineering and Computing

2. Professor Mladen Kos, Ph.D.
   University of Zagreb Faculty of Electrical Engineering and Computing

3. Professor Saša Dešić, Ph.D.
   Ericsson Nikola Tesla Zagreb

4. Asst. Professor Fabio Ricciato, Ph.D.
   Universita del Salento, Lecce Italy

5. Asst. Professor Lea Skorin-Kapov, Ph.D.
   University of Zagreb Faculty of Electrical Engineering and Computing

Date of dissertation defence: 11th May 2012

For Antigona and Vukasil...

Acknowledgements

First of all, I would like to thank my mentor prof. Maja Matijašević, for all the help, guidance, and understanding she has given me during this work. Also, I want to thank her for giving me an opportunity to combine my passion about networked games with my work which made working on this thesis much more enjoyable.

I want to thank all of my colleagues from the Department of Telecommunications that I have collaborated with over the years. I want to give special mention to Ogi for all the help provided especially in the start of my research, Lea for the corrections, Valter for the great assistance regarding Linux and Free BSD, Tomo, Ana, and Vedran for all the advices, young folks Iva, Darijan, Marija and Kruno for all the fun and discussions, Saška for the long teaching cooperation, Marin and Krešo for the witting tips, and also former co-workers Agata, Ana, Jura, Matko, and Puh.

This work would not be possible without the assistance of my "virtual friends" who were constantly bugged by my research requests. Kuba, Daud, Dada, Neb, Zmaj, Divljak, and Ivek: you are the proof that while game worlds may be virtual, they create true friends. Special thanks to my best friend and love Tanja as she wholeheartedly helped me during all steps of my research, not just by participation, but also with unending support, understanding, and love.

I have had the honour of guiding some really talented and smart students from who I want to specially thank Ivana and Borna who were actively involved in the creation of this thesis.

Finally, nothing of this would be possible without the love, support, and encouragement of my family who are everything to me, sister Milena, and parents Jelena and Nikola. Also, a thank to all of my broader family, and especially aunt Nada.

Abstract

While network traffic characteristics of Massively Multiplayer Online Role-Playing Games (MMORPG) are known to be very variable and somehow linked to game dynamics and user behaviour, the actual relationships have, thus far, not been described in a comprehensive and unified way, which could sucessfully be applied for synthetic traffic generation. This thesis aims to fill this gap by proposing a novel source based MMORPG traffic model, which explains and captures observed variations of traffic characteristics.

The proposed model takes user behaviour at the application level as a starting point. As networked virtual worlds of MMORPGs are very complex, there is a wide variety of in-game situations based on "what users do" in the virtual world, which reflect onto network traffic in different ways. The model focuses on capturing, recognizing, understanding, and describing the relationships between user actions and network traffic. A classification proposed in this thesis distinguishes between the following user action categories: Trading, Questing, Player versus player combat, Dungeons, and Raiding. For each action category, a traffic model capturing the statistical characteristics of network traffic has been developed and validated through network traffic measurements. Client and server application protocol data unit size and interrarival time have been modeled by a combination of several distributions, including Weibull, Normal, Lognormal, Largest Extreme Value, and Deterministic. Next, the player behaviour over a single gaming session has been studied and modelled based on the defined action categories by using a first order Markov chain. Finally, the aggregate behaviour of all active users on a single MMORPG server has been described. The arrival of new players and departure of leaving players are modeled as a Homogeneous Poisson Process (HPP). Based on the proposed model, a functional architecture of a MMORPG traffic generator based on player behaviour, called UrBBaN-Gen, has been developed and implemented by using Java, Python and bash scripts, together with the open source software traffic generator – Distributed Internet Traffic Generator (D-ITG). Synthetic traffic generated by UrBBaN-Gen has been compared with independent empirical traces, and it has been demonstrated that the characteristics of the generated traffic closely follow the real traffic. Also, the model has been compared with the models found in literature, and its advantages over the existing models have been shown.

The contribution of this thesis may be summarized as follows:

- Classification of user actions in the virtual world of MMORPGs, and characterization of associated network traffic;
- User behaviour model based on categories of user actions, motivational parameters, and identified behavioural patterns on application level; and
- Architecture and implementation of traffic generator based on the model and verification of the model through comparison of synthetic and real traffic.

Keywords Network traffic modelling, source network traffic model, communication network, Internet, user behaviour modelling, networked virtual environment, network traffic generation, online game, MMORPG

Sažetak

Karakteristike mrežnog prometa koji generiraju višekorisničke igre s preuzimanjem uloga su vrlo promjenjive. Varijacije u mrežnom opterećenju mogu dostizati i razliku od deset puta između najniže i najviše vrijednosti. Treba uzeti u obzir da je riječ o prosječnim vrijednostima u vremenskim periodima na razini minuta, pa čak i sati.

U ovoj disertaciji predložen je izvorišni model mrežnog prometa. Izvorišni modeli mrežnog prometa temelje se na ponašanju aplikacija koje se nalaze na krajnjim točkama mreže. Predloženi model objašnjava i obuhvaća uočene varijacije karakteristika mrežnog prometa.

Model se temelji na ponašanju korisnika unutar višekorisničkih igara s preuzimanjem uloga. Kao studijski slučaj koristi se umrežena igra *World of Warcraft* proizvođača *Activision Blizzard*. Kako su virtualni svjetovi ovih igara vrlo složeni, a broj interakcija i situacija u kojem se korisnik može naći velik, za potrebe modeliranja predložena je klasifikacija korisničkih akcija. Predložene kategorije korisničkih akcija su: trgovanje, traganje ili izvršavanje zadataka, borba između igrača, napadanje tamnica i masovno napadanje tamnica. Različitost identificiranih kategorija ponašanja potvrđena je kroz mjerenje i usporedbu karakteristika mrežnog prometa pojedinačne kategorije. Za svaku kategoriju kreiran je matematički model mrežnog prometa koji se sastoji od kompleksnih statističkih distribucija koje opisuju veličinu jedinica podataka koji se šalju na razini aplikacije (ne na razini, primjerice, datagrama protokola IP), te međudolazna vremena između dva uzastopna slanja jedinica podataka.

Na temelju identificiranih kategorija provedena su mjerenja ponašanja korisnika pomoću kojih je kreiran model ponašanja korisnika. Kreirani model opisuje ponašanje pojedinačnog korisnika, ali i zbirno ponašanje svih korisnika na razini usluge. Također, proučen je odnos između psihološke motivacije korisnika i njihovog ponašanja na razini aplikacije.

Razvijeni modeli ponašanja korisnika i njihov utjecaj na mrežne karakteristike prometa objedinjeni su kroz funkcijsku arhitekturu programskog generatora mrežnog prometa utemeljenog na ponašanju korisnika (engl. User Behaviour Based Network Traffic Generator - UrBBaN-Gen.). UrBBaN-Gen čine tri programska modula: 1) simulator korisničkog ponašanja, 2) sustav za kontrolu distribuiranog generiranja prometa te 3) generator mrežnog prometa. Simulator korisničkog ponašanja razvijen je primjenom programskog jezika Java, a sustav za kontrolu distribuiranog generiranja prometa primjenom Jave te Python i Bash skripti. Generator mrežnog prometa temelji se na softveru otvorenog koda *Distribuirani internetski mrežni generator* (engl. *Distributed Internet Traffic Generator – D-ITG*), koji je modificiran kako bi se u njega ugradili modeli prometa predloženih kategorija korisničkih akcija.

Razvijeni model mrežnog prometa je uspoređen s modelima za istu uslugu poznatima u literaturi te su pokazane njegove prednosti. Također, model je verificiran kroz usporedbu sintetičkog, računalno generiranog prometa sa stvarnim prometom, te je pokazano da karakteristike generiranog prometa zadovoljavajuće sliče karakteristikama stvarnog prometa.

Znanstveni doprinos disertacije je sljedeći:

- Klasifikacija tipova korisničkih akcija unutar virtualnih okruženja igara s preuzimanjem uloga i karakterizacija pripadajućeg mrežnog prometa,
- Model ponašanja korisnika, temeljen na kategorijama korisničkih akcija, motivacijskim parametrima te identificiranim uzorcima ponašanja na razini aplikacije, i
- Arhitektura i programska izvedba generatora mrežnog prometa zasnovanog na modelu i verifikacija modela kroz usporedbu sintetiziranog i stvarnog mrežnog prometa.

Ključne riječi Modeliranje mrežnog prometa, izvorišni model mrežnog prometa, modeliranje korisničkog ponašanja, komunikacijska mreža, Internet, umreženo virtualno okruženje, generiranje mrežnog prometa, igra, višekorisnička igra s preuzimanjem uloga.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the background and motivation for this thesis, followed by the problem definition and method of solution, and lastly, the main research contributions.

1.1 Background and motivation

The computer and video games industry (i.e., gaming industry) is one of the important, high-tech economy drivers with an annual revenue of over 25 billion U. S. dollars (USD), directly and indirectly employing 120,000 people with average salary of 90,000 USD (data for United States of America (USA)) [3]. The fact that 72% of American households play computer or video games [3] testifies how significant video and computer games have become in everyday life. Global trends regarding player numbers are similar as estimats show that games are played by 42.5% of Germans, 39.4% of French, and 50% of the United Kingdom population [4]. Currently, one of the major trends in the gaming industry is a shift towards online multiplayer games which is enabled by an increased accesibility of broadband Internet connections [5]. This shift is evident even in the game distribution trends, as sales of online downloadable content (DLC) surpassed the boxed product sales in the USA in 2010 [4].

Economic growth of the online gaming industry reflects on the network. As the number of online players increases, so does the volume of game generated traffic. The estimation from Cisco is that the global game traffic will grow with an annual rate of 37% in the period from 2009 to 2014 [6], which represents the second largest growth after video related categories.

While the gaming traffic still presents a relatively low overall load on the network, compared to video and file sharing, it is a category of traffic requiring high Quality of Service (QoS). In order to immerse themselves into virtual worlds of online games, players need more or less "real time" responsivness (depending on the game genre). As reflected on the network level, this means primarily low latency and low jitter values, but also low data loss rates. Thus, adequate traffic models are necessary to correctly predict the load and to dimension the network according to given techology and business needs.

Massively Multiplayer Role-Playing Games (MMORPGs) are a game genre in which a player takes a role of an virtual character (i.e., avatar) which represents him/her in the persistent virtual world. Through the avatar, the player interacts with the virtual world in many ways (e.g., exploring, fighting, trading, etc.). MMORPGs are one of the fastest growing game genres and are played by over 11% of all player population in USA [3]. For game publishers, MMORPGs present a very lucrative field of computer entertainment industry, as they provide very good and continuous revenues. According to the report of the NewZoo, an international full service market research and consulting firm completely focused on the games industry, revenues from MMORPGs in 6 countries (USA, Germany, United Kingdom (UK), France, Belgium, and The Netherlands) reached 2.98 billion USD in 2009 [4]. The number of MMORPG titles in the market is growing constantly and game providers are trying to keep the existing player base, as well as to attract new players. In order to do so, game publishers depend on adequate QoS, since virtual worlds need to be responsive in order to be realistic. If the network is not performing adequately, the players can not immerse themselves into the virtual world (hence the complaints about "lag" by the player population). For example, recently there was a revolt of the customers of a network provider in the UK [7]. This problem arose as a popular MMORPG introduced a new version which changed network traffic properties so much that the traffic management system of the network provider had problems with correctly labelling that traffic as gaming class. This resulted in significantly lower QoS levels for players who massively responded by switching network provider. In order to provide the best suitable network service, traffic must be constantly studied, analysed, and modelled.

Traffic characteristics of MMORPGs vary significantly over time on both client and server sides, due to differences in the application states [8, 9, 10, 11, 2]. These variations are not fully understood, and not adequately described by traditional traffic modelling approaches based on observing and “mimicking” the statistical properties of real traffic traces. In order to model the network traffic, it is necessary to understand and describe the relationship between application state and network traffic. In other words, players’ behaviour and its influence on aggregate traffic characteristics must be taken into account.

1.2 Problem statement

As MMORPGs are a relatively recent phenomenon, there are still many open research issues regarding understanding and modelling the corresponding network traffic. There are, in general, two approaches to network traffic modelling: 1) packet-level replay based on statistical properties derived from real traffic measurements, and 2) source level models, which start from the behaviour of the network application. This thesis adopts the latter approach. MMORPGs are complex applications, involving many players, many situations and set-ups in the virtual world, and a wide range of user interactions. Thus, to devise a source based model of a MMORPG, all these need to be taken into account. What is missing – as will be shown in the literature overview section – is the following:

- There is no generally accepted definition of the “virtual world state” or “application state” for MMORPGs;
- There is no classification of virtual world states which can be used as a base for player behaviour measurements and modelling; and
- There is no clear understanding of how the network traffic relates to virtual world state and player behaviour patterns, i.e., what players do in the virtual world.

Existing solutions in literature cover certain parts of the problem, such as characterizing different states of the virtual world, or modelling the player session lengths, however, there is no comprehensive solution which comprises all aspects of the MMORPG network traffic. Therefore, there is a strong need for a source based model which would incorporate influences from all aspects which shape the MMORPG network traffic, from virtual world state influence on single flow characteristics, to player behaviour patterns and their influence on aggregate traffic. Also, for the purpose of verification, the proposed model needs to be implemented, and generated traffic compared to empirical data.

1.3 Summary of contributions

The contributions of this thesis may be summarized as follows:

- Classification of user actions in the virtual world of MMORPGs, and characterization of associated network traffic;
- User behaviour model based on categories of user actions, motivational parameters, and identified behavioural patterns on application levels; and,
- Architecture and implementation of traffic generator based on the model and verification of the model through comparison of synthetic and real traffic.

1.4 Thesis structure

The structure of the thesis follows the methodology depicted in Figure1.1. First, categories of application level user behaviour need to be identified. Based on the these categories, measurements of both traffic and application level behaviour can be performed. Results of the measurements are transferred into models of network traffic and user behaviour. Finally, both models are combined into a source based network traffic model. Using this generic approach is application independent.

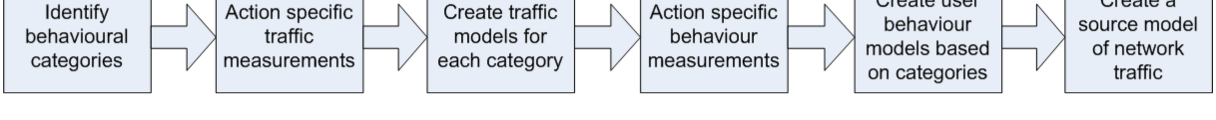


Figure 1.1: Applied methodology [1]

The thesis is structured as follows. After this introductory section, Chapter 2 explains MMORPGs in detail, covering their history, architecture, and basic concepts. Chapter 3 covers the current research efforts in measuring, characterizing, and modelling the network traffic, with respect to problems specific for MMORPGs. Research results regarding player behaviour are presented, with respect to a variety of parameters, such as session duration, player movement, and avatar distribution in the virtual world. Also, works exploring the relationship between the application and network level are presented. Lastly, a survey of open source software traffic generators is presented. The following chapter, Chapter 4, examines user behaviour in the virtual world, and presents user behaviour action categories. The proposed categorisation is validated through network traffic measurements, and traffic models for each category are established. Chapter 5 comprises the details about measurement and modelling of player behaviour in terms of the action categories defined in the previous section. Modelling of a single session is presented first, followed by the aggregate behaviour of all active users on a single MMORPG server. A functional architecture and software implementation of a MMORPG traffic generator based on player behaviour (UrBBaN-Gen) is presented in Chapter 6. Chapter 7 presents the conclusions, summarizes the contributions, and notes some open issues for further research work.

Chapter 2

Massively Multiplayer Online Role-Playing Games

Massively Multiplayer Online Role-Playing Games (MMORPGs) are a genre of role-playing computer games in which a very large number of players interact with one another within a virtual game world [12]. MMORPGs are based on the concept of Networked Virtual Environments (NVEs) and are currently one of the most popular applications of NVEs. An NVE may be defined as a software system in which multiple users interact with each other in real time, even though those users may be located around the world [13]. User are graphically displayed in the NVE as virtual characters or avatars. Each user stores a copy of the virtual environment on his/her computer. The data about users' movements, and interaction with each other and the virtual environment, is transferred over the network in order to synchronize copies amongst all users.

2.1 Terminology

The term MMORPG was crafted by Richard Garriott, the creator of *Ultima Online*, in 1997 [14] while previously this game genre was referred to as "graphical MUDs" (i.e., multi-user dungeons).

The attributes *Massively* and *Multiplayer* referring to online games indicate that a large number of players are concurrently present in the networked virtual world. The numbers vary based on the game architecture and may span from several hundreds (e.g., a WoW server [15]) up to tens of thousands (e.g., World of Tanks [16]). On the other hand, the number of players in other genres of games which support multiplayer mode, such as First Person Shooter (FPS) games or Real Time Strategy (RTS) games, go up to several tens of players. In the recent years, following the success of MMORPGs, other genres also aim at becoming massively multiplayer, though with less market success so far. Additional aspects which indicate the massive scale of MMORPGs are the size and persistency of the virtual world. Virtual worlds of MMORPGs can be very large in terms of virtual space (e.g., estimated size of a virtual world of WoW is around 100 square kilometers, without virtual space added in expansion packs [17], while Lord of the Rings Online is estimated to have 50 square kilometers of virtual space [18]). Persistency of the virtual world indicates that the virtual world continues to exist and evolve, even if the player is not a part of it. This is different, for example, from virtual worlds of FPS games (i.e., an FPS map), where, once the player leaves the game, he/she is no longer relevant to the game world. While the game continues for other players, the virtual world is static, with no changes of relevance to the player who is currently not participating. On the other hand, in MMORPGs, while the player is not in the virtual world, the world continues to change; for example, economy aspects are changing (e.g., auctions on virtual items), progress of other players, and also the player's abilities and *cooldowns* (i.e., the minimum length of time that the player needs to wait after using an ability before they can use it again).

The term *Online* in MMORPG indicates using the Internet to provide the connectivity to the virtual world. While many multiplayer games can be played without an Internet connection (e.g., in a Local Area Network (LAN) mode), MMORPGs are played exclusively over the Internet. This is due to the architecture of the MMORPG (i.e., all commercial MMORPGs use a client – server architecture, in which only the provider of the game hosts the servers), and also due to the need to connect a large number of players, which can only be done through the Internet.

A *Role-Playing Game* (RPG) is a game in which players assume the roles of characters in a fictional setting. RPGs can be pen and paper, live action, and electronic (computer). Computer RPGs and MMORPGs have evolved from pen and paper versions. The most notable pen and paper RPG was *Dungeons & Dragons* (D&D), built based on the adventures in J.R.R. Tolkien's novels *The Hobbit* and *The Lord of the Rings* series. In 1974, a company *Tactical Studies Rules* financed and published the Dungeons & Dragons game which was the first tabletop role-playing game [19]. D&D had significant influence on the evolution of computer RPGs which have taken over many of the major D&D concepts.

2.2 Basic concepts

All MMORPGs have certain common characteristics, regardless of the game theme:

- Health of the virtual character,
- Abilities of the avatar (virtual character),
- Role-taking,
- Progression,
- Customization, and,
- Social interaction.

Health of a virtual character is usually determined through the percentage of "health points", or current "amount" of health. Each entity in the virtual world, whether the (human) player or a Non-Player Character (NPC), has a certain number of health points. In combat, which is performed through certain damaging abilities (e.g., hitting an enemy with a sword), health points are reduced, and when they fall to zero, the entity dies. Health points can be restored through several means, such as healing abilities, resting, etc. Other than health points, the virtual characters commonly have another set of points, which determine how many actions of particular type they can perform. For example, "magic" users commonly have "mana points", or mana (i.e., magical energy points), which determine how many spells they can cast. When the mana is spent, it needs to be restored in some way (e.g., drinking mana potions, resting, etc.).

Abilities of the avatar are usually a set of skills through which avatars interact with the virtual world. Abilities can be various, and can be classified as non-combat and combat related. For example, "mage" class in WoW has the "fireball" ability, which hurls a ball of fire onto the enemy and damages it. Part of the in-game progression process is learning how and when to use a certain ability for greatest gain.

Role-taking of a virtual character can be studied from several standpoints. "Roleplaying" can be considered as a improvisational theatre, with the participants impersonating characters in unscripted situations. In other words, the human players are acting their virtual characters. The character's personality, goals, morals, and quirks may resemble that of the human player, or be completely different. Whichever the setup, roleplayers recognize a boundary between what is "in character" and what is "out of character". Taking a functional role in a fantasy world is reflecting on the role which the character is going to play in a group-based combat. The following types of roles are common:

- *Tanks* are characters whose role is to redirect enemy attacks toward themselves in order to protect other characters or units. Tanks usually have a way of averting the incoming damage (e.g., they rely on large amounts of health or armour, or alternatively evasiveness and misdirection of damage);

- *Healers* are characters responsible for looking after the health of other players or units. Through their abilities they can significantly prevent damage or increase health points; and

- *Damage dealers* are characters which specialize in inflicting damage (often referred to as "DPS", meaning Damage Per Second).

In Figure 2.1 class selection from MMORPG *AION* by *NCSoft* is depicted. The player can choose to be a *warrior* (tank), *scout* (damage dealer), *mage* (damage dealer), and *priest* (healer). The difference between mage and scout is that one is focused on dealing damage from a distance, while the other is doing melee damage (i.e., damage that requires to be close to the target). The three roles listed above are primary and common to almost all MMORPGs. Depending on the type of MMORPG, there can be additional roles, such as support (i.e., characters which provide extra benefit for all other roles), or crowd control (i.e., those characters who temporarily control the hostile NPCs, also called mobs, and make them lose control of actions and abilities).

Progression is present in all MMORPGs on different levels and scales. Virtual characters primarily progress in terms of *levels* indicating how much experience the virtual character has

Figure 2.1: Class selection screen in AION

gathered. With new levels, players are, in general, more powerful and can also gain new and more powerful abilities. There are several ways to gather experience: combat with mobs, doing quests (i.e., tasks assigned by friendly NPCs which upon completion provide rewards in terms of experience and/or virtual items), exploring, trading, gathering and making virtual items, fighting with other players, etc. Players start at level one, and progress towards the level cap which differs across different MMORPGs. This is the most common type of progression. Some games (e.g., *Darkfall* by *Aventurine SA*) do not use a level system, but a similar, *skill point system* in which the players improve their skills by using them. Another type of progression is *equipment based progression*, in which players acquire better virtual items (e.g., weapons, armour), which make them better at performing their roles (e.g., a player with a better sword will deal more damage). Virtual items can be acquired as rewards from quests, can be created by players, and can be "looted" from the corpses of mobs (i.e., after a mob has been defeated, virtual items can be taken from its body). Mobs which are significantly harder to defeat, and require a large group of players are commonly called "bosses", and killing them provides the best rewards. Also, there are types of progression which may not have the impact on the actual power of the virtual character. A player may progress in terms of economic strength or accumulation of virtual items and currency. Also, player can progress in ranking with respect to other players (e.g., some players are ranked in player versus player combat). In the end, players may progress in terms of the "self contained goals" in the virtual world, as some games keep track of those. Achieving those goals may (or may not) provide a reward of some sort. As an example, in WoW there is an achievements system for all aspects of the game (e.g., there is an achievement to defeat certain boss, to eat one hundred (virtual) chocolate bars, or just to explore all areas of the virtual world).

Customization of virtual characters can be done in various aspects and it highly depends on the specific game. Each MMORPG contains some form of visual customization of parameters, such as avatar's hairstyle, skin colour, eye colour, etc. Some games focus on detailed visual formation of avatars, such as AION, in which almost every parameter of the character's appearance can be modified, while others, such as WoW, do not have such detail in character customization, but still offer a sufficiently large number of appearance options (i.e., number of possible avatar permutations in WoW is around 1.25 million [20]).

Social interaction is a very important aspect of MMORPGs and one of the main components leading to their significant market success. Virtual worlds provide anonymity, as each player is displayed with a self-designed virtual character. This yields a potential positive side effect of difficult stereotyping, as all players have an equal ability to design their avatars, regardless of their initial appearance [21]. Players can form *friend lists* which enable easier finding of, and playing with, other friendly players, or *ban lists*, which help them ignore problematic players. In-game player associations can be formed, and are commonly referred to as *guilds*, *alliances*, or *clans*. Guilds can have various structures, goals, and lifespans, depending on various factors [22].

2.3 History and timeline

This section briefly reviews the games considered most relevant for the evolution of MMORPGs and their main contribution to the genre. In Figure 2.2 games considered predecessors of MMORPGs are displayed in a time line.

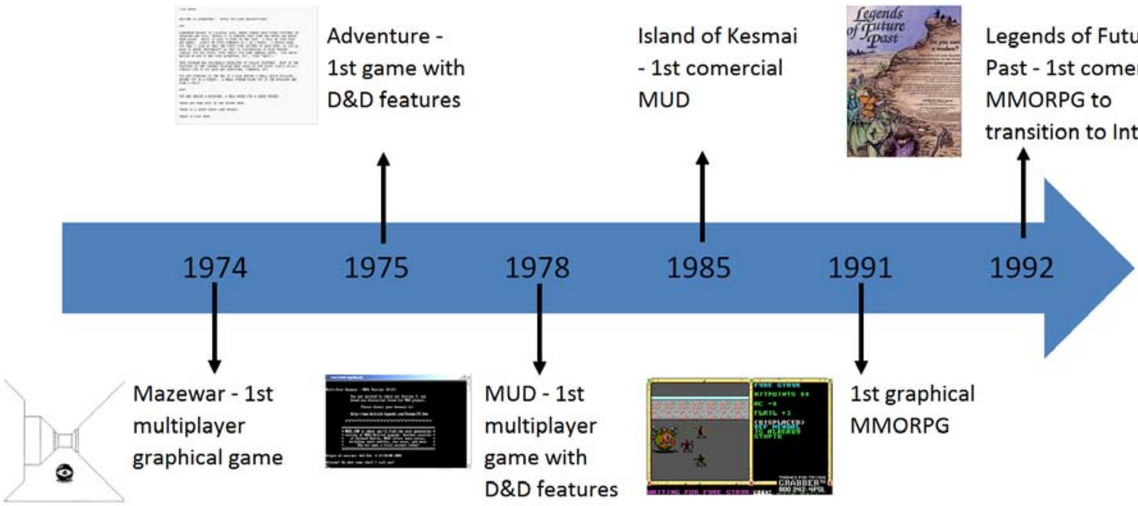


Figure 2.2: Time line of MMORPG predecessors

The first game comprising a multiplayer graphical virtual world was *Mazewar* created in 1974. Mazewar was a FPS game in which the players roamed the maze and tried to shoot each other. The initial implementation was over a serial cable, but when one of the authors began attending Massachusetts Institute of Technology (MIT) in 1974, the game was enhanced so that it could be played across the Advanced Research Projects Agency Network (ARPANET), the forerunner of the modern Internet [23].

Adventure, created in 1975 by Will Crowther, was a first game containing many D&D features and elements and can be viewed as an ancestor of MMORPGs.

In 1978 Roy Trubshaw, a student at Essex University in the UK, started working on a multi-user adventure game he named *MUD* (Multi-User Dungeon). Richard Bartle, a fellow student at Essex University, took over the development of MUD in in 1980. MUD, better known as Essex MUD and MUD1 in later years, ran on the Essex University network until late 1987 [23].

The first commercial MMORPG appeared in 1985 for a price of 12.00 USD per hour via the CompuServe online service. This game was *Island of Kesmai* designed by Kelton Flinn and John Taylor and it could run up to 100 players [23].

The first graphical MMORPG was Neverwinter Nights created for PC by designer Don Daglow and programmer Cathryn Mataga. The game was running from 1991 to 1998 on AOL for a price of 6.00 USD per hour to play [23].

The first commercial text-based MMORPG to make the transition to the Internet from a

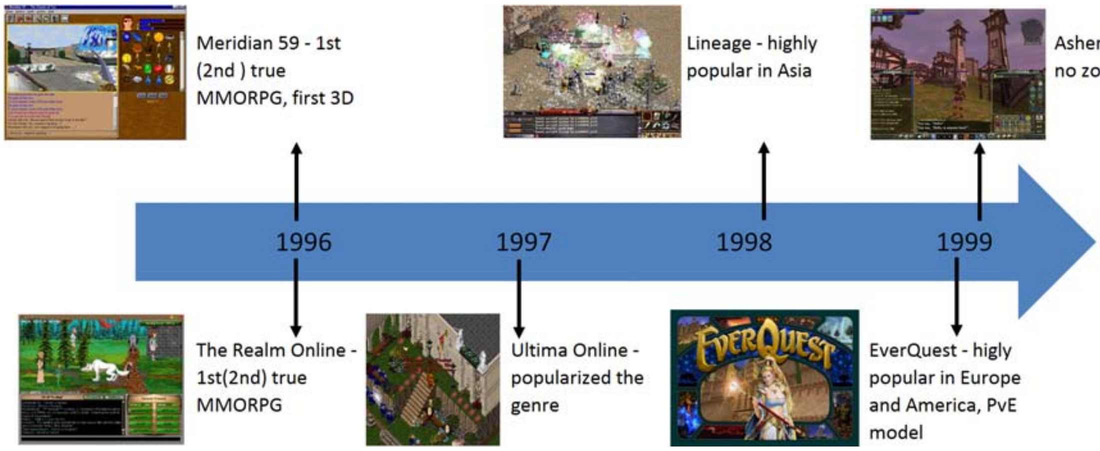


Figure 2.3: Time line of 1st generation MMORPGs

proprietary network provider was *Legends of Future Past* released in 1992. Also, Legends of the Future Past was one of the first titles to have featured professional Game Masters who conducted online events [23].

While there is significant discussion about the history of MMORPGs, the general consensus is that so far there were two generations of MMORPGs, with a third expected soon [24]. They are briefly explained next.

2.3.1 First generation of MMORPGs

The first generation of MMORPGs had the first graphical engines similar to the computer RPGs and friendly user interfaces without the text commands. These online role-playing games allowed many players to simultaneously play in the same universe. From an online perspective, these games generally had only text-based MUDs to look up to as models. In Figure 2.3, the most significant MMORPGs of the 1st generation are displayed together with their main contributions to the genre.

One of the first Internet MMORPGs were *The Realm Online* by *Sierra Online* and *Meridian 59* by *3DO* both launched in 1996. While Meridian 59 was launched earlier, The Realm Online had beta testing phase prior to that, so there are debates which of the these two games was the first “true MMORPG” i.e., comprising all characteristics: massive number of users, Internet for connectivity, and graphical virtual world [23, 24, 25]. While The Realm Online had a 2D graphical interface, Meridian 59 had a 3D graphical interface and was a first game for which the term “massively multiplayer” was used [25].

Ultima Online by *Origin Systems* was released in September 1997, and is widely acknowledged as a MMORPG which made the genre popularity widespread [26] as it quickly gathered 100,000 subscribers. It featured 3D isometric/third-person graphics, and was set in the already popular Ultima universe (RPGs placed in Ultima universe had around 5 million players).

In 1998, *Lineage* by *NCsoft*, was released in South Korea. Lineage was a huge market success and gained millions of subscribers in Korea and Taiwan.

EverQuest, launched in March 1999 by *Verant Interactive* (later acquired by Sony Online Entertainment) was the most commercially successful MMORPG in the USA for five years. Rather than on Player versus Player (PvP) combat, EverQuest focused on player cooperation against computer controlled entities (i.e., Player versus Environment (PvE) model).

In 1999, following Ultima Online and EverQuest, *Asheron's Call* by *Turbine Entertainment* was released. Unlike many other games in the genre, Asheron's Call had no zones in the game, which means that players could cross the world on foot without loading screens or invisible barriers and any terrain that can be seen in the distance was a real object in the world. Together, Ultima Online, EverQuest, and Asheron's Call are sometimes referred to as the original “big three” of the first generation of MMORPGs.

2.3.2 Second generation of MMORPGs

The second generation of MMORPGs mostly did not introduce radically new innovations and the majority of them copied the concepts of the first generation MMORPGs. Rather than revolution, the second generation evolved the genre, improving many aspects, especially regarding graphics, and interface. As there have been many game titles, this section focuses on only the few, most significant ones, which introduced important new concepts to the genre. In Figure 2.4 and Figure 2.5 the time lines with the most notable MMORPGs from this period are shown.

Dark Age of Camelot (DAoC) by *Mythic* was released in beta version in early 2001. It introduced the concept of Realm versus Realm combat which enabled huge Player versus Player (PvP) battles. PvP battles are focused on the combat between players, and not on the combat between players and NPCs. Additionally, the time to level a character was significantly reduced

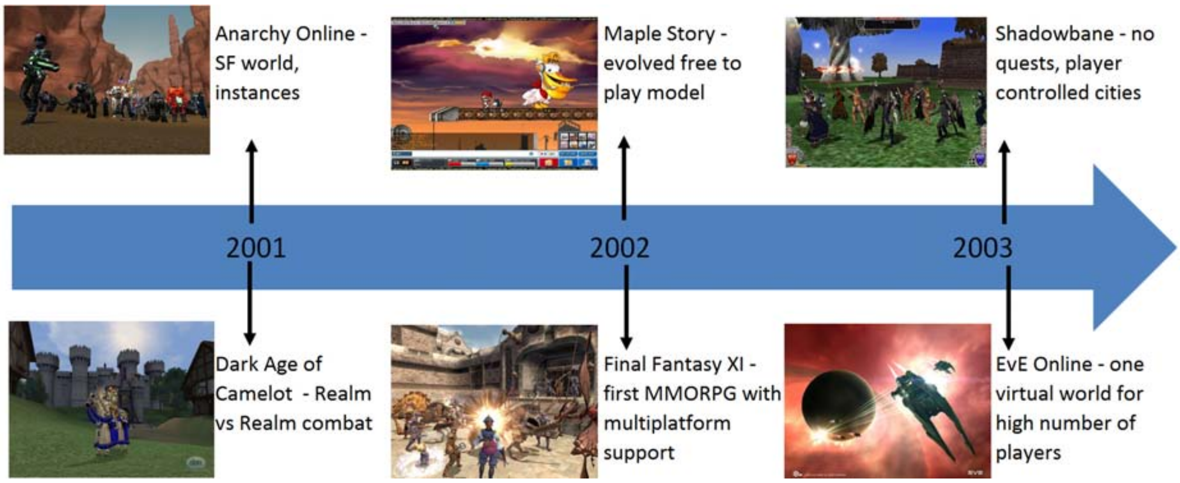


Figure 2.4: Time line of 2nd generation MMORPGs (part1)

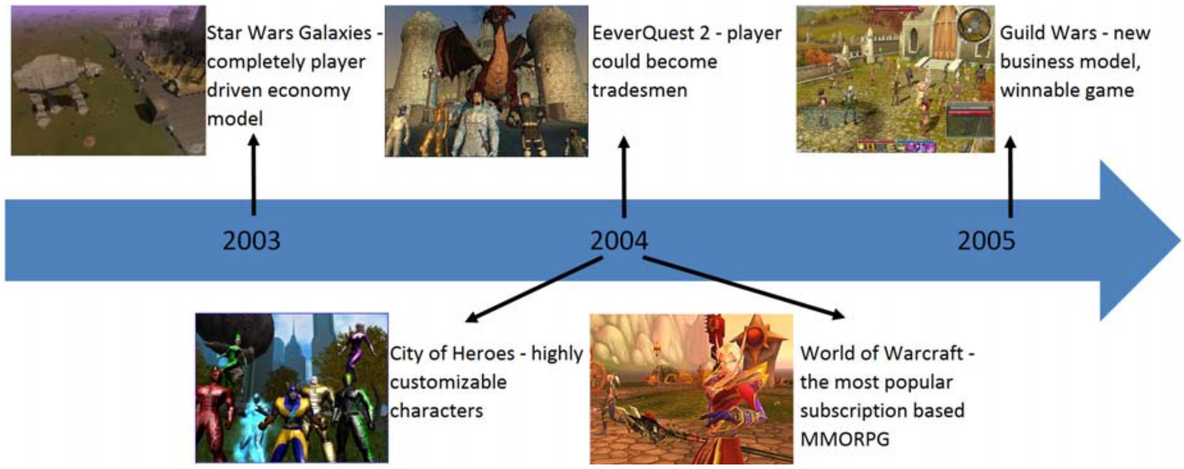


Figure 2.5: Time line of 2nd generation MMORPGs (part2)

in comparison with other MMORPGs, which made the game more accessible to casual players (i.e., players investing smaller amounts of time into the game).

Anarchy Online by *Funcom* was released in June 2001. This was the first MMORPG not set in a fantasy world (i.e., mythical worlds commonly set in the past), but rather in a science fiction (SF) setting (i.e., a future world). Anarchy Online also introduced dynamic quests, free trials, and in-game advertising. The concept of "instanced dungeons", more commonly referred to today as "instances" was the most significant innovation. *Instances* are special zones in the virtual world that generate a new copy, or instance, for each group that enters the zone. This allows players or groups in the instance to receive no help or attacks from other players while in their private copy of the zone. Instances aimed to prevent players from "camping" (i.e., patiently waiting in one area for a resource or a creature to spawn so they could acquire the resource or kill the creature before anyone else does (knowing that after being harvested the virtual resource will appear again after some time)). Through instances the designers allowed everyone to have an equal chance to experience the content. Later on, instances showed as a significant mechanism for achieving scalability, especially for densely populated areas of the virtual world.

In 2002, *Ragnarok Online*, by *Gravity Corp* was released. While not popular in Europe and USA, the game was highly successful in Asia with 25 million players, although this number is based upon a number of registered users (rather than active subscribers).

In 2002, a free to play (F2P) business model was modified and popularized with the release of *MapleStory* by *Nexon*. First F2P MMORPG was *Tibia* released in 1997, but Tibia was free with an option of upgrading it to a premium account. MapleStory was completely free to play. Instead of charging a monthly fee, it generated revenue by selling in-game virtual goods, such as avatars, virtual pets, and other in-game items. MapleStory would go on to become a major player in the new market for F2P MMORPGs [23].

In September 2002, *Earth & Beyond* by *Westwood Studios* was released. It was the first 3D SF space-ship based MMORPG.

Final Fantasy XI by *Square-Enix* was the first MMORPG with multiplatform support which was enabled in November of 2002. The same set of servers was used for players on PCs and PlayStation 2 console. It was also the first MMORPG to be released on consoles earlier that year (May 2002). Support for additional platform, XBoX 360, was added in 2006.

In March 2003, *Ubisoft* launched *Shadowbane* which introduced a concept of no quests, instead relying on player warfare to provide immersion. To support this goal it featured player-built, player-owned, and player-razed cities and capitals, and a system for player government.

A single universe for a very large number of players was realized in May 2003, with *Eve Online* (EvE) by *Crowd Control Productions* (CCP). While other MMORPGs solve the scalability issues regarding a high number of players through replication of the virtual world, and splitting the player base across those replicas, EvE created a single universe for all players. EvE is a space-ship based SF MMORPG with focus on space exploration, trading, and space combat. The current record number of concurrent players in EvE is over 60,000 [27].

Star Wars Galaxies (SWG) was released in June 2003 by *Sony Online Entertainment*. It was set in the Star Wars universe, which originally had many fans, resulting in a large number of players. SWG introduced the most extensive set of emotes, moods, and associated animations, which enabled players to express themselves and was, later on, implemented in many following MMORPGs. A big innovation of SWG was the virtual world economy model in which almost every item in-game was created by players, which resulted in completely player-driven economy.

An option for creation of highly customizable characters was introduced by *The City of Heroes* by *Cryptic Studios* in April 2004. This MMORPG introduced a new concept of creating a superhero with highly diverse powers and abilities. A player is able to customize everything, from facial features to the superhero's outfit. Such high level of customization allowed players to be visually distinguishable from one another, which many players asked for. Initially, the players could only create heroes battling the computer-controlled villains, but later on the game expansion opened the option of creating a villain as well.

EverQuest II (EQ2) by Sony Online Entertainment was released in November 2004. One of the most important new features was that players could become tradesmen, spending all their time crafting items to sell to the game community, rather than adventuring and levelling up a character class.

Although improved in all aspects in regard to its prequel, EQ2 never achieved high subscription numbers as it was released at the same time as *World of Warcraft* (WoW) by *Blizzard Entertainment*. While not introducing radically new concepts in comparison with the ideas of the first generation of MMROPGs, WoW was created with a lower system requirement for the graphical engine, an intuitive interface, easy learning curve, and reduced levelling time. All these characteristics, coupled with a big fan base of Blizzard's Warcraft universe (i.e., previous games from the RTS genre *Warcraft I, II, and III* were a big market successes), resulted in high market penetration of WoW. Even 7 years after the release of the game, WoW is still the leading MMORPG in the subscription based market with 11 million subscribers [28]. Currently only *RuneScape*, a F2P MMORPG released in 2001 by by Andrew and Paul Gower, can compete with WoW player numbers with approximately 10 million active accounts per month, and over 156 million registered accounts [29].

A new business model was introduced in 2005, by *ArenaNet* (a subsidiary of NCSoft) in the game *Guild Wars* (GW). GW has only a one-time purchasing fee. (Optional) additions to the virtual world which contain new content are available for purchase. GW was also designed to be "winnable", as developers would not profit from customers' prolonged playtime. Other differences compared to traditional MMORPGs include PvP-only areas, a relatively short playtime requirement to access end-game content, instant world travel, and strategic PvP. Based on these differences the game was termed a "Competitive Online Role-Playing Game" (CORPG) by its developers.

Many new MMORPGs are under development or have recently been deployed, as the revenues by the successful ones are attracting more and more developers. Most of the newly added or currently developed MMORPGs focus on increasing the quality of graphics while not introducing significantly new concepts. Notable titles include AION, *Age of Conan*, *Rift*, *Star Wars: The Old Republic*, and *Guild Wars 2*.

2.4 Game application architecture

In current commercial MMORPGs, the dominant architecture is client-server (C-S). While peer to peer (P2P) architecture has some inherent advantages in terms of scalability and cost compared to C-S the drawbacks weigh over the gains for MMORPG applications. Main drawbacks of the P2P architecture are related to the following issues: cheating mitigation, virtual world state distribution, virtual world state consistency, NPC host allocation, and game event dissemination [30]. Also, client-server model provides better administrative control when compared to other architectures [25]. Regarding the client-server model, it should be noted that the MMORPG infrastructure is typically multi-tier, comprising a proxy server farm, a number of game servers, and also the account management system (Figure 2.6) [31]. Proxy servers are in charge of communication with the clients, and act as dispatchers of messages between game servers and clients (e.g., forwarding the client request to the appropriate game server). Game servers are in charge of handling the game logic, modifying the status of game world, and sending back the updated status to clients based on Area of Interest (AOI) management. AOI will be further explained in the following subsection regarding network traffic. As for the

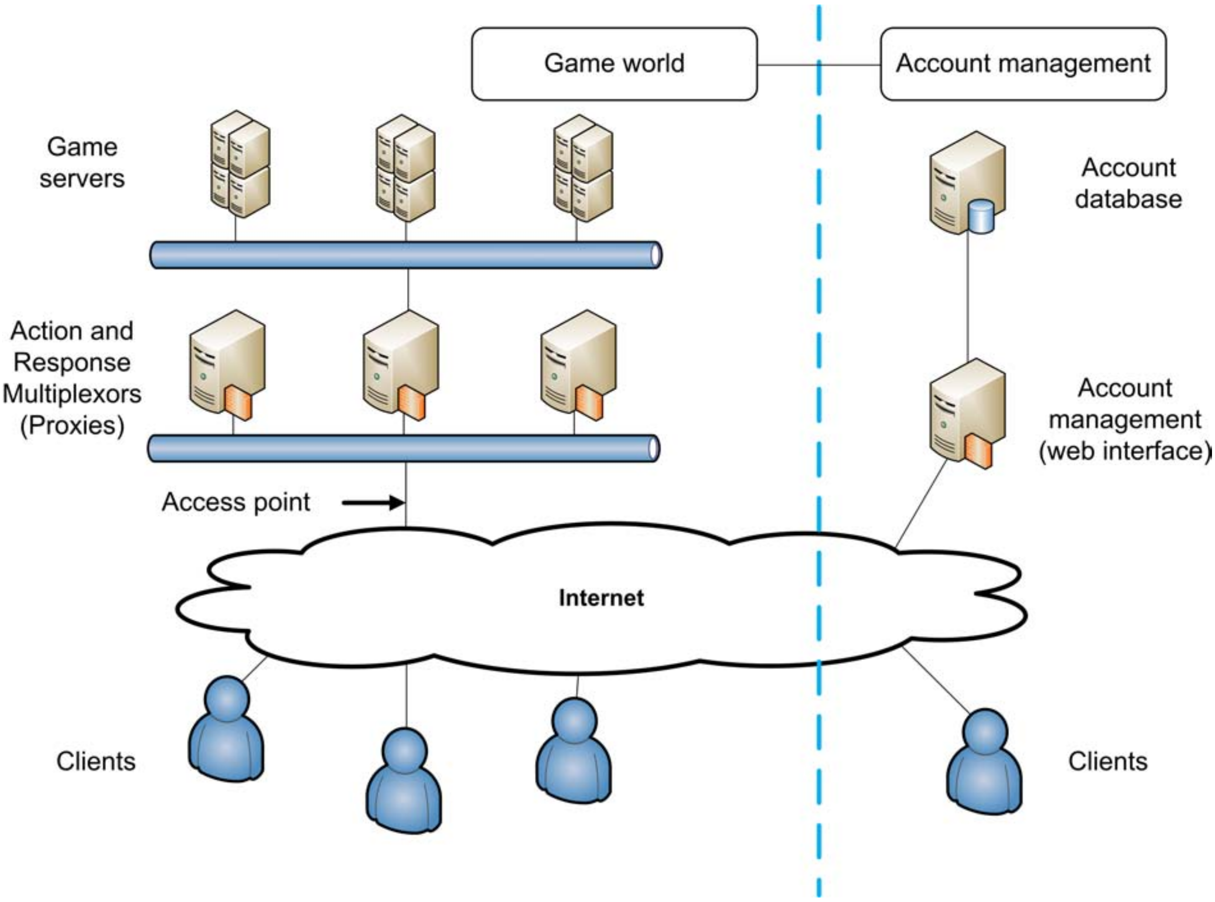


Figure 2.6: Architecture of a MMORPG system

accounting system, use of web interfaces completely separated from the game logic servers is typical for MMORPGs. Also, additional servers can be used, such as chat servers (i.e., server responsible for text-based or voice communication), login servers (i.e., server for authorization of user accounts, patch servers (i.e., servers responsible for distribution of the new game version), and database servers and (i.e., servers for storing information regarding the persistent world) [25]. One such group of all listed types of servers logically act as one "server" in the generic client-server model, and are often referred to as *server group* or *server farm*.

The client application in MMORPGs has several functions, such as rendering the virtual 3D scene, receiving updates from the server, and sending client's updates towards the server. The virtual worlds in MMORPGs are mostly static, meaning that users can not add their own 3D objects like in social virtual worlds (e.g., *Second Life*). This enables client application to hold all the information regarding the 3D models in the virtual world. This has two significant consequences: 1) the updates sent from the server do not carry information about the 3D virtual objects, which significantly reduces the traffic demands, and 2) the client application has significant requirements on the amount of storage space, which tend to grow as the game is expanding. For example, WoW increased the requirements on storage space from 4GB at release in 2004, to 25GB at the release of the 3rd expansion pack in 2010 [32].

2.5 State of the virtual world

Persistent virtual worlds of MMORPGs comprise various objects or entities which can be classified as follows [33]:

- avatars or virtual characters, which represent the players in the virtual world;
- non-player characters (NPCs), which are computer controlled mobile entities that have the ability to act independently;
- passive movable objects, which are passive entities which can be manipulated but do not initiate interactions; and
- immutable entities.

In order to simulate a real time virtual world, the following steps are needed (assuming client-server architecture): processing events coming from the connected clients (e.g., casting a a spell, shooting, collection of items, chatting); calculation of the new state of the active entities; processing state updates received from other servers; and broadcasting state updates to the connected clients [33].

In MMORPGs, the state of the virtual world for a particular player is determined by the Area of Interest Management (AOIM). As virtual worlds can be large in terms of size, number of users, and computer controlled entities, calculating and transmitting all actions of all entities would impose significant load on both servers and the network. As all entities do not need all updates, but only those updates which actually impact them (i.e., those who they can actually see in the virtual world) AOIM is employed. In Figure 2.7 we can see an example of two users, User 1 and User 2, and their AOIs. Users 1 and 2 are marked with squares, and all other

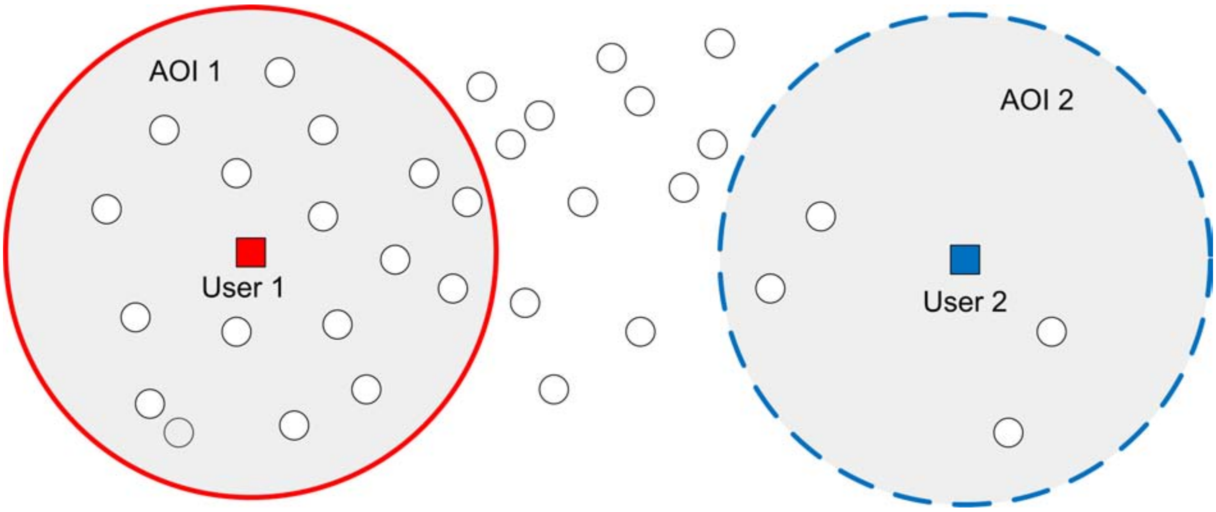


Figure 2.7: Example of two players' AOIs

users or entities in the virtual world with circles. Borders of both users' AOIs are indicated with lines. Only the updates related to entities located within a given user's AOI are sent to that user. Obviously, the User 1 has many more entities in his/her AOI than User 2, so he/she needs to receive and process a lot more updates than the User 2. This means that simulating a real-time virtual world for User 1 imposes a significantly higher load on both network and server computational capabilities than for User 2. Not only the number of entities in the AOI affects the load, but also the size of the game world, the total number of entities, and the level of interaction. Currently, the influence of the virtual world state is acknowledged, but what is missing is the specification of the virtual world state based on all the parameters which have an influence on it, and also the classification of different states.

2.6 Scalability techniques

In order to handle a high number of users, multiple scalability techniques are employed. A virtual world can be distributed across game servers in two ways: one logical instance of the virtual world is created for all players and is spanning across all game servers of the server farm (e.g., EvE), and replicating the virtual world on more than one logical instance, called *shard*. Shards partition the player base across several logical instances of the virtual world (those players can not interact between each other), and through that shards reduce the computational load.

Typically, one shard corresponds to one or more game servers, which enables dynamic scaling in order to adapt to the player population changes [34].

Scalability on a single shard is achieved through parallelization technique called “zoning” [33]. This technique partitions the virtual world into geographical areas called *zones*, which can be handled independently by separate machines, as depicted in Figure 2.8. In the past, zones had borders (e.g., invisible walls) with transition spots between them, such as portals, because transition between zones required certain time. These events had a bad influence on the players' immersion in the virtual world, which led to the development of the technique called “seamless zoning”, in which zones might be divided by some geographic markers (e.g., mountains), but there is no loading screens when players cross from one zone to another.

Another technique, called *mirroring*, targets parallelization of game sessions with a large density of players located and interacting within each other's AOI [35]. Such “hot spots” can include major cities in the virtual world, or zones in which gathering of players is common. To address this problem, mirroring is performed by distributing the load by replicating the same game zone on several servers. In each replicated server, the state for a subset of entities (i.e., active entities) is calculated, while the remaining entities (i.e., shadow entities) states are calculated in the other participating servers, and are synchronized across servers, as shown in Figure 2.8. The overhead of synchronizing shadow entities is much lower than the overhead of computing all entities as active entities [35].

Instancing is a technique which can be perceived as a simplification of mirroring, or even sharding on a smaller scale. This technique distributes the session load onto multiple parallel instances of the highly populated zones. The instances are independent of each other, which means that two avatars from different instances will not be able to interact with each other, even if they are located at coordinates within their AOI. This technique is common in many MMORPGs. All scalability techniques, including instancing, are depicted in Figure 2.8.

2.7 Summary and outlook

In this chapter we described various aspects of the MMORPG genre. We explained the terminology used, and introduced the basic concepts of an MMORPG, such as health of a virtual

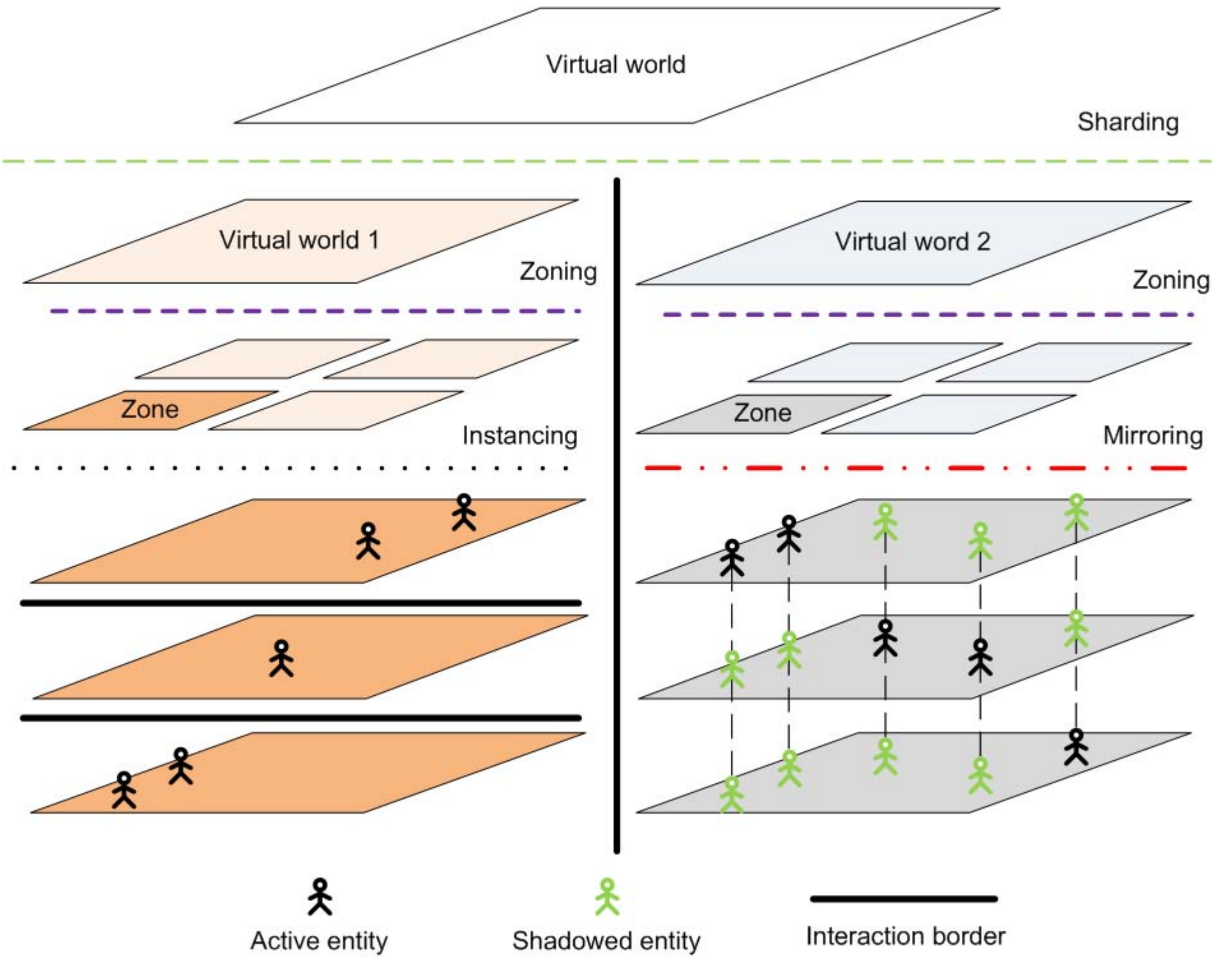


Figure 2.8: Scalability techniques for reducing computational load caused by high number of players

character and combat. Also, the development of MMORPGs through years has been described in order to show the evolution of the genre. Finally, the technical aspects, such as architecture, state of the virtual world, and scalability mechanisms are described. Through introducing the different aspects of MMORPG, this chapter prepares the reader for the following chapter in which the related work will be described, and also the chapter after that, which further explores the state of the virtual world, on both the application and the network level.

Chapter 3

State of the art review

Extensive research has been done in several past years on various aspects of MMORPGs, as their popularity has been in constant rise in the past decade. Due to a multidisciplinary approach used to describe and explain the characteristics of MMORPG network traffic, this chapter will provide a state of the art review in several aspects of MMORPGs. Focus is on one side, on the players, their psychological motivation, and behaviour in the virtual world, and on the other side, on measurements, analysis, and modelling of the traffic of networked games. Also, a survey of traffic generator platforms is provided.

3.1 Why? Player motivation in MMORPGs

The increasing interest of industry for MMORPGs is a result of the high revenues these games provide. But the question remains, *why* are the players so interested in these virtual worlds. Why do millions of players choose and pay to spend large amounts of their free time (average 22.71 hours a week [36]) to perform activities in the virtual world. Many studies have been done in order to find psychological features that arouse people to play games in general, and some studies have been especially focusing on the phenomenon of MMORPGs. Usually in virtual worlds of MMORPGs players are given a large variety of possible actions (e.g., gathering virtual items, battling hostile players, or organizing a group of players in order to defeat an opponent which can not be handled alone). Various MMORPGs have different characteristics such as compelling graphics, many options in creation of the player's avatar, or support for high number of interacting players. In order to determine why are MMORPGs so compelling, a popularity analysis of their many features is needed. Achterboch, Pierce and Simmonns [24] try to find an answer to these issues by determining which aspects make a successful MMORPG, and trying to assert which new features are users expecting in the future. Online survey is used in which 122 players revealed their perceptions on past MMORPGs and their expectations for the future ones. Authors inquired about players preferences and it is shown that MMORPG players most often additionally played single player RPGs and strategy games. The features of the MMORPGs mostly appreciated by the players are *Lots of Class/Skill Options, Graphics and Effects, Large World to Explore, Player vs Player*, and *Socialization*, surprisingly *Raid Content* ranked second lowest, which is surprising as most of the new content added in games is focused on raid content. As for the negative motivators *Exploits, Cheats, Item Duping* and *Running Out of Content* are the most serious issues for the players. Players' wishes for new features are mostly revolved around dynamic changing content and environment. Players want abilities to change the game world, and to feel that their actions are having an impact. Player created and controlled content is another suggestion. These results indicate that the creative side of the players is one of the resources of motivation in which the MMORPGs have not tapped yet. Adding voice communication into the game is also suggested, and most of the new MMORPGs, as well as other network based games, tend to provide built in voice communication support.

When trying to answer why people play MMORPGs, the history of this genre's development needs to be taken into account. Multi User Dungeons (MUDs) are multi user virtual worlds described entirely in text and are considered to be the predecessors of MMORPGs. MMORPGs draw many conceptual origins from MUDs, proved by the fact that first MMORPGs like *Everquest* were, at first, referred to as graphical MUDs. First study on player motivation in MUDs was done by Bartle [37] who divides the players into four major categories through a player interest graph with two axes: players – world and acting – interacting. Each of the categories has a distinct set of motivational parameters driving the player to play the game. Categories are killers (players, acting), achievers (world, acting), socializers (players, interacting), and explorers (world, interacting).

In the following years Bartle has expanded his model of player types by introducing another axis into his graph: implicit – explicit [38]. This extension lead to a total of eight new player types: Griefers, Opportunists, Politicians, Planners, Networkers, Scientists, Friends and Hackers. Explicit behaviour is defined as doing something with a forethought and implicit behaviour is defined as doing something without previously thinking about it. This is best described on the example of Griefers and Politicians which have been derived on the previous Killer type. Griefers act on players in an implicit fashion while politicians act on them in an explicit fashion (i.e. politicians act on players without killing them). Additionally, player development through types is defined, as it is noted that, during their lifetime in the virtual world, many players develop in the same way (e.g., many players start with killing other people, then they try to explore the virtual world, after that they try to "conquer" the world and in the end they "settle down" and mostly socialize).

Bartle's original model was criticized by Yee who argued that each MMORPG player shows a number of different motivations for play, which are related to different types, and that players can not be categorized through types [39]. Based on a 40 question survey taken on 3000 players of different MMORPGs, Yee has determined 10 distinct motivational subcomponents through the principal component analysis (PCA). Performing additional PCA, he grouped resulting motivational factors to three major components: *Achievement*, *Social*, and *Immersion*. Players can be motivated by multiple motivational components. The following are the descriptions of those psychological motivational components.

Achievement consists of three factors:

- *Advancement:* The desire to gain power, progress rapidly, and accumulate in-game symbols of wealth or status. Players who score high on this subcomponent derive satisfaction from reaching goals, levelling quickly and accumulating in-game resources such as gold. They enjoy making constant progress and gaining power in the forms offered by the game - combat prowess, social recognition, or financial/industrial superiority. Gamers who score high on this subcomponent are typically drawn to serious, hard-core guilds that can facilitate their advancement.

- *Mechanics:* Having an interest in analysing the underlying rules and system, in order to optimize character performance. Players who score high on Mechanics derive satisfaction from analysing and understanding the underlying numerical mechanics of the system. For example, they may be interested in calculating the precise damage difference between dual-wielding one-handed swords vs. using a two-handed sword, or figuring out the resolution order of dodges, misses, and evasions. Their goal in understanding the underlying system is typically to facilitate templating or optimizing a character that excels in a particular domain.

- *Competition:* The desire to challenge and compete with others. Players who score high on this subcomponent enjoy the rush and experience of competing with other gamers on the battlefield or economy. This includes both fair, constrained challenges - such as duelling or structured Player vs. Player (PvP) or Realm vs. Realm combat, as well as unprovoked acts - such as scamming or griefing. Players who score high on this subcomponent enjoy the power of beating or dominating other players.

Social consists of three factors:

- *Socializing:* Having an interest in helping and chatting with other players. Players who score high on this subcomponent enjoy meeting and getting to know other gamers. They like to chit-chat and gossip with other players as well as helping out others in general - whether these be less-experienced players or existing friends. Gamers who score high on this subcomponent are typically drawn to casual, friendly guilds.

- *Relationship:* The desire to form long-term meaningful relationships with others. Players who score high on this subcomponent are looking to form sustained, meaningful relationships with others. They do not mind having personal and meaningful conversations with others that touch on real life issues or problems. They typically seek out close online friends when they need support and give support when others are dealing with real life crises or problems.

- *Teamwork:* Deriving satisfaction from being part of a group effort. Players who score high on Teamwork enjoy working and collaborating with others. They would rather group than solo, and derive more satisfaction from group achievements than from individual achievements. Gamers who score low on this subcomponent prefer to be solo and find it extremely important to be self-sufficient and not have to rely on other gamers. They typically group only when it is absolutely necessary.

Immersion consist of four factors:

- *Discovery:* Finding and knowing things that most other players don't know about. Players who score high on Discovery enjoy exploring the world and discovering locations, quests or artifacts that others may not know about. They enjoy travelling just to see different parts of the world as well as investigating physical locations (such as dungeons and caves). They enjoy collecting information, artefacts or trinkets that few others have.

- *Role-Playing:* Creating a persona with a background story and interacting with other players to create an improvised story. Players who score high on Role-Playing enjoy being immersed in a story through the eyes of a character that they designed. These players typically take time to read or understand the back-story of the world as well as taking time to create a history and story for their characters. Also, they enjoy role-playing their characters as a way of integrating their character into the larger ongoing story of the world.

- *Customization:* Having an interest in customizing the appearance of their character. To these players it is very important that their character has a unique style or appearance. They like it when games offer a breadth of customization options and take time to make sure that their character has a coherent colour scheme and style.

- *Escapism:* Using the online environment to avoid thinking about real life problems. Gamers who score high on Escapism use the environment as a place to relax or relieve their stress from the real world. These players may use the game as a way to avoid thinking about their real life problems or in general as a way to escape real life.

Richard Van Meurs tries to unite these two different approaches to modelling the player motivation in MMORPGs [40]. He further explores Bartle's player types using some of Yee's results, and tries to answer the question: *Are there different online playing styles of MUD players in general and can these playing styles be related to off-line personality traits?* As a main goal of his thesis, categorization for different types of playing styles and to relate online behaviour to off-line personality traits is stated. For categorization of the off-line behaviour of players he used Big Five factors, and used the following terminology for them: Extroversion, Agreeableness, Conscientiousness, Emotional Stability, and the last factor identified with several terms: Intellect, Openness, or Imagination. Van Meurs performed measurements on 1773 players and found the following results:

- Five clusters of items – five anticipated playing styles account for more than fifty percent of the variance among the items;
- Type of MUD relates heavily to the type of player - roleplayers play role-playing MUDs, griefers play player-killing MUDs, and socialisers play social and educational MUDs; and
- Introducing the Big Five personality factors in the regression analyses lead to significant better explained variances.

The result of his findings is a new model which is based on the hierarchical approach, and type of the game.

As their popularity grew MMORPGs have spread on the mobile phone platforms as well. Mobile platforms limit many aspects of MMORPGs (e.g., large and detailed virtual world, user interface, high combat pace, etc.), so the same motivation models can not be applied. A player taxonomy model for mobile phone MMORPGs is presented by Fang et. al. [41]. Two motivation parameters are defined: *Relationship* and *Achievement*. They form following player types based on the model: *Socializer, Moderator*, and *Fighter*. Verification of the model is done on *ZhanGuo*, a popular mobile MMORPG in China. Also, analysis of the payments across all player types is done, and it is shown that although Socializers make only 15% of the player base, the average payment is 10 times higher than Fighters'. This means that although the achievement is important for the survival of the MMORPG, relationship is an important factor that makes players pay for the game. While a mobile platform is limiting for some features, it opens a range of different possibilities for players (e.g., players can access the virtual world more frequently, connection between real loaction and virtual world location, etc.), and offers a much higher level of presence of the player in the virtual world. Koivisto and Wenniger [42] analyse the possibility of enhancing MMORPGs with mobile features on a player sample of 20 players through usage of focus groups. Participating players show interest in advanced communication access, such as using voice chat on mobile platform. Event notifications of the happenings in the virtual world and passive participation are not so welcomed amongst the players, and privacy concerns are shown regarding the use of parallel reality and event notifications. Younger players were more open to new ideas, while older were more cautious.

Many MMOPRGs' business models are based on trading of virtual items (i.e., as micro transactions business model). According to [43] players spend 1.5 billion dollars on virtual items every year. Even the most popular subscription based MMORPGs are introducing some aspects of micro transactions (e.g., Blizzard Entertainment earned in one day between 1,4 and 3,5 million dollars by selling a new virtual item [44]). Investigation of users' perception of trading digital contents is presented by Choi et. al. [45]. Authors construct and verify a model which identifies the relationship among attributes of the transaction, users' perceived fun and transaction cost, and the intention to trade and use digital contents. Authors use the classification of motivation into extrinsic (i.e., focus on the performance of an activity because of its perceived to be instrumental in achieving valued outcomes that are distinct from the activity itself) and intrinsic (internal motivation to do something because it brings pleasure). Authors conducted an experiment on 86 players of *Mabinogi* by *Nexon Corporation* in which they verify their transaction models.

3.2 Who? Demographics of MMORPG players

There are currently several business models for MMORPGs namely, a) subscription based – paying a monthly fee to play the game (e.g., World of Warcraft), b) free to play using micro transactions – purchasing virtual items or abilities for the real money (e.g., *Runes of Magic*), and c) retail – purchasing the game client and additional zones and content in the virtual world (e.g., *Guild Wars*). This variety of models makes it hard to estimate the current number of players involved in these virtual worlds. According to the estimates based on subscription based games, strong growth in the number of users of MMORPGs started in the end of 1990s and by 2011 the estimated number of subscribers rose to over 20 million [46]. One thing is certain, MMORPGs are getting more and more popular amongst players, but also among publishers due to their constant revenues.

The common stereotype is that most of these players are mostly male teenagers, but according to the report from Entertainment Software Association [3], the average age of a player is 37 years, 53% of the players are 18-49 years old, and women age 18 or older represent a significantly greater portion of the game-playing population (37%) than boys age 17 or younger (13%). Some more specific research in the area of MMORPGs shows simmilar results. Yee [36] provides statistics from a series of online surveys on 30,000 MMORPG players (The Daedalus Project [47]), and a framework of user motivations which he will further improve in his other works. Results of his survey show that common stereotypes are wrong, the average MMORPG player age is 26.57, median age is 25, with a range from 11 to 68. The results show that the average female player is significantly older than the male player. Also, it is shown that players spend at average 22.71 hours a week playing MMORPGs with lower and upper quantlie 11 and 30 respectively. In [48] Yee further explores the issues of gender in MMORPGs. His results show female players are much more likely to play with a romantic partner regardless of the age, and also much more likely to play co-located (playing with someone else in the same room). The author also explores the differences between motivations amongst different genders and while it is not the case that women play only to socialize, and men play only to kill monsters, there are slight differences in their motivations. Male players have scored higher on Advancement, Mechanics, and Competition while female players scored higher in Relationship and Customization motivations. Williams, Yee, and Caplan [49] combine a survey with behavioral data retrieved from 7000 players with the help of Sony Entertainment game operator of *EverQuest2* (EQ2). Authors surveyed the players about their age, gender, race, household income, education and religion and compared the results with the data derived from the 2000 U.S. census. Results show that the average age of the player is 31.16 years, and again counter to the stereotype most of the players are in their thirties (36.69%). While male players are dominant (80%) female players play on averege more than male players (29.31 hours versus 25.03). White and Native American players play at higher rates, while Asians, Blacks and Hispanic play at lower rates. EQ2 players are wealthier than average, and are less spiritual. It can be concluded that the average MMORPG player is an employed person, and not a teenager, which further has implications on the overall behaviour patterns and consequently on network characteristics and load.

3.3 How? Player behaviour in virtual worlds

Understanding player behaviour is essential for several key aspects of the MMORPGs such as planing and developing better architectures for virtual worlds, predicting game life-span and players disinterest in the game, scalability solutions in order to avoid server saturation leading to heavy degradation of QoS and even unavailability of the certain areas of the virtual world. In this section we present the current research efforts regarding session characteristics. Created logs about session activity are described in Table 3.1. The following sections contain definitions of the most important parameters regarding player behaviour measurements and modelling for MMORPGs.

3.3.1 Session duration

In MMORPGs, aa session can be defined in two ways. The first definition of a session is the time between log in and log out of a specific character. Defined as such, session length can be measured based on characters availability in the virtual world. Several works [15, 56, 57, 58] perform measurements of this type of session length. Those measurements are based on creating a script which runs in the MMORPG client, commonly called an add-on, which enables polling of all active characters in the virtual world. In this manner, data about a large number of characters can be gathered with relative ease, as the only thing needed for this measurements is one computer running the game and the specified script.

The second definition of a session is the time a player spends playing, the time between starting the game and exiting the game. One player can have multiple characters so these two types of session lengths are not the same. Commonly a player has one main character and several alternative characters commonly referred to as “alts”. As the player session comprises several character sessions, measurement results describing player sessions differ from those describing

Table 3.1: Summary of player behaviour datasets and session characteristics

Game	Measur. duration	Measur. point	Dataset size	Session Characteristics	Additional parameters	Source
ShenZhou Online	20 hours	Server	1356 million packets (55TB)	Duration of game and map sessions, IATs of game and map sessions		[50, 51, 11, 52]
EvE Online	3 years	Server	Session log	Player load patterns, session duration (character), player churn	Impact of updates on player growth and amount of time spent playing	[53]
WoW	5 weeks	Client	Session log	Player load (hourly, daily patterns), arrival rate, departure rate, session duration (character), player movement	Identifying four facets for a complete behavioural model	[54]
WoW, Warhammer Online	4 months (WoW), 2 weeks (WaR)	Client	115,000 players and 75000 session with tracked movement	Player load over time, arrival rate, departure rate, session duration (character), player movement	Behavioural model: session length: Weibull, player distribution (per zone): Weibull, movement: linear equations, time in zones: Weibull	[55]
WoW	5 months	Client	Activity log	Session duration (character), session hourly patterns, player load (hourly, daily patterns), availability, downtime length	Player load observed in the 3 zone types: questing, transit, and city	[56]
WoW	664 days	Client	Activity log (1672820 sessions)	Player load (hourly, daily, subscription time), ON and OFF periods analysis	Prediction of long term behaviour	[57]
WoW	273 days	Client	Activity log	Player load (hourly, daily, weekly, monthly) patterns	Sever zone based consolidation strategy proposed	[58]
WoW	1107 days	Client	Activity log (667032 sessions)	Player load (hourly, daily, weekly, monthly) patterns, session length, daily session count, daily play time		[15]
WoW	20 hours + 1 week	Client and Network	1000 players	Session duration (player)	Session length:Weibull	[59]
WoW	1.5 month	Client	6 players	Session duration (player), session composition, action specific segment duration	Definition of user action categories	[8]
WoW	1.5 months	Client	104 players	Session duration (player), session composition (hourly, daily, and overall patterns), action specific segment duration	Measuring of written and voice communication in game	[60]
EverQuest II	8 months	Server	Complete experience data	Player churn, session time, session length	Algorithm for player churn prediction	[61]
Rockymud	12 months	Server	556 avatars	Session length, session IAT, transition probability between rooms, residence time in rooms	Modelling of user behaviour	[62]
Second Life	1 day	Client	165000 avatars	Population over time, arrivals and departures, mobility of avatars	-	[63]
Second Life	100 hours	Client	-	Analysis performed across moving speeds and differently populated places	-	[64]

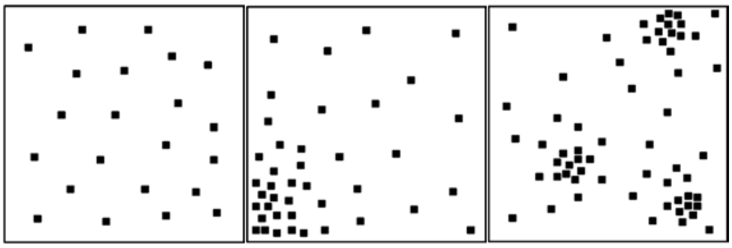
character sessions. This can cause confusion for the readers so it should be noted which type of session is measured. Player sessions are measured in two ways, either by measuring network traffic and extracting the data about the player sessions from the network traffic traces [59, 65], or by measuring session length by deploying an add-on on players' computers [60].

3.3.2 Player load

The number of active players in a virtual world (i.e., player load) of a MMORPG can vary significantly based on the time of the day or the day of the week. Server's computational load and the network load in terms of number of packets per second and used bandwidth, is very dependent on the number of active players. Depending on the architecture of the virtual world, player load may vary. In the sharded systems (i.e., the virtual world is replicated on multiple shards and the players on specific shard can not interact with the players on other shards (e.g., WoW)), the number of active players is approximately several thousand. On a single shard systems (e.g., EvE), all players are located in one virtual world and the number of players is measured in tens of thousands. Commonly researchers report player load patterns in time such as hourly (i.e., dependence of the hour in the day), daily (day of the week), weekly (i.e., week in the month), and monthly (i.e., month in the year).

3.3.3 Spatial distribution of players in the virtual world

Virtual worlds of MMORPGs can be very large in terms of virtual space (e.g., estimated size of virtual world of WoW is around 100 square kilometres without virtual space added in expansion packs [17], while Lord of the Rings Online is estimated to have 50 square kilometres of virtual spaces [18]). Thus, it is very important to understand the spatial distribution of the users in order to assign an appropriate amount of processing power for specific parts of the virtual world. Examples of avatar spatial distribution across the virtual world are shown in Figure 3.1. It is proven that players are not uniformly distributed in the virtual world [54], but that there are hot spots in which high number of players in concentrated while other parts of the virtual world can be almost “deserted”.



(a) Uniform (b) 1 hot spot (c) Multiple hot spots

Figure 3.1: Avatar spatial distribution types

3.3.4 Player movement

Movement of the players in the virtual world is often depicted through the number of visited zones in one session, and the time a character has spent in a specific zone [55]. Zones are geographical partitions of the virtual world which are processed by separate servers or server resources. Some works focus on smaller scale and model the player movement patterns in a specific zone [66], [67]. Player movement information is important for evaluating geometric routing schemes, performance of some P2P architectures is heavily dependent on player distribution, dynamics, and density of the avatars [68, 69].

3.3.5 State of the art

There are many approaches in measuring and describing player behaviour. From general session statistics, arrival and churn rates, to more specific works focusing on the low level player behaviour such as moving rate and speed, or action taken. First attempts to model user behaviour in networked games focused mostly on modelling the number of players in a game, player arrival rates, inter-arrival times, and session lengths. These first works have been done on the First Person Shooter (FPS) game genre so we briefly summarize a few of the most important works regarding that genre.

In an effort to see the impact of game traffic on different networks Sinha, Mitchell and Medhi [70] discuss player behaviour and game traffic. Authors identify effects of the migration to broadband services for the *last-mile* access. The trace used in the study is collected in four different markets for durations lasting from four days to one week, the game observed was *Counter Strike* (CS) a First Person Shooter (FPS) realized as a fan made modification of *Half-*

Life by Valve. They identify that only 1% or less players in their markets plays that game, and that average bandwidth consumption is from 3-7kbps with maximum value of 50kbps. They examine user arrivales and session duration and notice daily and weekly patterns in the number of active players. Upstream bandwith is identified as a main issue for the service provider. While not trying to analyticaly model the user behavior they suggest the use of the ARIMA (Autoregressive Integrated Moving Average) model [71]. The recommended model is realized by Henderson and Bhatti [72]. They examine the behaviour of players *Quake* and *Half-Life* and find that there are daily and network externalities dependencies of the number of players. They model the session membership with ARIMA. Session duration is modelled with Exponential distribution, and inter-arrival times of players also with Exponential distribution but with a much worse fit. Snowball phenomenon is identified meaning that new users joining will be followed by other users joining.

With the rise of the popularity of MMORPGs many scientists started to examine player behaviour in these virtual worlds. Due to their scale and architecture, MMORPGs are more complex in terms of scalability issues than FPS games (FPS are mostly realized through many small servers with player numbers up to 50 while in MMORPGs thousands of players are located on one server), as well in terms of behaviour due to wide variety of possible actions these worlds offer.

Session characteristics of the game *ShenZhou Online* have been a subject of analysis in a number of papers [50, 51, 11]. In [50] authors, apart from the very detailed traffic analysis, examine two types of sessions: (1) map session which is defined as the time period in which the character stays on a particular map; and (2) game session which is defined as the period in which a player remains in the game. Game session duration is heavy tailed and session arrival process can be modelled by a *homogeneous Poisson process* within 1-hour intervals. Time of the day has significant impact on session inter-arrival times. On the other hand, map session arrivals can not be modelled as such, which is, according to authors, an effect of the team play - players tend to cross the boundaries of the maps together. As on the topic of player behaviour, authors state that clustering nature of player actions (i.e., issuing commands from the players tends to be successive and in bursts) have major impact on the ACF function of the packet IATs. Furthermore, authors state that the diversity of user behaviour makes it difficult to define a "typical player", which makes modelling user behaviour and source traffic of MMORPGs especially challenging. In [51] Chen et al. examine the relationship between network QoS and session times using the survival analysis. They determine that both the network delay and the network loss significantly affect players' willingness to continue the game (or leave it). Also, they determine that delay jitter is more important than absolute delay in terms of playing time. Quantitatively they determine the degrees of player intolerance to minimum RTT, RTT jitter, client loss rate, and server loss rate in the proportion of 1:2:11:6. Specifically, a player's decision to leave a game due to unfavourable network conditions is based on the following levels of intolerance: minimum RTT 5%, RTT jitter 10%, client packet loss 55%, server packet loss 30%. In [52] Chen, Huang, and Lei test whether network game players are as sensitive to the network quality as they claim to be (often complaining about high "ping times" or "lags"). They confirm the relationship between network QoS and the game playing time which could serve as a measure of satisfaction of the players.

Feng, Brandt and Saha [53] have performed a long term study of a popular MMORPG *EVE Online* (EvE) by CCP on a complete session log of EvE since it's launch. They aim to determine how predictable is the MMORPG workload and how predictable are individual players. The measurements taken over 3 years time show a significant raise of concurrent player population in EvE. Authors discover that workload has significant hourly and daily patterns, with regular hourly and daily peaks. Observation of players joining and quitting the game shows that those two processes are highly correlated meaning that most of the players try the game for a short while before quitting, as the game does not interest them enough to play for a longer period. Player churn increases strongly, as the game is maturing, meaning that new players joining are much more likely to stop playing. Also, by examining session times authors prove that there are ways to determine whether the player would soon quit the game by examining intersession times.

World of Warcraft (WoW) by Blizzard Entertainment, as the most popular MMORPG today with more than a 50% market share [73], was a test case of many studies. Pittman and GauthierDickey [54] study virtual populations and their behaviours in WoW over a time period of 5 weeks. Authors use the */who* command of the WoW interface which returns the list of players currently online with some additional information (e.g., zone in the virtual world the players are currently in, level of the players etc.). Polling was performed every 15 minutes. Four facets needed for a complete player behaviour model are identified: 1) population changes over time, 2) arrival rates and session duration of players, 3) spatial distribution of players in the virtual world, and 4) movements of players over time. They confirm the existence of the hourly and also daily patterns of the number of concurrent users. The difference in the amount of players in a hourly pattern is almost five-fold. As for the session lengths, they confirm that most of the sessions are very short lived, 50% of the sessions were less than 10 minutes long, while 90% of the sessions were less than 200 minutes long. The spatial distribution of players in the virtual world proves to be following a power-law with most players visiting 6 or fewer zones and 40% of the players stayed in only 1 zone. Artificial workloads using uniform distribution of the players differ significantly from the situation which is observed in their measurements. In their following work [55] they develop behaviour models for WoW and *Warhammer Online* (WaR). In this work it is proved that models for two MMORPGs with different play styles can be modelled using a unified set of functions. The modelling is based on traces gathered for 4 months (for WoW) and for 2 weeks (WaR). The dataset comprises 75000 sessions in which movement was tracked and 115000 individual characters. Their full behavioural model consists of modelling session lengths (Weibull distribution), player distribution through zones (Weibull distribution), number of zones visited related to session lengths (linear dependence), time spent in zones (Weibull distribution), and player movement (Log-normal distribution).

Zhuang et al. [56] perform an extensive study of WoW by monitoring the behaviour of over 1000 players over a course of 5 months. They use their add-on developed for the WoW client and poll the online user base and their location also using the */who* command. Polling was performed every 5 minutes. Regarding the player count, they determine an existence of an hourly pattern, but they do not find that there is a significantly higher number of players that enter the virtual world on weekends than on weekdays. Authors determine a median of 50 minutes for the session length, with the longest session lasting around 12 hours. Longer sessions begin in the evening and the night while much shorter sessions are beginning in the morning. Downtime shows a daily pattern, and median of 179.38 which is remarkable, because of the human daily behaviour patterns such as sleep and work. Measurements of availability show that it is around 2% which equals to 30 minutes per day. Players are highly independant with most of the players going "solo" or in pairs. Authors partition the virtual world zones in 3 types: *Questing, Traveling,* and *Cities*. As far as differences in specific zones, questing zone show the highest average staytime (i.e., amount of time player spends in one zone), followed by cities, and traveling zones. The density of players is highly variable, as main cities have the greatest dencity, while questing areas have far lower. Also, density changes over time (i.e., new added content attracts high number of players).

Tarng, Chen, and Huang [57] analyse WoW player's game hours, on a trace sample gathered for almost two years with the use of an add-on for WoW client which polled all active players on one server similarly to Zhuang et al. [56], although polling in this dataset was performed every 10 minutes. While authors talk about 34521 accounts observed, it should be noted that they are examining characters (i.e., one user account can have more than one character). Subscription time is modelled by using a survival function rather than estimating it through cumulative distribution function since players' subscription times are censored (spanning out of the time line of the measurements). They conclude that the game is very addictive as 60% of the users will be observed 1 year after their first visit, while 50% of the users will subscribe for longer than 500 days. ON periods are defined as days in which a player logs into the game at least once and OFF periods as the days in which a player does not login. Authors note that for around 80% of players ON and OFF periods are shorter than 5 days. Strong pattern in the number of active players is noted for the hours of the day. Also, daily trend in player login count is noted, but it is not as strong as the hourly trend. Additionally, authors try to predict player long term behaviour based on their short term behaviour. Short term behaviour is described based on average session time, average daily session count, and average daily playtime.

Lee and Chen [58] use the same strategy for gathering the trace on which they study the hourly, daily, and weekly patterns of player numbers based on measurements from "alliance" faction on WoW server in Taiwan - "Light's Hope". They propose a zone-based server consolidation strategy which uses the number of players and spatial locality to lower the hardware requirements and energy usage. In order to evaluate the consolidation strategy they performed simulations which indicate that using a dynamic hourly based policy can significantly (i.e., around 50%) reduce the number of required servers as well as the energy consumed for both single-game and multiple games scenarios.

Lee et al. present a large dataset [15] comprising measurements taken in a 1107-day period between Jan. 2006 and Jan. 2009. using an add-on running */who* command every 10 minutes on the same server in Taiwan, but on other in game faction - “horde” as opposed to the previous measurements [58]. The dataset includes the avatars game play times and a number of attributes, such as their race, profession, current level, and in-game locations. Strongest patterns are evident for hours in a day, while daily patterns in avatar numbers are also strong (more players on weekends). Weekly and monthly differences are not that strong.

Ducheneaut et al. [74] focus on the social aspect of a MMORPG and measure player behaviour in WoW through several very specific parameters such as: time spent per level, group ratio per class, percentage of time spent in group with other players, guild membership and loyalty. Authors also use the command */who* in order to obtain data about currently online players. They found out that the time required to reach a certain game level rises slowly but very regularly, which indicates that WoW is a balanced game in which the difficulty increases gradually with the possibility of progress always within reach. Players tend to choose the classes (i.e., types of characters, with specific abilities and roles) that are most “soloable”, or in other words, able to perform well on their own. Time spent in group per class indicates that players play alone if they can. Also, players who spend less time in groups gain levels faster than the players who are frequently grouped. Results show that the players grouping time increases as they reach the level limit, as they come to the part of the game where most of the content is too hard to complete as one person, meaning that the endgame is social and not the game as a whole. Regarding guilds (i.e. player associations in game), it is shown that players within a guild spent more time playing than those without the guild which confirms the hypothesis about social pressure. Analysis of the guilds structure showed that they are in general small with average of 35 people, and that most guilds posses a small core which tends to play together, while the rest of the guild is not so closely connected. This phenomenon has been recognized by the game publisher, which has systematically over the course of the game reduced the requirements for group actions (i.e., the most challenging and rewarding group based activities in the game, so called “raids” have at first been designed for 40 people, first expansion reduced the requirement to 25 people, and the newest informations from the game publisher suggest that the best rewards in the game will be obtainable with a group of 10 people). Authors continue their work on studying the social behaviour in MMORPGs with the emphasis on guilds [22]. They examine their structural properties such as size, demographics based on level, strength of the relationship between guild members, and guild evolution through time. Results show that the guild churn is very high, guild life time is short, guilds with balanced class composition and more active players tend to survive longer. Authors also develop a social dashboard tool in order to visualize and explore guild survival metrics.

Nardi and Harris [75] focus on the collaborative actions in WoW and describe effects of social organization of the game and player culture on the players' enjoyment and learning of the game. Authors isolated the following mechanisms which encourage players to collaborate: parties for groups of 1-5 people, raids for bigger groups of up to 40 people, battlegrounds groups in which players are joined in order to fight other players, guilds as long-term groups which can vary much in size, duels in which two players involve in one-to-one combat, and trades as involvement of two players for mutual benefit. They also describe a broad array of not so formally defined collaborations which occur frequently in the virtual world. Authors also describe social interactions as a great tool in the process of learning the game.

Kawale, Pal and Srivastava [61] analyse the dataset of the MMORPG EverQuest II (EQ2) provided to them by the game publisher *Sony Online Corporation*. The dataset consists of complete experience data of all the players for the month of Jan to Aug 2006. Dataset contains data about when users completed a task, quest, points received, so information on which players played together can be extracted. Also, the dataset contains the list of all EQ2 players that unsubscribed in months August, September, and October. Authors aim to predict the churned players based on player engagement estimation which they evaluate using session length and session time. They evaluate 3 models for predicting player churn: simple diffusion model, classification based on network and player engagement, and modified diffusion model. Result suggest that modified diffusion model for the social influence on the player, which takes into account player engagement, based on activity patterns performs the best.

Kwok and Yeung [62] model the user behaviour of *Rockymud*, based on the measurements taken during 12 months. Rockymud is a game in which avatars can change location between rooms in the virtual world from which the whole virtual world is built (the game comprises of 10000 rooms. User behaviour is described through several parameters: probabilities of moving between specific rooms (modelled as a Markov chain), residence time in rooms (modelled by Pearson distribution with various parameters for different rooms), IAT between arrival of new users (modelled with exponential distribution), and session length (modelled by Pareto distribution).

Liang, Tay, and Neo [63] study the behaviour of user in a social virtual world *Second Life* (SL) by *Linden Labs*. Their trace captures the network traffic on a client side running a bot for SL which visited the designated areas of the virtual world. Information about the other avatars in the region can be obtained through parsing the network traffic trace (i.e., from served side updates). They note that separate islands in the virtual world have different number of avatars. They examine session behaviour (population over time, arrivals and departures, stay time, and returning to same island), mobility of users (number of visits, average speed in cell, and average pause time), and contact patterns (meeting duration, meeting stability, and average meeting size). They suggest new mobility models, and improvements for load balancing and zone partitioning.

3.4 Traffic analysis and modelling for network games

In order to perform traffic analysis and modelling, the following four basic steps are needed: traffic capture, data filtering, data analysis, and traffic modelling [76].

3.4.1 Traffic capture

Traffic capturing is a process of recording the network traffic traversing a specific link. The traffic capturing can be performed with hardware or software tools (e.g., Wireshark, tcpdump). Logs created during the traffic capture are referred to as “traces”. In the case of MMORPGs, traffic can be recorded on either client or server side or in the network. As opposed to First Person Shooter (FPS) games, MMORPG servers are only hosted by the game providers which makes capturing traffic at the server side by a third party impossible without the consent and cooperation of the game provider. Most of the measurements are done on the client side or in the network, but several authors have, in cooperation with the game providers, done measurements on the server side [50, 53]. During the traffic capture, especially on the server side, it is essential to make sure that the capturing process does not change the characteristics of the traffic. During the capture it is very important to take note of the game version, as the MMORPGs constantly evolve and change.

3.4.2 Data filtering

Through a data filtering process the relevant traffic is extracted from the overall traffic by identifying the flows between the clients and the server and removing the rest of the packets which are not of interest. While traffic captured on the server side may consist mostly of game related data, traces captured on the client side are usually mixed with many non game related flows.

3.4.3 Traffic analysis

Based on the obtained and filtered trace, the characteristics of game traffic are determined. There are several traffic characteristics which are commonly of interest to the researchers:

- *Packet size* (PS)- Commonly depicted through plotting of the Cumulative Distribution Function (CDF) which describes the probability that a real-valued random variable X with a given probability distribution will be found at a value less than or equal to x. As the MMORPG packet sizes are relatively small, in comparison with the TCP/IP header size, often only payload size (i.e., packet size without 40 bytes of the TCP/IP header) is described. Overhead caused by the signalling and its relation to the actual game data transferred is also observed. The standard unit value is byte (B).

- *Packet rate* - As game traffic rate is usually very variable, burstiness is commonly described in order to get the full picture about the packet load. Common unit for depicting packet load is the number of packets per second (pps).

- *Bandwidth usage* - Often described with respect to peak values. Unit used for describing bandwidth usage is mostly kilobit per second (kbps) for the client traffic, and megabit per second (Mbps) or gigabit per second (Gbps) for server traffic.

- *Packet inter-arrival time (packet IAT)* - Defined as the difference of the arrival times of two subsequent packets (i.e., the i-th packet and the $(i-1)-$th packet). The term inter-arrival is sometimes replaced with inter-departure times, but in general, this terminology is used regardless of the direction of the packets. Similarly to the packet sizes, this characteristic is often depicted through CDF graphs. Standard unit value for describing IAT is millisecond (ms).
- *Temporal dependence* - Usually described by Auto-Correlation Function (ACF).
- *Self-similarity* - Analysis for self similarity is usually done for the aggregate flows. The level of self-similarity is determined by calculating the Hurst parameter [77].

For the game traffic, the most heavily analysed parameters are PS and IAT.

3.4.4 Traffic modelling

Traffic modelling for games is performed by defining analytic traffic models (i.e., simple mathematical descriptions). These models are easier both to convey and to analyse in regard to empirical models of traffic (e.g. tcplib [78]). Modelling of the network traffic of MMORPGs has followed the approach for application traffic modelling by Paxson [79], firstly used in the area of network games by Borella [80]. The algorithm is also described in detail in [81], and it is presented as follows:

1. The probability distribution of the dataset is examined and an appropriate analytical distribution is chosen. This is usually done through the visual examination of the Probability Density Function (PDF) or Cumulative Distribution Function (CDF) of the data. A very valuable tool in this process is a Quantile-Quantile plot (Q-Q plot), which is a graphical method for comparing two distributions by plotting their quantiles against each other. First, the sets of intervals for the quantiles are chosen. A point (x,y) on the plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate). In this way, by plotting an empirical distribution, $F(x)$, against the chosen distribution $G(x)$ we can observe the goodness of

fit. If the resulting points are in a straight line, it means that the $F(x) = G(x)$, but in practice there are often deviations in the fit. By using Q-Q plot, it is easy to observe where deviations occur (e.g., lower tail, the main body, upper tail).

In order to manage large datasets, the values are aggregated into “bins”. The final results may become skewed depending on the bin choice. The algorithm for choosing the optimal bin size w is taken from [82]:

$$w = 3.49\sigma n^{(-1/3)}, \tag{3.1}$$

2. The dataset is fitted to an analytical distribution by using the method of least squares to determine the parameters of the distribution.

3. If the fit is especially deviating from the part of the distribution (e.g., upper tail), it is possible to model the data with split distribution. Q-Q plot can be used to observe the deviations.

4. Calculating the λ^2 discrepancy measure. As the standard goodness of fit tests are biased for large and messy datasets, a discrepancy measure, defined in [83], is used. We will briefly explain the discrepancy measure. If we have observed n instances of a random variable Y which we want to model using another distribution Z. N is the number of bins in which we partition the distribution Z. Each bin has a probability p_i associated with it, which is the proportion of the distribution Z falling into the i-th bin. Let Y_i be the number of observations of Y that actually fell into the i-th. bin. Chi-Square goodness of test X^2 is defined as:

$$X^2 = \sum_{i=0}^{N} \frac{(Y_i - np_i)^2}{np_i} \tag{3.2}$$

   K is defined as:

$$K = \sum_{i=0}^{N} \frac{(Y_i - np_i)}{np_i} \tag{3.3}$$

   Estimator for discrepancy for the grouped data is defined as:

$$\hat{\lambda}^2 = \frac{X^2 - K - df}{n-1} \tag{3.4}$$

where n is the number of observations in the dataset and df is the number of degrees of freedom of the test. The value of df is calculated as the number of bins N minus the number of parameters that were used to estimate the analytical distribution. In the case of deterministic distributions, in which all observations are expected to have the same value, this equation causes a divide-by-zero ambiguity if the empirical dataset contains values that vary from the expected value where the expected value is zero (i.e., np_i is zero). In order to avoid this problem the following alternative versions of the equations are used:

$$\hat{X}^2 = \sum_{i=0}^{N} \frac{(np_i - Y_i)^2}{Y_i} \tag{3.5}$$

$$\hat{K} = \sum_{i=0}^{N} \frac{(np_i - Y_i)}{Y_i} \tag{3.6}$$

5. Examination of the tail, in search for deviations using following expression:

$$\xi = log_2 \frac{a}{b} \tag{3.7}$$

   Where a is the number of instances predicted to lie in a given tail, and b is the number of instances that actually lay in this tail. If the b equals zero it is replaced by 0.5. If the values of ξ are positive it suggests that the model overestimates the tail, and negative values indicate that the tail is underestimated.

6. Calculation of the autocorrelation function of the trace. Usually, short-term autocorrelation, or the autocorrelation at lag 1 is examined.

3.4.5 State of the art

In this section we present the traffic analyses and models of games found in the literature. First the works regarding traffic analysis and models for non MMORPG game genres are briefly presented, followed by a more detailed section regarding MMORPGs. Also, a list of examined MMORPG applications is provided with the time and setup in which the measurements took place in Table 3.2 . In order to simplify the terminology in this paper, the traffic originated on the client side will be referred to as the client traffic, while the traffic originating on the server side will be referred to as the server traffic.

The first work in the area of traffic modelling for games was done by Borella [80] who studies the traffic of a popular FPSs *Quake I* and *Quake II*, and tries to model the empirical game traffic with analytical models using micro scale modelling. The author determines the models of packet IAT, PS, and the autocorrelations of those two models. The author takes into consideration the fact that traffic models are developed in order to design specific traffic generators, so it is needed to present the resulting models with limited range of parameters. Also, it is noticed that the autocorrelation is strong for the packet IATs, but not significant for PSs. Tests are performed on 3 different stations and results show that the client's traffic inter-arrival times are correlated with the CPU speed of the host. While slower hosts have significantly higher and more variable inter-arrivals, higher speed hosts transmit most packets at 13 or 14 ms intervals. Resulting model distributions were as follows: Extreme distribution and Deterministic for client IAT, Deterministic for client PS, Extreme for server packet IAT, and Extreme for server PS. Mobile gaming traffic in the 3GPP2 evaluation methodology [84] is modelled through with use of Uniform distribution for initial packet arrival, Deterministic for packet inter-arrival times Extreme distribution for the packet sizes.

Lang et. al. published a number of papers in which they describe and model the traffic of various games. Lang, Branch and Armitage [85] measure and analyse the traffic of *Quake III*, and provide a synthetic traffic model. In addition to PS and packet IAT, authors investigate the impact of different graphic cards, number of the players, and maps (i.e., virtual worlds on which fights in the game happen) on those parameters. They notice significant importance of the number of players on the packet length for server traffic, each additional client increases the average packet length by 12 or 14 bytes. Different maps had much lower influence. Influence of the graphic cards is shown on the packet rate, as stations with better graphic cards had significantly higher packets per second for client traffic. Resulting model has following distributions: Gamma distribution for server packet IATs (for the sake of simplicity replaced with a simple spike at 50 ms), Lognormal distribution for server PS with an Exponential distribution as an additive per player, Normal distribution for client PS, and client packet IATs as Deterministic with Exponential (though modelled only for 32 MB graphic cards). Authors have realized their models as a ns2 script. Lang and Armitage [86] analyse and model the traffic of the game *Halo* by Microsoft, for the XBOX console. They note that both client and server traffic have highly regular distributions of PS and packet IATs which can be well modelled with simple Deterministic models. Lang et. al. also model the traffic of *Half-Life* by *Valve* [87] in which they examine the OpenGL versus software based rendering and its impact on the network traffic.

Färber [88] analyses the trace of the game *Counter Strike* (CS) taken during a 36 hour LAN party involving 37 players. The main characteristic of the server traffic is its bursty nature, because the server generates a burst of packets each transmit cycle. Data rate is dependent of the number of clients as one more packet is generated for every active client. Author models the packet size and inter-arrival times treating server and client independently. The resulting model parameters are as follows: Extreme distribution for server PS and packet IATs (though author notes that Lognormal or shifted Weibull distributions lead to acceptable fits), Deterministic distribution for client PS and Extreme for server packet IAT. Author suggests modelling the number of active clients as well as session durations on a higher model level.

Dianotti, Pescape and Ventre [89] analyse and model the traffic of *Starcraft* by *Blizzard Entertainment*. They model the source traffic on the level of PS and packet IAT, and also study the autocorrelation at lag 1. As Starcraft uses the P2P model for multiplayer games, client and server traffic are similar and are modelled with Deterministic values. For packet IAT it is shown that they are dependent on the number of players. For server packet IAT times Exponential distribution is used and for client packet IAT both Deterministic and Uniform. Authors note that autocorrelation function of packet IATs decays very slowly which indicates Long Range Dependence. Investigating the Hurst parameter of the packet IAT, reveals non significant estimates of 0.5 for server packet IAT, but a 0.7 value for client packet IAT suggesting burstier nature of outbound traffic which is preserved over different time scales.

Cricenti and Branch [90] try to model the traffic of the *Quake 4* by *Raven Software*. In their approach they use mixed autoregressive/moving average (ARMA) models. For a Stationary time-series Xt, ARMA(p,q) process is defined as:

$$\phi(B)X_t = \Theta(B)Z_t$$

where $\phi(B)$ is the autoregressive polynomial of degree p and $\Theta(B)$ is the moving average polynomial of order q.

$\phi(B) = 1 - \phi_1 B - ... - \phi_p B^p$

$\Theta(B) = 1 + \Theta_1 B + ... + \Theta_q B^q$

In any ARMA process the random behaviour of the process is labelled by its “innovations”. This means that each term in the sequence, while having a strong dependence on the previous value or values, deviates from the expected value in some random fashion. Sequence of these random values makes the innovations of the specific sequence. By using this method authors model the packet sizes of the client traffic and accomplish good results with ARMA (1,1) model. For the innovations distribution Gaussian or less commonly Laplace distribution is used, but both of them fail to capture the long-tail nature of the innovation distribution.

Table 3.2: Summary of network traffic datasets and traffic models

Game	Measur. duration	Measur. point	Dataset size	Traffic Characteristics	Model	Additional parameters	Source
ShenZhou Online	20 hours	Server	1356 million packets (55TB)	PS, packet rate, bandwidth, packet IAT, IAT ACF, burstiness, self-similarity	N/A	N/A	[50, 51, 11, 91]
World of Legend	7x90 minutes	Client	N/A	PS, packet IAT	Client PS: deterministic, Client packet IATs: Extreme Value, Server PS 2 deterministic + Extreme Value, Server packet IATs: sum of 2 Extreme Value	N/A	[92]
WoW, Lineage II	253 min	Network	526 million packets (382 GB	PS, packet rate, bandwidth, source/destination addresses, port number, protocol number	N/A	Classification algorithm	[93]
Lineage	8 days	Server	3.5 trillion packets	PS, packet IAT, bandwidth load	PS: Lognormal	Correlation of bandwidth load and number of players	[94]
Lineage II	4 days	Server	12 trillion packets	PS and packet IAT of aggregated traffic, bandwidth, RTT, packet IAT within session	packet IAT: Extreme Value	Correlation of bandwidth load and number of players	[95]
Anarchy Online	1 hour	Server	N/A	packet rate, RTT	N/A	Testing different TCP variants	[96]
Ragnarok Online	206 hours	Client	3 million packets	packet IAT, burstiness	N/A	Bot identification techniques proposed	[97]
WoW, Silkroad Online	9 hours	Client	N/A	N/A	N/A	Classification of virtual world state	[98, 99]
WoW	20 hours + 1 week	Client and Network	N/A	Bandwidth, packet IAT	Client PS: deterministic, server PS: Weibull, packet IAT: joint distribution of 3 random variables	N/A	[59]
WoW	N/A	Client	1.4 million packets	PS, packet IAT, bandwidth	Modelling of application data unit sizes and IATs over 5 classes of user actions	Analysis and modelling performed across five classes of user actions	[9, 8, 100]
WoW	N/A	Client	N/A	Delay, jitter, RTT, bandwidth, packet loss, carrier to noise ratio, handover impact	N/A	WiMaX measurements in subway, bus, and stationary, 3 game scenarios (downtown, hunting, battlefield)	[10]

Continued on next page

Table 3.2 – continued from previous page

Game	Measur. duration	Measur. point	Dataset size	Traffic Characteristics	Model	Additional parameters	Source
WoW	N/A	Client	N/A	PS, packet IAT	Client PS: normal, server PS: Exponential, client IAT: normal, server IAT, normal	Analysis on 4 classes of player actions (no play, movement, hunting, battle)	[2]
WoW, Silkroad Online	3 days	Client, Network	200 WoW, 100 other MMORPG flows	Wavelet analysis	N/A	Classification of virtual world state on two axis (moving-staling, inside city - outside city)	[98, 99]
Second Life	100 hours	Client	N/A	Bandwidth, PS	N/A	Analysis for actions: standing, walking, flying in popular and unpopular areas	[64]
Second Life	∼3 hours	Client	17 million packets	Bandwidth, PS, chanell utilization, accuracy of avatar position	N/A	Analysis of parameters over separate data channels and classes of user actions: standing, walking, teleporting, and flying	[101]
Second Life	N/A	Client	N/A	PS, packet IAT, bandwidth	N/A	Parameter analysis over classes of actions and virtual world surroundings	[102]
Second Life	8 months	Client	N/A	PS, packet IAT, bandwidth	Traffic modelling of PS and packet IAT across user actions: standing, walking, flying in popular and unpopular areas	Analysis and modelling performed across six classes of virtual world situations	[103]

The research group lead by Kuan-Ta Chen has performed very extensive research on traffic patterns of the MMORPG *ShenZhou Online* by *UserJoy Technology Co. Ltd* [104]. They traced 55TB of network traffic which was obtained on the server side, with permission and the assistance of the game operator. They analyse the dataset in several papers [50, 51, 11, 91, 52, 105].

Chen, Huang, and Lei [50] make an in depth analysis of the gathered trace and search for the evidence of the self similarity in the MMORPG traffic. Their most important findings are:

- Bandwidth consumption of MMORPGs is very low compared to other action paced games such as First Person Shooters (FPS) (e.g., 7 Kbps for *ShenZhou Online* compared to 40 Kbps for *Counter Strike*).
- Packet IATs of client traffic conform to exponential distribution with a packet rate of 8 pps. Server IATs are more regular with approximately 50% of the CDF concentrated around 200 ms.
- Traffic in both directions exhibits short-term temporal dependence within connections. Traffic from the client to the server shows dependence due to the clustering nature of user

actions. The same effect in server to client traffic is caused by the spatial locality of the number of characters near the player's character, which shows up in terms of the temporal locality in traffic as the player's character moves continuously on the map.

- Temporal dependence exists in the aggregate client traffic. Furthermore, the aggregate traffic in both directions exhibits strong periodicity, which implies synchronized game processing for all clients. Such periodic packet bursts can cause network performance problems.

- Aggregated traffic exhibits self-similarity over a wide range of time scales (for at least 2 h), which could be explained by the heavy-tailed active/idle periods of individual players.

Also, authors note that TCP produces a large overhead consisting of 73% of overall client traffic and find the TCP delayed ACK to be the limiting factor for the RTT. The differences between traffic characteristics for active and idle players are found very high, especially in the packet IAT.

Using the same 55TB trace authors evaluate the TCP performance for MMORPGs [91]. They try to find out which transport protocol should be used for online games TCP, UDP or some other protocol. Based on the characteristics of the traffic such as: tiny packets, low packet rate, application–limited traffic generation, and bidirectional traffic they claim that TCP is not an appropriate protocol for MMORPGs. As TCP was originally designed for network-limited bulk data transfers, it does not adapt well for the requirements of a MMORPG. Fast retransmit mechanism is highly ineffective as only 0.8% of all dropped packets are detected by the fast retransmit mechanism. As the packet rate is slow in MMORPGs and the inter-sending packet time can be longer than the retransmission timeout (RTO), TCP assumes that connection is idle and unnecessarily resets the size of the congestion window. Tiny packets that MMORPGs send also cause a high signaling overhead (46% of the overall bandwidth used). In the end, the authors suggest following guidelines for designing an appropriate efficient protocol for online games:

- Support for both reliable and unreliable delivery

- Support for both in-order and out-of-order delivery

- Accumulative delivery
- Multiple streams
- Coordinated congestion control

Wu et al. [105] use the gathered network trace in order to evaluate the usage of TCP for MMORPGs and to propose guidelines for designing a protocol for this genre of games. The following findings suggest that TCP is not appropriate for MMORPGs: small game packets make TCP/IP headers occupy up to 46% of the total bandwidth; as some packets can be processed out-of-order, forcing a in-order delivery causes unnecessary transmission delays; the congestion control mechanism is inefficient because game traffic is application-limited, rather than network-limited; the fast-retransmit algorithm is ineffective because game traffic is bi-directional and packets are generated at a slow rate. They design the following transporting options: *multi-streaming* which enables separate streams for different types of game messages; *optional ordering* which allows certain messages to be processed as soon as they are received without being buffered if the preceding messages have not yet arrived; *optional reliability* - the messages which do not require reliable transmission and can be ignored. In order to compare their new approach with existing transport protocols (TCP, UDP, DCCP, and SCTP) they conduct network simulations with a real-life game trace from the game *Angel's Love*. Their comparison shows that the proposed strategies can effectively raise the level of game satisfaction among players through reducing end-to-end delay and jitter significantly. The main problem is that it is necessary to identify game packets according to the types of content.

World of Warcraft (WoW) [106] from *Activision Blizzard*, currently the most popular MMORPG with over 12 million subscribers [46], is a common use case for research involving MMORPGs for both traffic analysis and player behaviour.

Svoboda, Karner, and Rupp [59] analyse the traffic trace captured within the 3G mobile core network and note that the significant part of traffic is coming through port TCP: 3724 a well-known port for WoW. They establish that WoW is amongst top 10 TCP based services in the monitored network (i.e., a 3g mobile core network in Austria) and consumed 1% of all TCP based traffic. Also, they perform active measurements and capture a trace of a two groups consisting of five WoW client connected to the ADSL line. Analysis at packet level shows that the packet size in the downlink direction is quite smooth and can be modeled with a Weibull distribution, while the uplink packet sizes have large discrete steps. They note that the high number of transmitted packets are ACK packets carrying no application data. Inter-arrival time is modelled by a joint distribution of three variables. Authors state that the modeled traffic conforms well to the original traffic in the bandwidth CDF, but the original burstiness of the traffic is not captured.

Hilven and Woodward present a bit different analysis of the traffic of WoW [107], focusing on security aspects and analysing in detail the traffic regarding login, launcher, authentication and updating the WoW client. Their security analysis concludes that those aspects of MMORPGs are adequately secured.

Shin et al. [108] propose an novel method for modelling the network traffic of games. They analyse packet size and inter-arrival times of WoW and first person shooter game *Left 4 Dead* (L4D) [109] by *Turtle Rock Studios*. As the gaming traffic is highly erratically distributed, it is very difficult to analyse and properly model, even with composition of several distributions. Authors propose a transformational scheme in order to simplify the shape of the traffic so it can be mapped to an analytical model. First, they classify the data into major dataset and minor dataset in respect to a moving average value. The minor dataset is input for the next iteration of the process until the proportion of minor datasets is less than a chosen threshold. Major dataset is fully changed into a transformed PDF, and a transform table is generated for the reverse transform in the traffic generation procedure. In the third step the non-linear curve fitting is used to obtain an analytical model for each transformed PDF. Through this methodology authors model the traffic of both games better, and especially the highest erratic aspect of these two games - packet size of L4D which shows almost 10 times better Chi-square statistics for transformation modelling in comparison to composition modelling. Authors implement their model in a online game traffic generator [110] and show that the generated traffic is following the model in case of L4D traffic but not WoW. These distortions came from the Nagle algorithm and Delayed Acknowledgements of TCP. Disabling these options in the generator resulted in adequate goodness of fit levels.

Han and Park [93] perform a traffic analysis of popular game applications on the transport layer and propose a classification method called alternative decision tree (ADT). The traffic trace on which analysis is performed has been collected on the border router of a campus network with average utilization of 120 Mbps. Their characterization is done on a number of game applications, including two MMORPGs: WoW and *Lineage II* [111] by *NCsoft*. They examine various characteristics of the traffic: PS, packet load, bandwidth usage, correlation of packet size distributions between flows of the same application. The ADT algorithm is based on a simple decision tree with few discriminators. Critical discriminators are PS statistics (i.e., minimum, maximum, mode, mean, distribution), transport behaviours (i.e., mean bytes per second, mean pps), and flow information. For testing the algorithm, they captured 562 million packets with total volume of 282 GB on which they show that their algorithm has high precision and recall.

Kim, Choi, Chang, et al. [94] analyse the traces captured on the server side of the MMORPG *Lineage* by *NCsoft*. They have captured over 281 Gb of data over 8 days. Client packet sizes are very small and narrowly distributed while client packet IATs were between 0 and 20 seconds with an average value of 386 milliseconds. Server packet sizes have a wider distribution while packet IATs are similar to client side traffic with average value of 438 milliseconds and 99% of packets departed within 4 milliseconds. Traffic models are formed as follows: for packet size the Power Lognormal Distribution is used and for packet IAT Extreme Value Distribution is used, though authors state that the model is not a good match. Also, authors have looked into average connection time per flow (49.68 minutes). Bandwidth consumption during a week has an evident daily pattern which authors explain through linear correlation between number of users and bandwidth.

Kim, Choi, Chang, et al. [95] analyse the traces captured on the server side of the MMORPG *Lineage II* and show that the major characteristic of the MMORPG traffic is asymmetry of upstream and downstream traffic. In their measurements on the server side they have captured around 7.7 billion data packets. They note that client traffic has 22.9% data packets while server traffic has 97.6% data packets. Comparing this game with it's prequel *Lineage*, authors note a significant increase in the size server packets (arround 15 times larger). Authors confirm the linear correlation between number of active users and client traffic (correlation coefficient 0.99), as well as, server traffic (correlation coefficient 0.95). The deviation of the correlation of the server traffic and the number of players is explained by packet segmentations (packets larger than MTU) which incurs additional TCP/IP overhead. Avereage IAT of packets within session is arround 200 ms due to the TCP delayed ACK system. Authors suggest that delayed ACK system should be changed in order to reduce the Round Trip Time (RTT). Also, a strong correlation between number of users and aggregated upstream and downstream traffic is found. As for the distribution of session durations, significant heavy tailed characteristics are found (e.g., longest session duration measured was 80 hours). Session IATs show an average value of 401 milliseconds, while at peak times on average 2.5 new users arrive every second.

Wi, Huang, and Zhang model the traffic of one of the most popular MMORPGs in China - *World of Legend* [112] from *Shanda Corporation* [92]. Their modelling is based on a trace obtained on a client accessing the game through a mobile GPRS access network. The trace consists of 7 traces each lasting 90 minutes. For packet IATs they use an Extreme Value distribution for the client side traffic and a sum of two Extreme Value distributions for the server side traffic. Packet size is modelled as a sum of discrete steps (on 66 bytes and 72 bytes) and an Extreme Value distribution for the server side and as a deterministic distribution for the client side (73 bytes).

Griwodz and Halorsen [96] analyse a 1 hour long network trace of an MMORPG *Anarchy Online* [113] provided to them by the game provider *Funcom*. They show that while single TCP streams are thin, the server link can be carrying hundreds or thousands of concurrent streams which together may cause congestion without reducing the sending rate. Authors claim that using TCP does not have to be slower than using UDP, as the send buffer is usually empty and an event may be sent immediately. Repeated timeouts ruin the game experience because the number of in-flight packets is so small that fast retransmission due to multiple duplicate acknowledgements is very rare. They tested multiple TCP implementations and found little effect on the performance of a MMORPG, but were able to get better results with modifying Linux kernel with new calculation for round trip time timeout (RTO). The QoS gain from this method is significant when there is need for higher number of retransmissions (i.e., the lag experienced by the player due to repeated packet loss is reduced). Authors suggest introducing a proxy server as many clients share a network path in order to reduce the load on the server up to 40% and reduce maximal and overal latency to the clients (due to a lower loss and fewer retransmission, as introducing proxies always causes the increase of latency).

Chen, Jiang, Huang et al. [97] analyse the traffic of the game *Ragnarok Online* [114] by *Gravity*, in order to identify the game bots (i.e., auto playing game clients). Through their measurements on the client side they have gathered over 3 million packets of 206 hours. The traffic has been captured during the gameplay of four real players and two bot programs on different networks. In their analysis it is evident that compared to real players packet IATs of bots show significant difference due to timers. They calculate the entropy of each trace in order to describe the degree of randomness of packet IATs and confirm that players have higher entropy than bots. Also, they provide a suggestion for a threshold which divides the players from bots, but indicate that the entropy is dependent of the segment size. Authors also look at the client response time (i.e., difference between a client packet's departure time and the most recent server packets inter-arrival time) and burstiness of the traffic. Based on noted traffic characteristics authors propose four general decision strategies and two integrated schemes for bot detection.

3.5 What? Influence of virtual world state to network traffic

The parameters of the of the virtual world state such as number of active players, number of active Non-Player Characters (NPCs) in the area, or avatar mobility, have an influence on server load as well as on the characteristics of network traffic. Depending on the values of those parameters, different situations can occur in the virtual world. The first step is the identification of classes of situations or classes of player behaviour as possible situations are numerous. The next step is traffic characterization of the classes in order to validate their definition.

3.5.1 State of the art

Many of the works from the area of traffic modelling acknowledge the influence of different situations in the virtual world on the traffic patterns [85, 95, 50]. The following works explore this relationship further.

Chen and Lei [11] examine their collected trace in order to find out the implications of player interactions on generated network traffic. They found out that the distribution of the players in the virtual world is heavy-tailed, which implies that the static and fixed side partitioning mechanisms for the game world are not adequate. They prove that the neighbours and teammates tend to be closer in terms of network topology which is beneficial for the timely delivery of the state updates, and by that, for the fairness of the game. As for the session times, they discover that players with higher degree of social interaction tend to have longer sessions.

Wang, Kim, Vasilakos et al. [10] measure the traffic of WoW in a WiMAX network and focus on analysis of the performance in terms of application level packet dynamics such as RTT and jitter, and WiMAX link level statistics such as wireless link quality and handovers. They measure the traffic in several scenarios: subway, bus, and campus. Also, authors note the impact of the state of the virtual world and make measurements for 3 types of situations in the virtual world: *Downtown*, *Hunting*, and *Battlefield*. For each of the combinations of real and virtual world scenarios RTT is studied and depicted with CDFs. Bandwidth consumption shows significant differences for separate virtual world situations even up to 4 times in the downlink direction (8.93 kbps for hunting towards 32.11 kbps for battlefield). Packet loss is defined across real world scenarios and it is highest for the subway in uplink direction and for bus in downlink direction.

Park, Kim and Kim [2] collect and analyse network traffic traces of FPS *Quake 3* (Q3) and a MMORPG WoW. They define user actions based on the number of players and player behaviour. For Q3 actions are defined as: Shooting, Moving, Normal, No Play. For WoW, actions are defined as: *Hunting the NPCs*, *Battle with players*, *Moving*, and *No play*. Q3 measurements have been done on the server side, while WoW measurements have been taken on the client side, as servers are not publicly available. Analysis is performed and data and packet rate, packet size and inter-arrival times are observed. Battle is the most demanding in terms of data rate and hunting in terms of packet rate. The size of the packets is modelled by the Exponential distribution on the server side and normal on the client side, while IATs are modelled with the normal distribution.

Packet level analysis of MMORPG traffic is done by Szabó, Veres, and Molnár [98, 99]. Authors claim that the nature of human behaviour has a high impact on traffic characteristics and that it influences the traffic both at macroscopic level (e.g., traffic rate) and at microscopic - payload content - level. They measure and analyse the traffic of WoW and *Slik Road Online* by *Joymax*, and examine the server generated traffic. The states of the virtual world are defined through two axis, movement of the player and number of surrounding players (in or out of the cities i.e., densely populated areas). This results in four possible states: *Moving in the city, Moving outside the city, Stalling in the city,* and *Stalling outside the city*. Identification of the separate states is done by active measurements and wavelet analysis. Validation of the model is done by controlled measurements and comparison with the defined states. Secondly, they measure the traffic while capturing in-game video and applying a heuristic algorithm on the character's screen in order to determine in which of the states the character is currently in. The algorithm achieves an accuracy of 68%. In the third step of the validation authors create several measurements on other MMORPG games such as *Guild Wars*, EvE, and *Star Wards Galaxies*. Using Deep Packet Inspection (DPI) authors present a novel method for classification of the MMORPG traffic on the basis of dynamic signature. They presume that the payload segment encoding the location of players in gaming environments show such statistical properties that characterize human motion models.

Second Life (SL) by Linden Labs is currently the most popular social 3D virtual world. The main difference between SL and MMORPGs is the lack of game mechanics and rules, and a dynamical virtual world mostly generated by the users which leads to much higher traffic demands of SL. Due to these characteristics the virtual world of Second Life varies very much in terms of number of users and content which results in different traffic characteristics. Fernandes et al. [64] collected and analysed the network trace of SL for 100 hours on different locations in the virtual world. Authors note that SL requires much more bandwidth in comparison with other MMORPGs with 500Kbps for a full experience which include external audio stream and 200Kbps as an average. So much difference in the bandwidth requirements is a result of the fact that the world of MMORPGs is static and the 3D models are stored on the player's computer, while the world of SL is constantly changing as players add their own models, which other players need to download. Authors also notice a significant difference in traffic requirements of popular and unpopular regions of the world (in terms of the number of players). Movement pattern of the avatar has a strong correlation with the required throughput as *Standing, Walking,* and *Flying* result in a different throughput. Authors found a clear relationship between the context of the virtual world (in terms of number of users and players movement pattern) and generated network traffic. The same group of authors [103] expands their research with devising traffic models for SL. As they note the significant differences in traffic characteristics for specific actions of the user, and virtual world characteristics, they perform their modelling for combination of 3 categories of player movement - Standing, Walking, and Flying and 2 categories describing density of avatars in a zone - Popular and Unpopular. They model 4 parameters inbound and outbound PS, and packet inter-arrival and inter-departure times. Inter-departure times for different combinations of avatar density action are modelled with split distributions combining 2 or even 3 distributions among which are Beta, Gamma, Lognormal, Extreme, and Weibull distribution. Inter-departure times are modelled with a Beta distribution. Outbound PS is modelled as either deterministic or Extreme distribution. Inbound PS is modelled as a split distribution of 4 models combining: Uniform, Exponential, Deterministic and Extreme distributions.

Kinicki and Claypool [102] add an additional transport movement category - *Teleporting* and also take into account density of objects in the virtual world, as well as population (i.e., number of active avatars). They focus on bandwidth, packet size and packet IATs, stating that the impact of zone characteristics and avatar actions is different than the results received by Fernandes et al. [64].

Further and more detailed analysis of the relationships between application functionality, traffic control system and the wider network environment was done by Oliver, Miller, and Allison [101]. Two sets of studies are done: one of the traffic generated by a hands-on workshop which used SL (17 Gigabits of data); and a follow-up set of controlled experiments to clarify some of the findings from the first study. They determine the average throughput of SL on 231 Kbps, average RTT of 153 ms, and loss rate of 0.02%. They monitor all network parameters seperatly for four mobility states of the avatar: Standing, Walking, Flying, and Teleporting. The results show that the avatars' mobility has significant impact on the characteristics of the network traffic. Also, they note that SL has sophisticated traffic management mechanisms, as it performs application level framing, provides reliability for some packets, tracks RTTs and congestion levels, and reduces bandwidth at high loss levels. SL also has seven channels each of which correspond to data that fulfill a distinct functionality (e.g. Texture, Asset and Wind).

3.6 Traffic generators

In network planing or performance testing it is often useful to to put real data in the network. As obtaining the appropriate real data traces can be in some situations hard or even impossible, network traffic generators based on mathematical traffic models are used as an alternative. Traffic generation is creating a time-stamped series of packets arriving at and departing from a particular network interface with realistic values. This synthetically created traffic should accurately reflect arrival rates and variances across a range of time scales (e.g., capturing both average bandwidth and burstiness). In this section a list of existing generators and their capabilities is presented.

Harpoon [115] is a flow-level traffic generator. It can work in two modes. In analytic mode, based on the analysis of captured traffic, it uses a set of distributional parameters which are automatically extracted from Netflow traces to generate flows that exhibit the same statistical qualities present in measured traces, including temporal and spatial characteristics. In user defined mode the user defines the properties of the generated traffic through XML files. Haproon uses client - server model, and it is designed for Linux, Solaris 8 i FreeBSD platforms. Harpoon toolkit comprises additional four elements. Config_validator – a utility for validating the structure of a configuration file, and harpoon_flowproc – a utility for pre-processing flow records (raw Netflow version 5 or flow-tools format) for self-configuration are written in C++. Other two elements, harpoon_conf, a utility for generating configuration files for harpoon (the self-configurator) and harpoon_reconf, a utility for tuning existing configuration files to produce desired traffic volumes are developed in Python scripting language. More information about Harpoon can be found in [116].

Distributed Internet Traffic Generator (D-ITG) is a platform capable to produce traffic at packet level accurately replicating appropriate stochastic processes for both inter departure time and packet size random variables (Exponential, Uniform, Cauchy, Normal, Pareto). D-ITG supports both IPv4 and IPv6 traffic generation and it is capable of generating traffic at network, transport, and application layer. It is available on Linux, Windows and Linux familiar platforms. It enables evaluation of a set of QoS performance metrics such as throughput, packet loss, delay (one way delay and round trip time) and jitter. D-ITG consists out of several components : Internet Traffic Generator Sender, Internet Traffic Generator Receiver, Internet Traffic Generator Log Server, Internet Traffic Generator API, and Internet Traffic Generator Decoder. More about D-ITG can be found in [1].

MGEN [117] is a toolset for generating real-time traffic patterns of unicast and/or multicast UDP/IP applications. Script files are used to drive the generation of loading patterns over the course of time. The receive portion of this tool set can be scripted to dynamically join and leave IP multicast groups. MGEN log data can be used to calculate performance statistics on throughput, packet loss rates, communication delay, and more. Currently there are two versions of the system 3.x and 4x, 3.x. Version 4.x runs on various Unix-based (including MacOS X) and Win32 platforms.

Rude/Crude [118] stands for Real-time UDP Data Emitter and CRUDE for Collector for RUDE. It is a command line traffic generator and measurement tool for only UDP. These programs are a "side product" of the project Faster 2000 (former Faster Pro) which concentrates on different QoS mechanisms in IP-networks and were designed and coded because of the accuracy limitations of MGEN. Rude/crude runs on Linux, Solaris and FreeBSD.

UDPgen [119] is a simple unicast traffic generator application, with two components gen_send and gen_recv. The traffic generator application is designed to be able to fill a 100Mbit Ethernet, and account for all data sent. Thus, reserved bandwidth flows and/or best effort flows can be sent, competing for the resources of a 100Mbit Ethernet. Packets are marked with a sequence number. Per second bandwidth and packet loss data are collected.

Network Traffic Generator [120] supports Linux, FreeBSD and Microsoft Windows platforms and it is written in order to check scalability issues or in other words what massive amounts of traffic of certain type will do to an intervening network. It does not measure throughput or response times.

MXtraf [121] can be used in order to saturate a network with a tunable mix of traffic with a small number of hosts. It can generate three types of traffic: periodic TCP flow with specified length, long-lived infinite-source TCP flow, and non-responsive UDP traffic at a specified rate.

NTGen (Network Traffic Generator) [122] is realized through a Linux kernel module. The module sends these packets on the network interface. It is capable of bypassing the overhead of a user-application during generation process to achieve the best performance and maximum utilization of network's bandwidth and allows the user to configure multiple (parallel) packet generation streams with arbitrary parameters to achieve ultimate free-choice and random traffic generation including packet rate. It supports common network protocol packets (Ethernet, IP, TCP, UDP, ARP) and it uses a well-defined meta language (Bison and Lex) for configuring packet generation streams.

Netpref [123] is a benchmark that can be used to measure the performance of many different types of networking. It is written in C and supports TCP and UDP via BSD Sockets for both IPv4 and IPv6, DLPI, Unix Domain Sockets, SCTP. It provides tests for both unidirectional throughput, and end-to-end latency.

Traffig Generator (TG) toolkit [124] supports the following platforms: Linux, FreeBSD, SunOS, and Solaris. TG is a traffic generator program that creates one-way UDP or TCP streams between a source and a sink. Inter-arrival times and packet lengths are used to describe the traffic. Information regarding the source and sink, such as packet transmit and receive times, is recorded in a binary log file for later post processing. Additionally, toolkit consists of dcat which takes the binary log file and produces an ascii representation, and a perl script gengraph which transforms this data into a format suitable for viewing via public domain graphing tools such as xplot, xgraph, and gnuplot.

TFGen [125] is a Windows GUI 32bit application (made for WinNT/2000 but usable on Win9X), which supports TCP/IP. It can generate traffic with certain traffic patterns and multicast traffic. Speed is one octet per 10 ms

Generator of Self-Similar Network Traffic [126] is a program that generates self-similar traffic by aggregating multiple sources of Pareto-distributed ON- and OFF-periods. Every source generates packets of only one size. Pareto distribution of burst sizes is achieved by using Pareto distribution for the number of packets in a burst (minimum is 1). Inter-burst gaps are also Pareto-distributed.

GenSyn [127]is a synthetic traffic generator implemented in Java. The stochastic user behaviour is described by state diagrams. The model is scalable because it allows a composition of users in each state, instead of creating a new instance of the process for every user. The stochastic user behaviour model controls the creation of TCP connections and UDP streams through interface modules that links the GenSyn process to the underlying Internet protocol stack on the workstation. This means that on transitions between specific states in the stochastic model, an interface process will be initiated and IP packets are sent and received through the network. More information about the GenSyn can be found in [128].

Poisson traffic generator [129] is a Linux application for generation of traffic with Poisson distribution of packet arrivals.

NetSpec [130] is traffic generator/emulator that allows the user to define multiple traffic flows from/to multiple computers. Netspec runs on Linux, FreeBSD, Solaris and IRIX. It emulates TCP, UDP, WWW, FTP, MPEG, VBR and CBR Traffic.

Jugi's Traffic Generator (JTG) [131] only sends one stream and the characteristics of the stream are given only with command line arguments. Receiving jtg process can be set to loop and receive several streams but, only one at a time. The received packets can be logged and the log files can be analysed with jtg_calc. It includes a support binary for IPv6 networks.

TrafGen [132] supports UDP (unicast or multicast), TCP, RTP, HTTP, DNS on the same machine or between 2 different machines. In UDP mode, generation of traffic is done continuously or according to statistical models (Poisson or M/Pareto). For multimedia transmissions (RTP flows), javaMediaFramework 2.1 is necessary.TrafGen can be used in interactive mode (swing) or in command mode.

Kernel-based Traffic Engine (KUTE) [133] has been developed to have a maximum performance traffic generator and receiver mainly for use with Gigabit Ethernet. It consists from the sender and receiver and is designed to run on Linux 2.6 kernel or later.

Ostinato [134] is an open-source, packet/traffic generator and analyser for Windows and Linux with GUI. It supports most common standard protocols on several layers Ethernet, 802.3, LLC SNAP, VLAN (with QinQ), IP, ARP, and TCP, UDP, ICMP. Modification any field of any protocol is supported (some protocols allow changing packet fields with every packet at run time e.g. changing IP/MAC addresses). It supports multiple streams which can be configured in terms of number of packets, bursts,and packet rate. Exclusive control of specific ports is provided to prevent the OS from sending stray packets provides a controlled testing environment. Integration with Wireshark is included in order to capture and analyse packets.

3.7 Summary and outlook

As MMORPGs have relatively recently gained significant popularity (i.e., last ten years), research focusing on these games is still rather new. The purpose of this chapter was to provide summarized insight into current research activities regarding MMORPGs across fields of network traffic characterization and modelling, and player behaviour. Based on the covered research we present several established claims which should be taken into account when conducting research in the area of MMORPGs.

- Traffic characteristics vary significantly depending on the state in the virtual world. The most demanding state should be taken into account when performing calculations based on these characteristics (e.g., network link capacity planing).
- Client and server traffic are highly asymmetrical (i.e., server traffic is larger in volume).
- During measurement process it is necessary to note the version of the game, as MMORPGs constantly change and evolve (e.g., in the game version 2.0 of WoW , an ability to fly was added which adds another axis for player movement measurements).
- Several TCP mechanisms proved to be bad for the purpose of MMORPGs (e.g., Nagle algorithm, Delayed Acknowledgment), while other are highly ineffective (e.g., Fast Retransmit).
- Avatar distribution in the virtual world is not uniform, rather conforms to the multiple hot spots model, which needs to be taken into account when performing simulations (e.g., for new load balancing algorithms).
- Significant hourly and daily patterns exist in several aspects of player behaviour (e.g., player number, type of player actions).
- Session lengths should be defined as referring to either character sessions or player sessions in order to convey correct information.

We also list some of the currently open research questions in this area.

- Current transport protocols (TCP, UDP, DCCP, and SCTP) are not appropriate for MMORPGs due to characteristics of game traffic. The design of the new protocol for MMORPGs is still an open research issue.
- Detection of bots is an important issue as bots can have an influence on virtual world economy, and significantly reduce the level of satisfaction of regular MMORPG players.
- Player behavior models in terms of both mobility and player actions (i.e., what exactly players do in the virtual world, when, and for how long).
- Models for prediction of players leaving the game (in terms of completely stopping playing the game). MMORPG providers try to keep each and every player, and especially "old" players (in terms of time spent in the game), as it is proved that new players have significantly higher churn rates than old players.
- Influence of player behavior on server architectures and load balancing algorithms.
- Scalability issues and new techniques for load balancing problems in cases where a large number of avatars is populating a small fraction of the virtual world.
- Relationship between player behavior and perceived Quality of Experience (QoE), as same QoS levels will not equally satisfy a player performing a simple and repetitive task (e.g., picking virtual flowers), and a player trying to perform a highly complex task (e.g., trying to defeat a raid boss (i.e., highly complex NPCs)) which requires a possibly large group of coordinated and highly organized players.

In the following chapter we further explore the state of the virtual world, through devising metrics which describe such a state, and creating a categorization of possible states. We also, further inspect the identified categories on the network level.

Chapter 4

Categorisation of user actions in the virtual world

Each player participating in a MMORPG has his/her copy of the virtual world on the client side. That copy does not include the whole virtual world, but only a fraction of it defined by AOIM. We identify the *personal state of the virtual world* (PSOVW) as a set of parameters which characterize the personal copy of the virtual world for a particular player in terms of both player actions and characteristics of his/her surroundings in the virtual world (e.g., number of players in his AOI, number of NPCs, etc.). A "situation" in the personal virtual world for specific player is determined by the values of that set of parameters. This set of values is referred to as "virtual context". On the server side, the virtual context determines how much computing is necessary to calculate the next step in the simulation for the particular player. On the network side, the virtual context reflects on the characteristics of traffic sent and received by the particular player (game client). The impact of virtual context on traffic characteristics is defined as the "application aspect" by Matijasevic et al. [135] which relates to the question: "How does the interaction at the application (user) level affect the communication characteristics?".

As there are many possible situations, the idea behind this approach is to group similar situations and form categories of the situations on which further study can be performed. The first step is to identify the parameters which compose the PSOVW. The second step is to categorize the possible situations. The third step is to validate the proposed categories through measurements of network traffic. The computational load on the server is considered out of scope of this work. If it were of interest for future work, the validation would require cooperation with the MMORPG provider. The fourth step is to create the models of network traffic for each of the established action categories. The remainder of this chapter is structured around the four steps outlined above.

4.1 Metrics for characterizing the personal state of the virtual world

The following metrics (properties) are used for the characterization of the personal state of the virtual world:

- *Number of other active participating players* (other players in the vicinity of the observed player in the virtual world, which (may) in some way affect the player). This number can vary significantly, from tens or even hundreds of players (e.g., big battles amongst players, capital cities or other player hubs) to none (e.g., a player alone exploring a sparsely populated part of the virtual world);

- *Required cooperation amongst participating players* indicates how active are the participating players, how much they influence each other, and how complex their interactions are. While in some situations all players in the vicinity are participating in an action (e.g., defeating a boss), in others the players are only observers (e.g., only standing in capital cities chatting with other players);

- *Rate of player activity* (dynamics of player's input). In certain situations players need to input commands often, very quickly, and responsively (e.g., on boss fights if a healer is only one second late with the healing spell, the tank might die), while on the other hand some situations do not need prompt reactions nor frequent input (e.g., studying prices at the auction house (i.e., in-game system for off-line trading of virtual goods amongst players));

- *Mobility of the player in the virtual world* has significant influence on both network traffic and server computing load. Fast avatar movement results in frequent position updates which need to be sent, as well as more entities exiting and entering player's AOI, and more complex calculation to present consistent state to other players in the virtual world (e.g., if the player is moving too fast, and changing directions constantly, his/her avatar may appear as "teleporting" around the virtual world for other players, which breaks immersion and lowers the other players' perceived Quality of Experience (QoE));

- *Number of actively participating NPCs* (same argument applies as with the number of players);
- *Combat requirement* is important as MMORPGs are mainly focused on combat, either amongst players, or between players and NPCs. Combat is performed in various ways and styles and most of the player's abilities in game are related to combat. Most players try to progress their avatars in order to improve their performance in combat. Also, there are non-combat based actions which are much simpler, and with less different player abilities; and
- *Communication aspect* represents the need for additional means of communication, most often text or voice. Using of voice communication, namely Voice over IP (VoIP), in MMORPGs is very common today.

4.2 Categorisation of situations from the perspective of the user

As we aim to use the categorisation of the virtual world situations for the model of player behaviour, the categorisation can not be based on purely numerical methods, as we also need to take into account the players' motivations, mechanics, and the high diversity of MMORPGs. Also, the proposed categorisation needs to be meaningful in relation to game design. Player's motivation is needed because each situation in MMORPG is eventually a result of players wanting to achieve something (e.g., a player wanting to gain the maximum level will perform quests as they provide extra experience). When creating such categorisations, one must take into account the mechanics of the game (i.e., a construct of rules on which gameplay is based) in order to know how the player will achieve his/her goals, to which kind of situations will those goals lead (e.g., mechanics often focus on group combat and high cooperation amongst players for the best rewards). To make the categorisation of situations in the virtual world of MMORPG as general as possible, and applicable on most MMORPGs, we must focus on common elements.

In a typical MMORPG, the player controls a virtual character (i.e., avatar) which represents the player in the virtual world. While these virtual worlds can differ significantly in theme and settings, from fantasy worlds (e.g., WoW, Lineage), over virtual worlds placed in current time (e.g., *APB* by *Realtime Worlds*) to science fiction worlds (e.g., EvE Online), all of them have certain common characteristics. Several fundamental elements may be identified in most MMORPGs: some form of progression, social interaction within the game, in-game culture, system architecture, and character customization [12]. We decided to focus on player progression, as in most games this is the player's primary goal [12]. As mentioned earlier in this thesis, progression can be achieved through upgrading the avatar's level or skill points, equipment, and other means.

Thus, the players' intentions and goals are the main factor in shaping the situations in the virtual world, in other words, the ***categorisation of situations in the virtual world from the perspective of the user can be achieved through categorizing player actions which fully characterize the situation in the virtual world (including parameters of the virtual world which surrounds the player)***. For example, a player which has an intention of collecting a lot of virtual resources (e.g., virtual ore) will visit an area in the virtual world which is sparsely populated in order to gather more resources. Therefore the parameters of the virtual world which surrounds the player (e.g., number of other active players and NPCs, etc.) are a result of the player setting to perform actions of gathering virtual resources.

The algorithm of categorisation is as follows. Firstly, we perform the categorisation of situations in terms of player motivation, and game mechanics. The proposed categories are assigned with values for each of the parameters defining the personal state of the virtual world. Based on the assigned values, we further analyse and group the proposed categories.

As a case study MMORPG application we use WoW, which has a fantasy based virtual world with humans, elves, orcs, and other fantasy races. Like many other MMORPGs, WoW is based on a client/server architecture with multiple servers. The virtual world of WoW is replicated on many servers (server shards), with each server handling a complete copy of the WoW virtual world. Only the players using the same server can communicate and interact with each other. The following description of MMORPG player actions is illustrated by examples and screen shots from WoW.

The proposed novel user action categories are:

- Trading;
- Crafting;
- Gathering;
- Questing;
- Dungeons;
- Raiding; and
- PvP combat.

The proposed categories are considered general for the MMORPG genre. They are discussed in more detail next.

Trading of virtual goods is an important part of almost all MMORPGs. The main motivation for trading is progression in terms of economic power (e.g., in-game money or virtual items) of the character on the seller side, and progression in other ways on the buyer side. For example, in WoW a player can buy gems which can be inserted into weapons and armour and make them more powerful. As for the mechanics of trading, we can identify two types of trading: on-line and off-line. On-line trading is a direct exchange of virtual goods, either done between a player and a NPC, or between two players. Off-line trading is an exchange of virtual goods between two players through an assisting system, which in WoW is Auction House (AH). In AH players create auctions for virtual items, and those auctions last for a certain amount of time (up to two days) during which other players can buy auctioned items. In Figure 4.1 examples of trading situations are depicted with a character browsing an AH and collecting mail (i.e., all items purchased through the AH are received by in-game mail system). This action type

(a) Browsing auction house (b) Collecting mail

Figure 4.1: Examples of trading

involves very few active players (only one or at most two players are involved), though it should be noted that the number of inactive players in the AOI can be high, as trading is commonly performed in main hubs of player gathering (e.g., capital cities in WoW). The cooperation level between players is also very low, as the task of trading is fairly simple - the players just need to “meet” in the virtual world and exchange the items for money by using drag-and-drop in the trade window. Also, out of all players in the vicinity only two can be actively participating, and others are just observers. Mobility of the virtual characters is very low, most often players stand still while negotiating, or inspecting prices. Similarly to the number of players, there can be quite a large number of NPCs present in the AOI, but a trading player typically interacts only with the other party involved in the trade. This type of activity does not include combat, nor does it require additional means of communication. Estimated scores of the trading category on the defined parameters are displayed in Figure 4.2

Crafting and **gathering** virtual items are usually referred to as professions. Combined with trading, professions make the base of all virtual economies. Players are interconnected through professions as they need products which only other professions provide (e.g., in WoW in order to make a certain explosive, engineers need a liquid which only alchemists can make) [136]. Commonly in MMORPGs many professions exist and one virtual character can learn only some of them to enable the exchange of goods. The total number of available professions in WoW is thirteen, but the player’s character can only learn two “main professions” and four “secondary professions” at any time. The main motivation of the professions is again economic

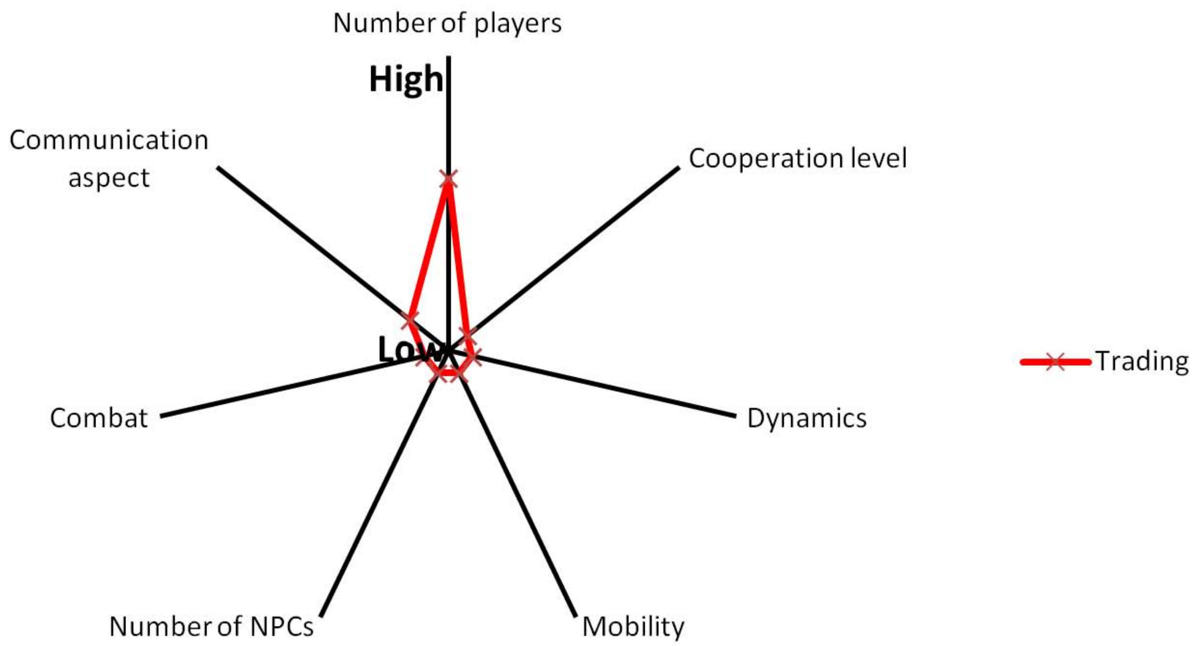


Figure 4.2: Scores of trading on PSOVW parameters

progression on one side, and different progression types on the other (e.g., in WoW players that learn “blacksmithing” can create armour and weapons). As for the mechanics of the professions, they consist of player alone interacting with certain virtual items in specific places in the virtual world. For example, in order to create the most powerful potions, players which learn “alchemy” in WoW must create those potions in alchemy laboratories. Examples of an avatar performing actions of crafting and gathering are displayed in Figure 4.3.

Crafting professions usually consist of creating new virtual items from the set of other virtual items (e.g., in WoW an alchemist creates potions from different herbs). On the other hand, gathering professions consists of extracting the virtual items from interactive objects scattered in the virtual world (e.g., in WoW a player with “mining” can extract minerals from ore nodes). Interactive objects disappear after the gathering procedure, but appear again after a certain time period. Due to the different mechanics, states of the personal virtual world for two types of professions have some differences. Common aspects are that, as this is a single player activity, there are no other players involved, no cooperation or additional means of communication are needed. Also, both types of professions have a low rate of player input (e.g., creating a virtual item is done only by activating a certain ability once all required ingredients are gathered).

(a) Crafting – mixing potions

(b) Gathering – picking flowers

Figure 4.3: Examples of crafting and gathering

Players usually engage in crafting professions in towns and cities with access to a "bank" or an AH, in order to obtain needed materials, store them, or sell their crafting products. Banks in virtual worlds serve as an extension of player's storage space in which items and virtual currency can be stored and accessed, but only from certain locations in the virtual world, as opposed to players "inventory" which represent the storage space for items which are always accessible to players. Although professions are a single player activity, typically there are many players and NPCs present in the AOI (although the player does not interact with them). While

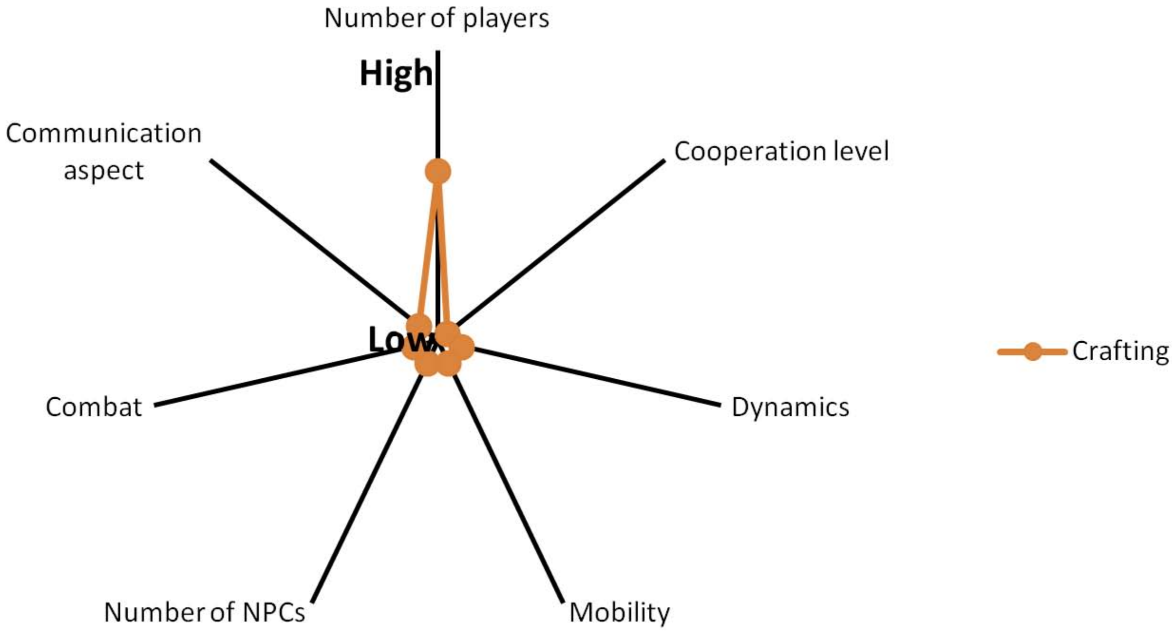


Figure 4.4: Scores of crafting on PSOVW parameters

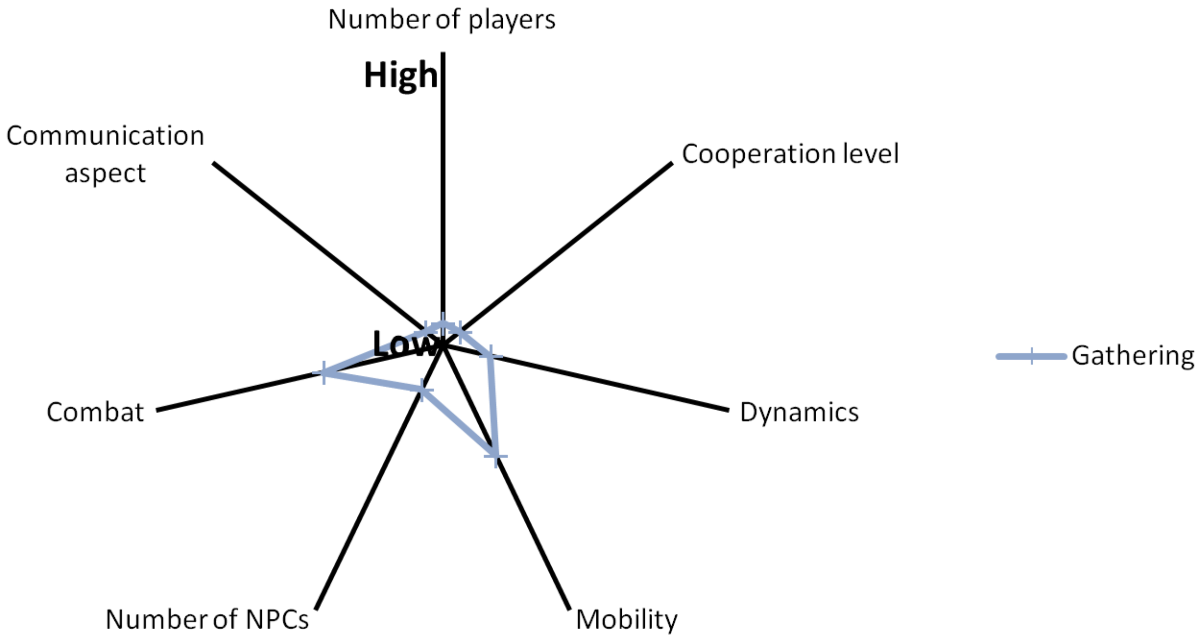


Figure 4.5: Scores of gathering on PSOVW parameters

crafting, the player must stand still in order to succeed in crafting an item, as moving cancels the action; hence, this action type may be characterized as having zero mobility. Crafting is a non combat activity. Crafting scores on the parameters of PSOVW are displayed in Figure 4.4

On the other hand, gathering professions are based on retrieving virtual items from various places in the world which results in high mobility of the players (e.g., upon gathering one mining node, a player moves to search for other one). Gathering is performed in areas of the virtual world with low player population, so number of other players in the vicinity is low. The number of NPCs is also rather low, but often nodes which can be gathered are in proximity to hostile NPCs so players trying to gather them sometimes have to engage in combat. Also, some gathering professions such as "skinning" are based on gathering items from dead mobs which requires combat. Scores of gathering on the PSOVW parameters are shown in Figure 4.5.

Questing is a player activity related to tasks (i.e., quests) assigned to players by the NPCs. Quests vary in nature and difficulty, but in general they always yield a certain reward in the form of items and experience points (e.g., a "mine warden" can ask the player to gather a certain amount of ore, while the "mine guard" can ask him/her to kill the robbers which have taken residence inside the mine). Thus, questing is a major activity for most players - especially

(a) Combat with multiple hostile NPCs

(b) Mounted duel with an NPC

Figure 4.6: Examples of questing

before they reach the maximum level - to gain experience, gold, equipment, and achievements, as well as to become familiar with the story of the game. Examples of questing situations in the virtual world are given in Figure 4.6.

Quests can be done in groups or by a player alone, but as indicated by Ducheneaut et al. [74], the players tend to play “alone together” and perform the tasks alone whenever the task is doable by one person. Questing can be performed together with other action types, as the goal of the quest may also require killing NPCs, venturing into dungeons in small groups, or even forming raids to beat the most formidable opponents. In most cases, questing requires a very small number of players participating. As this is mostly a single player experience, there is no need for specific coordination or communication techniques. Quests which require more than one player, require additional coordination amongst players, but present a small fraction of all available quests. Questing is an activity that mostly involves combat, although those fights are rather simple, repetitive, and do not require highly dynamic action rate from the player. Also, as quests usually include easy killing of more than one opponent, switching between them is needed which requires a certain amount of mobility. The number of NPCs in the AOI of the player may vary, but players usually take on one enemy at a time, which means that there is only one NPC actively interacting with the player. Scores of questing on the PSOVW parameters are shown in Figure 4.7.

Dungeons are instanced confined areas of the virtual world that only a limited number of players can enter. In dungeons, players in the group have the goal to fight their way through

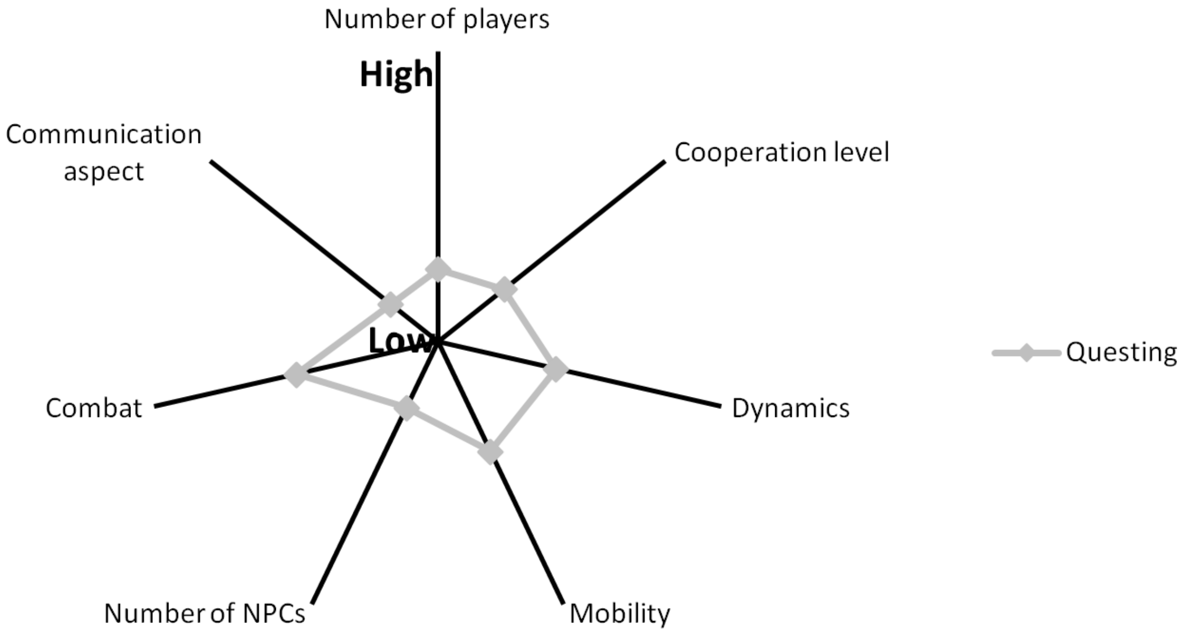


Figure 4.7: Scores of questing on PSOVW parameters

the dungeon and defeat all enemies, and especially bosses within the dungeon. Examples of players organizing at the start of the dungeons and fighting the NPCs in the dungeon are given in Figure 4.8. As dungeons are strictly designed for a given number of players, they constitute the main small group activity. Players which prefer group based combat will mostly involve themselves in dungeons, especially during the levelling period. Also, upon reaching a level limit players are offered better equipment in dungeons. Mechanisms similar to dungeons exist in all MMORPGs, enabling a group of players to jointly reach a goal which would not be attainable for a single player. In WoW there are also other incentives to encourage grouping, such as more difficult quests, which, for example, require more players and some forms of PvP combat, but dungeons are the primary grouping mechanism as they require a specific group setup, and also a relatively high time for completion, as will be shown later. Dungeons require an organized and highly coordinated group of players which often require additional communication. A group may, for example, involve a "tank", as a character who absorbs enemy blows and protects other members of the team; a "healer", as a character responsible for keeping up the health of the party; characters which inflict damage (often referred to as "DPS", meaning Damage Per Second); and those who temporarily control the mobs and make them lose control of actions and abilities (Crowd Control, or "CC", characters). Each dungeon instance is created for one

(a) Group of players preparing for a dungeon (b) Combat of players with a dungeon "boss"

Figure 4.8: Examples of dungeons

player group (in WoW, five players) so they can fight mobs inside the instance privately, without being interrupted or helped by the players outside the group. The instanced part of the virtual world is replicated for each new group that enters it, so at any one time there can be any number of instances of a certain dungeon on a single server, depending on the number of groups. As for the number of NPCs, in dungeons it is high and players in group can face up to tens of hostile, active NPCs. The NPCs in dungeons tend to be stronger so the combat is more challenging in dungeons than in questing, including more activity from each player (more dynamic player

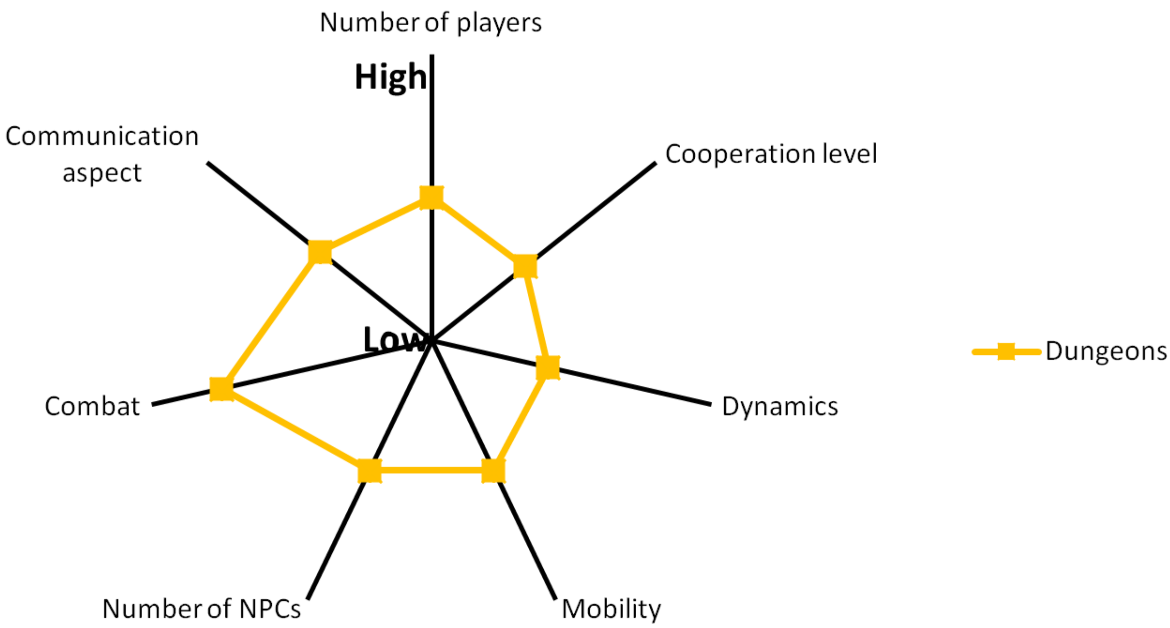


Figure 4.9: Scores of dungeons on PSOVW parameters

(a) Combat of players in a raid with hostile NPCs (b) Modified graphical user interface for easier raiding

Figure 4.10: Examples of raiding

input). Mobility in dungeons is also relatively high, as players are progressing through the instance in constant combat with small breaks. Scores of dungeons on the PSOVW parameters are shown in Figure 4.9.

While most dungeons are designed for a group of five players, there are also instances designed for larger groups of players (i.e., 10, 25, or 40), called **raids**. In general terms, we refer to dungeons as a "small group activity", and to **raiding** as a "large group activity" with the latter involving 10 or more players. Challenges for small and large groups in similar forms exist in most MMORPGs - for example, LOTRO has instances for small groups of six people, called fellowships, and raid instances for groups of two or more fellowships. Size of the raiding groups and the number of hostile NPCs can be observed from the examples of raiding situations in Figure 4.10. Raiding, as implied by its name, is an activity performed by raids, and it typically involves engaging in battle with very challenging and complex mobs. Raiding characteristics differ from dungeons, in terms of group size (typically, larger groups of players and mobs are participating), higher rate of actions performed (i.e., in raiding, a fraction of a second can decide the battle), and greater overall complexity. As the difficulty of the task increases, the rewards are higher - thus, in general, raids provide better quality items than dungeons, and dungeons' rewards are better than those obtained by questing. Raiding is one of the most complex tasks in a MMORPG, with very strict demands on players - not only on the required level to participate and the necessary equipment to perform certain tasks, but also on players' coordination and cooperation. As an example, we take a fight with Lady Vashj from first WoW expansion, *The*

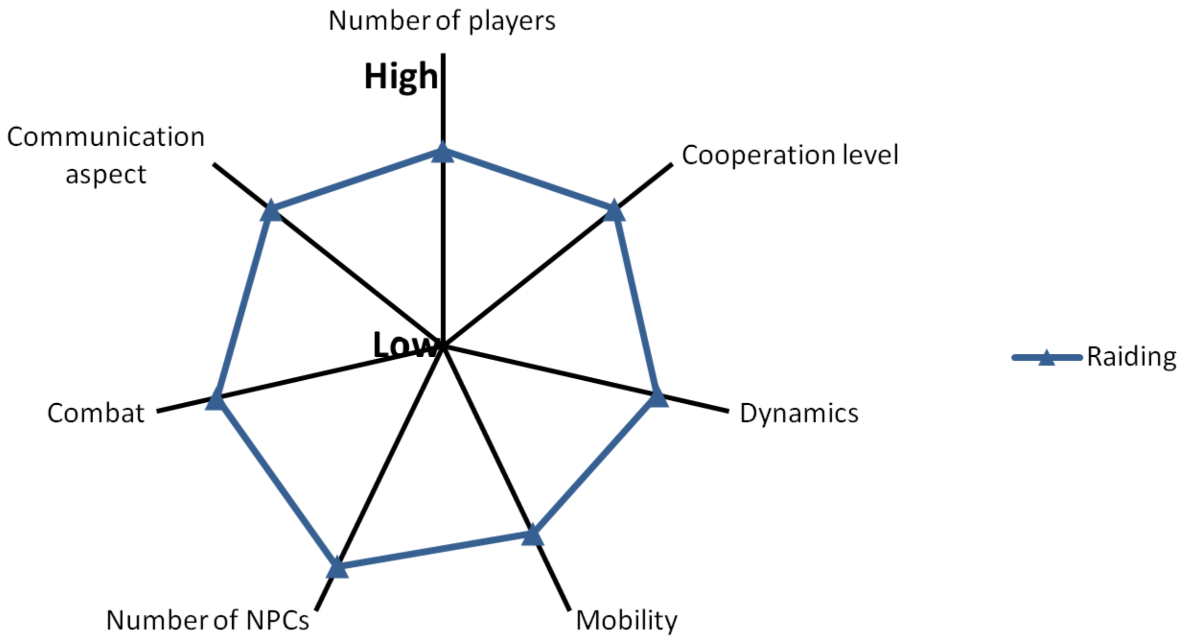


Figure 4.11: Scores of raiding on PSOVW parameters

Burning Crusade (TBC). The fight requires a raid of 25 players which needs to be strictly organized, with special positioning requirements, and raid setup requirement, and the ability to adapt to phases of the encounter, as described in the tactics guide [137]. Most guilds use voice communication as a rule while raiding. Progression made while raiding is mostly in terms of equipment, but also through achievements which are publicly acknowledged on the server, thus adding up to a person's "fame" in the virtual world. Although achievements do not change the power of the player's avatar, they have been shown to constitute a strong motivating factor for players' involvement in the virtual world. Ducheneaut et al. [74] offer the following explanation (on the purpose of audience in a MMORPG): "It is not the people that are addictive", but rather, "it's the image of myself I get from other people.", in other words - achievements basically present a way of "showing off" one's virtual self in front of other players. Mobility in raiding can be compared to dungeons, but as the combat is more challenging, it also requires slightly higher mobility from the players. Scores of raiding on the PSOVW parameters are shown in Figure 4.11.

PvP combat is a very important element of a MMORPG, and most MMORPGs have ways to involve players in combat with each other. For example, Darkfall Online bases the whole

(a) PvP combat in a battleground (b) World PvP combat

Figure 4.12: Examples of PvP combat

game on PvP combat rather than player versus environment (PvE) encounters. In WoW, PvP combat is realized in three different ways. The first way is “world PvP”, in which players, using the specific servers (PvP servers), can engage the members of the opposing faction in combat in most parts of the virtual world. In the two expansions, Wrath of the Lich King (WotLK) and Cataclysm, a whole zone in the world was especially dedicated for world PvP combat. Secondly, there are also “Battlegrounds”, instanced battlefields in which groups of players (i.e., 10, 15, or 40 players) can fight other faction members in order to achieve specific goals (e.g., capture the flag, acquire a certain amount of points, or, kill an enemy general). The third type of PvP combat is “Arena combat”, in which teams of two, three, or five players are engaged in instanced combat with each other. Variety of PvP combat situations with respect to the number of players can be observed in two screen shots displayed in Figure 4.12. In terms of complexity, PvP combat is one of the most complex action types in a MMORPG. Other than temporary goals, the only significant parameter differing between subcategories of PvP combat is the number of players which can vary significantly. In a 2v2 arena matches there are only four players, while in some battlegrounds the number of players participating in action can (theoretically) be up to eighty. Coordination of players can also vary, as in arena tournaments the players can be in strictly coordinated teams, with obligatory voice communication, while in some situations the players play “alone in a group” with no communication nor specific coordination with other members of the team. A typical battleground team in WoW consists of randomly selected enlisted players, who have the same goals set by the game rules, but do

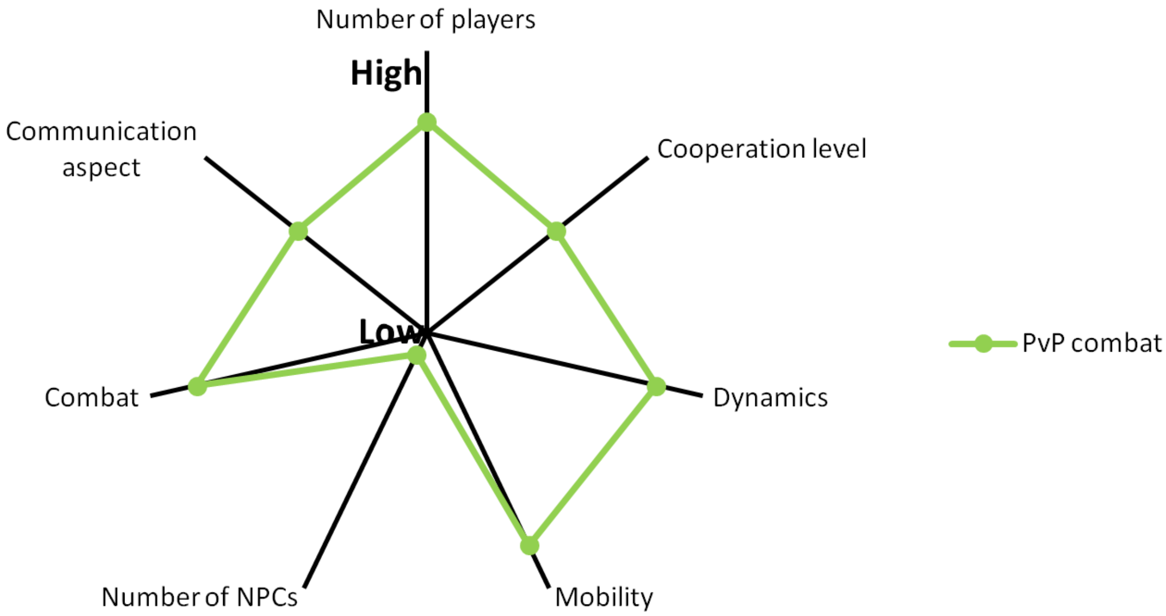


Figure 4.13: Scores of PvP combat on PSOVW parameters

not necessarily work in coordination in order to achieve those goals. Yet, all players actively participate in the combat so they can not be marked as inactive or observers. In PvP combat, the action rate is very dynamic, and the players are almost constantly on the move. The number of NPCs is usually low to none as the focus in this action type is on (human) players. Progression is made through achievements and equipment, which players buy for “honour” and “arena” points (earned in battles in battlegrounds and arena, respectively). Being victorious in battles also increases the position of the player on the competitive ladder. While the specific types of PvP combat differ in different games, with a number of players involved as the key parameter, we look at all aspects of PvP combat as one action type. Scores of PvP combat on the PSOVW parameters are shown in Figure 4.13.

In Figure 4.14 all defined user actions and their scores with respect to the parameters of PSOVW are shown. As overlapping of certain categories exists, there is a potential for consolidation. The crafting action category is the one with the lowest requirements, and is very similar to trading in all aspects, the only minor difference is that trading includes another active player. The graph indicates that the whole crafting space is a complete subset of the trading space and therefore we combine trading and crafting into one category. Gathering professions are usually heavily interleaved with questing (e.g., a player gets a task to kill ten giants, but while

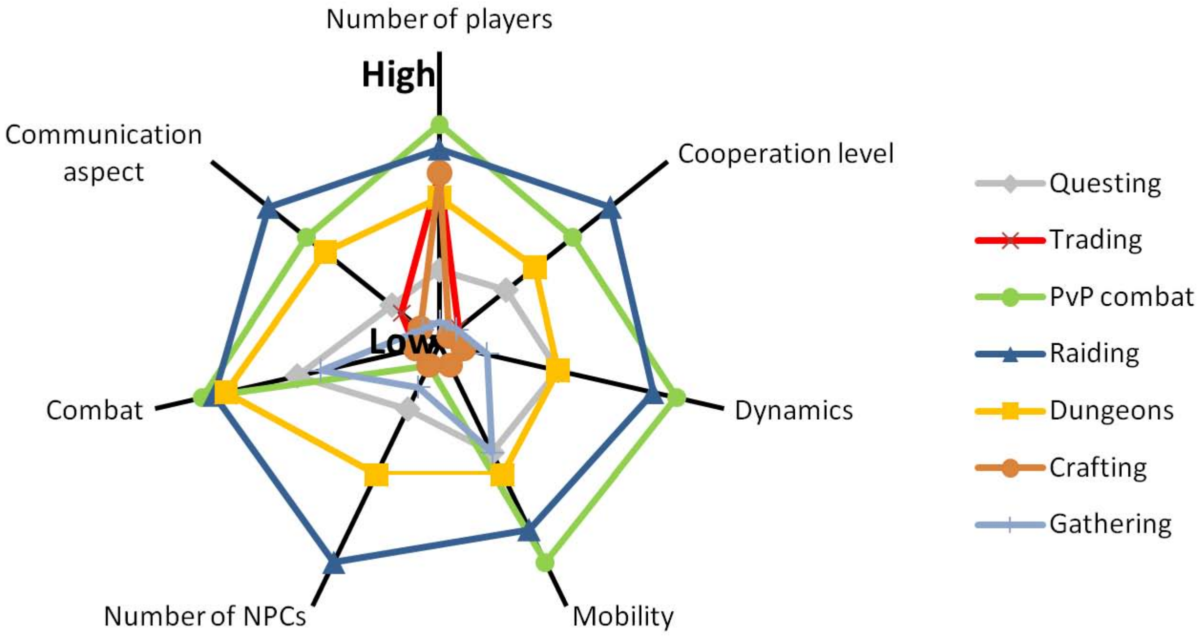


Figure 4.14: Scores of all identified player action categories on PSOVW parameters

searching for them he/she finds an ore deposit and mines it), and in Figure 4.14 we observe that gathering is a complete subset of questing, with slight differences on the number of users and NPCs participating. We have also decided to group those two action categories into one. While questing itself is a subset of dungeons, the differences on each scale are far too great for them to be grouped together. The result of this grouping procedure is the final categorization of user actions into following categories:

- Trading (and crafting);
- Questing (and gathering);
- PvP combat;
- Dungeons; and
- Raiding.

All of the final categories are distinct of each other in the space set by the parameters of PSOVW (Figure 4.15). Using this approach we have theoretically defined the user action categories and identified their categorisation. In order to validate proposed user action categories, we perform action specific measurements of network traffic.

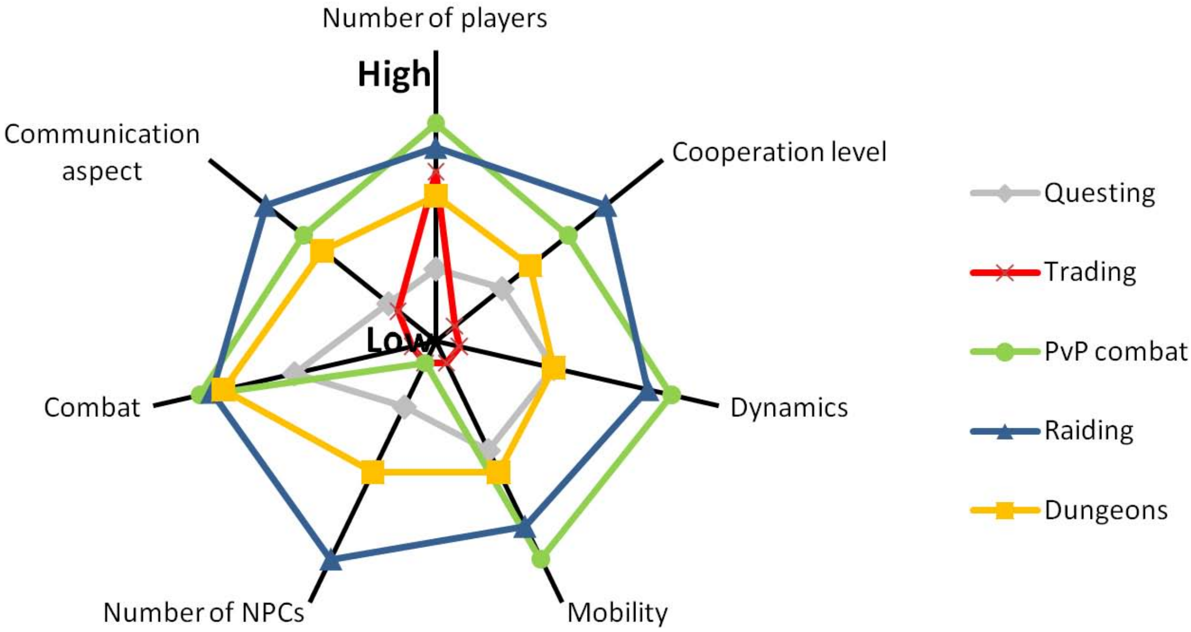


Figure 4.15: Scores of all consolidated player action categories on PSOVW parameters

4.3 Validation of user action categories through network traffic characteristics

The hypothesis under test is that that defined user action categories are distinct. We aim to validate this hypothesis trough inspection of the network traffic characteristics of each category. If the action categories, defined on application level characteristics, show significant differences on the level of network traffic, then the hypothesis is confirmed.

Network traffic characteristics of interest include packet size, packet inter-arrival time, throughput, and packet load from both the client and the server side. The packet inter-arrival time is the time interval between two subsequent packets. We assume that the inter-arrival time is related to the dynamics of user actions, types of interaction, and the processing load on the game platform, as well as in the network protocol stack. Throughput and packet load both represent the amount of traffic sent and received, where bandwidth usage refers to the part of used network capacity, while the packet load also shows the amount of processing for sending and receiving packets. Parameters such as packet size and throughput are assumed to have a correspondence to the number of users and NPCs, as well as on the cooperation activity level between them. Busse et al. [138] have also used most of these parameters for modelling their proposed QoS support for networked mobile gaming.

The analysis of network traffic characteristics is based on collected data traces. The traces were collected at the client, including the outgoing (client-generated), and the incoming (server-generated) traffic. In general, apart from the game owner, the Internet Service Provider (ISP) hosting the server, and the network operator, there is no easy (or perhaps even legal) way to collect traces on the server side. For the sake of simplicity, we will refer to the traffic originated from the client as the *client traffic* and to the traffic originated on the server side as the *server traffic*.

4.3.1 Measurements

In WoW each shard (server on which one whole replica of the virtual world is replicated) has its own IP address. For example, server on which the most of the traffic measurements is performed is EU-Bladefist with IP address 80.239.179.39 (later on it was changed to 195.12.234.219). If needed, the IP of each server can be obtained through simply connecting to each server in WoW client and noting the address from Wireshark or netstat utility in Windows.

In the time of the measurements WoW created only a single connection, but during the years and evolution of the game it was changed to two TCP connections. For both connections the port 3724 (blizwow) TCP is used. One connection is used for data regarding the game and the other for chatting between players. At the time of the measurements all data was sent through one connection. Additionally introduced battle.net service (Blizzard's service which integrates all of their games (WoW, Starcraft2 and Diablo3)) connection is created if the player has the "upgraded" WoW account to battle.net one (most players have done this). Port used for this connection is TCP 1119 (bnetgame). This connection is used for "presence" service and chatting with other players (which can be playing other games) through that service.

Also the ports 1120, 4000, 6112, 6113, 6114, and 6881 - 6999 may be used. The data from the official page eu.battle.net states the following:

"TCP 1119 - Login for Battle.net accounts, connecting to and playing on World of Warcraft realms TCP 3724 - Connecting to and playing on World of Warcraft realms as well as the Blizzard Downloader TCP 6881-6999 - The Blizzard Downloader (Peer-to-Peer functionality used for updating the game client) UDP 3724 - World of Warcraft Voice Chat feature TCP 1120 - Battle.net service, connecting to and playing on World of Warcraft realms UDP 1119 and 3724 for voice chat (Blizzard's VOIP solution)."

The traces analysed in this work only focus on the data during the course of the game. Only data on the port TCP 3724 is observed. Data regarding login and updates has not been taken into account.

The annotated, player action specific network traces described here were acquired from six real WoW players (volunteers). All players were experienced in WoW (active for more than one year), as the instructions for traffic gathering are very detailed and specific. Measurements have been performed in version WoW 2.x during The Burning Crusade expansion. The Wireshark software network protocol analyser (http://www.wireshark.org/) was used to capture the incoming and the outgoing traffic on a client.

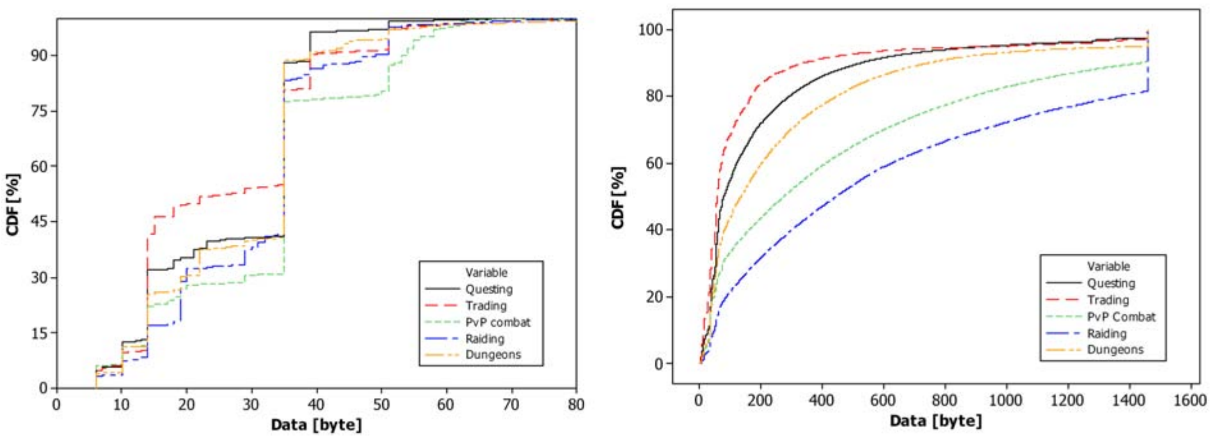
The measurement procedure was as follows: as the player was about to do a specific action in the virtual world, he/she would start data capture with Wireshark. For dungeons, raids, and PvP combat, the player would start the capture upon entering the instance (e.g., raid "Gruul's Lair", dungeon "Hellfire Citadel", PvP battleground "Arathi Basin"), and stop the capture when leaving the instance. For questing, the players started the capture upon receiving the quest and stopped it once they finished all questing related actions (i.e., logging out, or setting out to do some other action). Using gathering professions was allowed during questing. For trading related actions, the players were instructed to capture the session in which they tried to sell or buy something in the AH or from another player, including when checking in-game mail and/or bank for retrieving items. Also, for crafting professions, the players were instructed to start the capture before beginning to craft (at least ten items) and stop it as soon as they finish the crafting procedure. In the end, the players saved the trace into a file, and annotated it with a designation of what they did during that particular capture. It should be noted that the annotation followed the main activity categorization, meaning that, if the player was basically raiding and during that time briefly crafted something (like, for example, a health potion), the trace was labelled "raiding". Or, to give another example, if a character was in the dungeon and in the meanwhile completed a quest, the trace was labelled as "dungeons".

The biggest problem with the measurement process was the players' immersion in the game

- namely, the players often forgot to stop (as well as to start) the capture at the right moment, resulting in more than one type of player actions being represented in a given trace. As a result, through the filtering process, a large portion of taken measurement data was discarded and not taken into consideration. We captured 83 context specific network traces comprising of 1395940 IP datagrams. Interested parties may contact the author to obtain this data (anonymized), free of charge for use in research and education, under certain agreed conditions.

4.3.2 Packet size

Figure 4.16 shows the Cumulative Distribution Function (CDF) of the payload size of the client traffic (packet size without the TCP/IP headers). In general, the client payload size is very low which confirms the previous research results [59]. An average payload size across all action types is 36 bytes, which is less than the typical overhead of TCP/IP headers (40 bytes). The largest packet payload size is for PvP related actions, the next (in decreasing order) is for raiding, and the next for dungeons related actions. This confirms the hypothesis that client packet size is proportional to the rate of the player input. Also, we observed several values of the client payload size corresponding to specific action types. For example, raiding has a large number of packets with 19-byte size, and questing a large number of packets with payload size of 39 bytes. Difference of proposed categories is evident, especially between trading and PvP combat which have the biggest difference in values of user mobility and dynamics of user input.



(a) Client (b) Server

Figure 4.16: CDFs of packet payload size

The server-generated payload size distribution is shown in Figure 4.16b, and it may be noted that it is much wider than that of the client. The average payload size varies from 163 bytes for trading, up to 616 bytes for raiding. The amount of information sent by the server is highly dependent on the virtual world context, as the server has to send (a possibly large amount of) data about a number of entities it controls (e.g., mobs in the raid environment) and updates from other players. Again, differences amongst all categories are evident, with the highest average payload size of raiding (biggest number of both NPCs and players involved) over PvP combat, dungeons, questing to trading with the smallest packet sizes. This ordering can be anticipated from Figure 4.15 (moving from outer rim to the center of the graph in player number considering cooperation level and the number of NPCs).

4.3.3 Bandwidth usage and packet statistics

As shown in Figure 4.17 and Figure 4.18, the bandwidth usage in WoW can be remarkably low (under 10 kbit/s) but varies significantly (up to 10 times) depending on the situation in the virtual world. Comparing trading bitrate shown in Figure 4.17 with raiding shown in Fig-

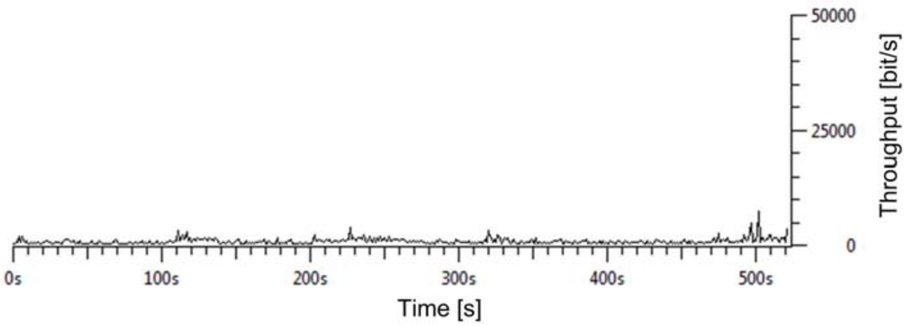


Figure 4.17: Bitrate (both client and server traffic) for trading

ure 4.18, one might think that they belong to different applications. The low bandwidth usage of MMORPGs is made possible by storing most of the media-rich elements (e.g., information about 3D graphics, video objects) of the virtual world locally, i.e., on the client (note that the complete WoW installation takes up 25 GB of hard disk space). What needs to be sent over the network includes only updates with position/orientation and actions of the players' avatars and NPCs.

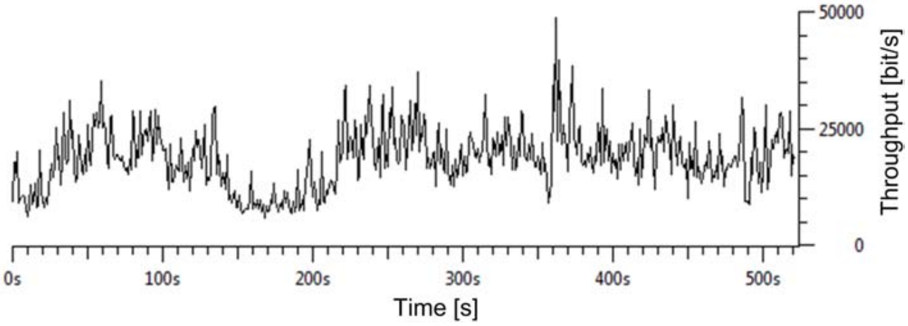


Figure 4.18: Bitrate (both client and server traffic) for raiding

Average bitrate per category (Figure 4.19) closely follows the definition of action categories (Figure 4.15) which validates the categorization and the logic behind it. Raiding generates the largest bitrate, followed by PvP combat, dungeons, questing, and trading in the end. Server traffic is a much more significant portion of the overall traffic with the ratio of server to client traffic bitrate varying across categories from 5 to 10.

The signalling overhead is the additional protocol related data that needs to be transmitted over the network in order to properly deliver the required application data. In our case we look

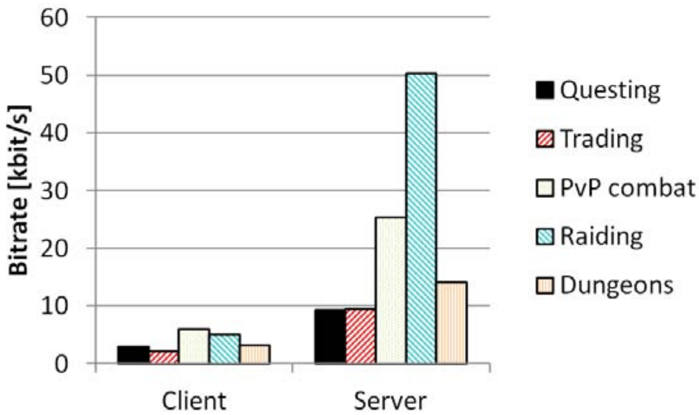


Figure 4.19: Average bitrate per category for client and server traffic

at how much signalling overhead is caused by only the TCP/IP stack. The measurements show that the TCP/IP headers are a significant part of overall sent data. Figure 4.20 illustrates the percentage of packets which carry a non-empty game data payload across the action categories in the overall data. The percentage of outgoing (client generated) data packets varies significantly, as PvP combat generates about double the number of data packets (58%) compared to trading (23%). The incoming (server-generated) traffic has an average value of 76% data packets with

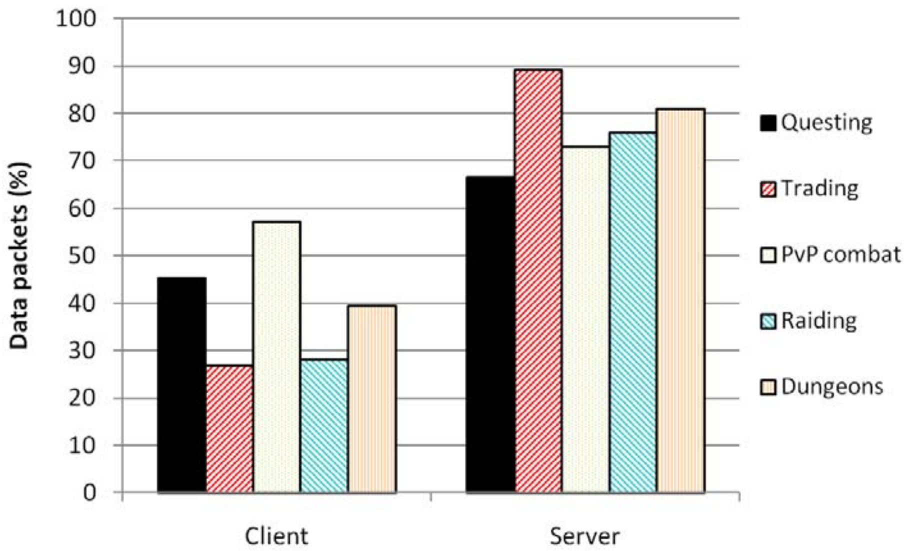


Figure 4.20: Percentage of packets with (non-empty) game data payload

a smaller variation than the client side, with the largest portion of data packets sent for trading (86%), followed by all other categories.

As shown in Figure 4.21, the overall packet rate per second is rather low, which confirms previous results [59, 50]. The differences of packet rate among action categories are high, because the packet load depends also on the dynamics of user input and mobility from the user side. This results in PvP combat having the highest packet rate, as it is the most mobile and dynamic action category. Trading again has the lowest requirements with under 10 packet per second in both ways which is remarkably low for a real time virtual world. Packet rate also confirms the hypotheses behind action category classification depicted in Figure 4.15.

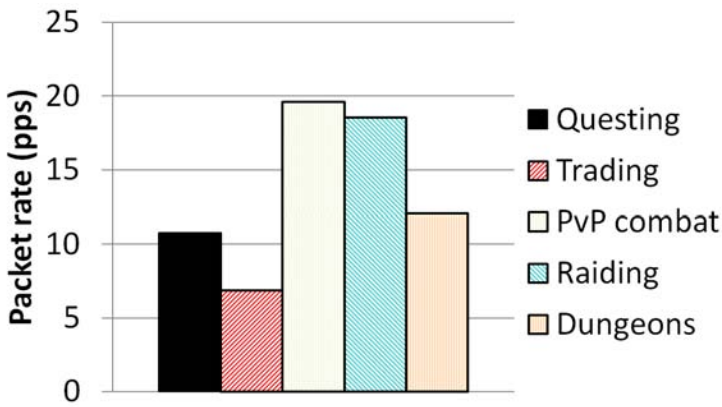
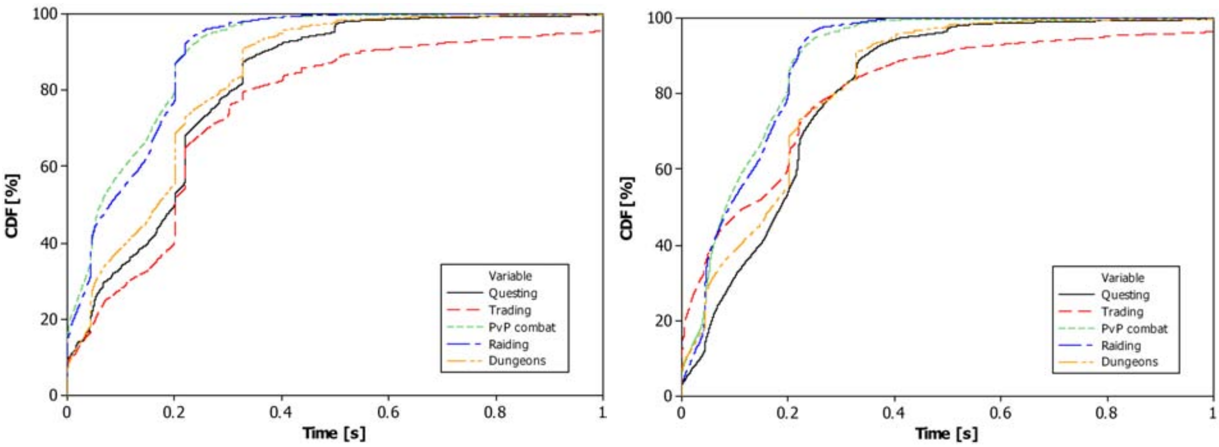


Figure 4.21: Packet rate



(a) Client (outgoing)

(b) Server (incoming)

Figure 4.22: CDFs of packets IATs

4.3.4 Inter-arrival times

Regarding the IAT, the assumption is that client IATs will be inversely related to the game action pace: the faster the action rate, and higher the mobility of the avatar, the lower the inter-departure time as more updates need to be sent. Figure 4.22a shows our assumption is confirmed as PvP combat with highest required input rate has the shortest client IATs (50% mark is around 0.05 seconds), followed by raiding, dungeons, and questing in the middle, and trading with longest IATs (50% mark is around 0.21 seconds). It may also be noted that 5% of the updates generated by the client in the trading category have inter-departure times greater than 1 second which is remarkably low for the real-time virtual world.

Almost 50% of all packets in all categories except questing have server IAT less than 0.1 second, as shown in Figure 4.22b. This can be explained by the reduced need (and, hence, frequency) for sending updates for other players' positions in questing and dungeons categories (players usually quest alone, and dungeons are limited to five simultaneous players). It seems likely that WoW client software autonomously controls the mobs and NPCs for a while without server side updates (passively applying the principle known as dead reckoning), thus resulting in fewer updates. It can be seen that raiding and PvP combat have, in general, the shortest IATs, as these categories require the most frequent updates, as well as the highest amount of data that needs to be sent. The high probability of a 200 ms IAT values can be explained with the Delayed Acknowledgment mechanism of TCP.

Based on the traffic characteristics identified for each of the action categories, we can conclude that defined action categories are distinct, and that the main hypotheses of the categorisation are confirmed. Mobility of the avatar, combat, and dynamics of player input heavily shape the client traffic, while the number of avatars and NPCs, and their coordination mostly influence the server traffic. Voice communication (VoIP) has not been taken into account thus far. Use of VoIP per action category will be studied in detail in the next chapter.

4.4 Network traffic models of player action categories

As the traffic of MMORPGs appears to be highly erratic, we do not model the overall traffic characteristics in a way agnostic of user behaviour, but perform the modelling across a set of action categories which represent different situations in the virtual world. Through this methodology we aim to model the traffic more accurately and to exploit the relationship between the application state and the network characteristics. We want to provide a source based traffic model, as by inspecting player behaviour, we model the player as a source of the traffic. Therefore we can not inspect just the sizes and IATs of IP datagrams captured on the link, but we need to inspect Application Protocol Data Unit (APDU) sizes and IATs. It will be shown later that such a source model is more adaptive and realistic than "traditional models" which only describe the IP datagram sizes and IATs observed on the link [2].

We further illustrate the concept of APDU. At a certain point time, an application needs to send a specific amount of data which we call APDU. This data is encapsulated into a TCP segment, which consists of a TCP header and TCP payload. Maximum Segment Size (MSS) is the parameter that specifies the largest amount of data, specified in octets, that a computer or communications device can receive in a single TCP segment. The length of the header is not included in the MSS. If the size of the APDU which needs to be transmitted is larger than MSS, then the TCP/IP protocol stack breaks the APDU into fragments and sends it through multiple TCP segments. For example, if the APDU size is $n * MSS + x$, the number of generated TCP packets will be $n+1$ where the data size of n TCP segments will be MSS, while the last TCP segment will carry data with size of x.

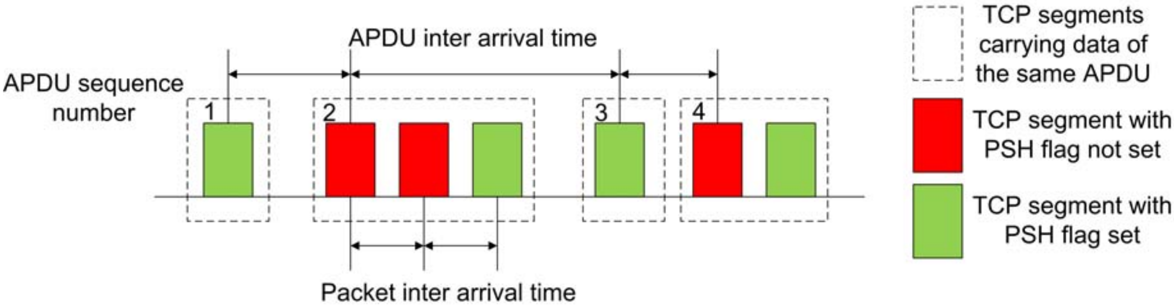


Figure 4.23: Sending of APDU in WoW

In order to determine APDU size from the network traffic trace, we have implemented an algorithm proposed by Svoboda et al. [59]. The authors noted that some TCP segments in WoW network traffic traces had the TCP PSH flag set which is the case when the application has data that needs to be sent across the network immediately. According to the assumption, if APDU is larger than the MS, the TCP service splits the APDU into multiple segments and sets the PSH flag for the last segment in the sequence. If the APDU is smaller than the MSS, the whole APDU is sent as one segment with the PSH flag set. We follow this approach by processing our dataset in order to calculate the correct APDU sizes and inter-arrival times between subsequent APDUs, and modelling those size values (not the segment size or segment IAT). The relationship between APDUs regarding single TCP segments is further depicted in Figure 4.23. We note that there is a large TCP signalling overhead in our traffic trace, as the TCP ACK packets carrying no data payload are quite common. For the modelling procedure we performed the filtering of the trace in which we excluded TCP ACK packets with empty payload, but we have noted the percentage of those for each category.

Traffic modelling in the area of network games is based on analytical traffic models (i.e., mathematical description). These models are easier both to convey and to analyse compared to empirical models of traffic (e.g. tcplib [78]). In our modelling procedure, we follow the approach for application traffic modelling pioneered by Paxson [79], firstly used in the area of network games by Borella [80]. The algorithm is also described in detail in [81]. We have fully presented the algorithm used for obtaining the traffic models in Section 3.4.4. The algorithm was repeatedly applied for each defined action category. In Tables 4.1-4.5 we depict the obtained models across categories for APDU sizes and APDU IATs for both client and server

Table 4.1: Network traffic model parameters for *Trading*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	15612	Deter. p=5.25% Deter. p=4.21% Deter. p=34.05% Deter. p=5.72% Deter. p=3.19% Deter. p=32.50% Deter. p=9.14% Deter. p=5.94%	$a = 6$ $a = 10$ $a = 14$ $a = 15$ $a = 18$ $a = 35$ $a = 39$ $a = 51$	0.0911	32(0.20)/0	0.24
Client IAT	15602	Weibull p=50.53% Weibull p=28.53% Deter. p=17.60% Deter. p=3.34%	$\gamma = 0.99, \alpha = 176.74$ $\gamma = 0.66, \alpha = 1220.33, \mu = 500.95$ $a = 0$ $a = 500$	0.0817	10(0.06%)/-	0.29
Server APDU size	27082	Lognormal	$\mu = 4.16, \alpha = 1.15$	0.0888	145(0.54%)/-	0.05
Server IAT	27081	Lognormal p=82.68% Deter. p=9.62% Deter. p=7.7%	$\mu = 5.62, \alpha = 0.95$ $a = 200$ $a = 218$	0.1063	23(0.08%)/-	0.17

traffic. Also, we present the goodness of fit of the models in terms of discrepancy measure, and details about the dataset from which the model has derived.

Client APDU sizes include several discrete steps with very low values so a great majority of client APDUs fits inside one packet for all categories. We modelled them as deterministic with several values. The most frequent values of payload size vary constantly across different actions, but packets of size 35 B are the most frequent, which is in contrast with the work of Svoboda et al. [59] who modelled WoW client traffic with packets of size 6 B, 19 B and 43 B. We are assuming that these packets are responsible for carrying information about character's movement as suggested in [99].

Client side APDU IATs can be divided in two sectors for single player based categories (questing and trading), below 500 ms and above. For both areas, Weibull distribution showed as a good fit. Also, significant "spikes" exist at values 0 ms and 500 ms, which we assume to be due to dynamics of player activity. Subsequent packets (0 ms IAT) are sent while player is performing a highly dynamic action (e.g., highest percentage of 0 ms IATs is in *PvP combat*), while we assume that the frequent occurrence of 500 ms IAT is probably due to some sort of keep alive mechanism. Such mechanism would also explain why the IAT dataset is divided in two sections modelled by two distributions with a limit around 500 ms. The group based action categories have "smoother" CDF curves, due to the fact that they are more dynamic and time-outs occur more rarely than in single player actions. Those have been modelled mostly

Table 4.2: Network traffic model parameters for *Questing*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	63541	Deter. p=4.96% Deter. p=7.34% Deter. p=20.75% Deter. p=2.82% Deter. p=2.36% Deter. p=50.18% Deter. p=9.20% Deter. p=2.39%	a=6 a=10 a=14 a=18 a=21 a=35 a=39 a=51	0.0415	11(0.02%)/0	0.46
Client IAT	63531	Weibull p=55.7% Weibull p=12.6% Deter. p=16.46% Deter. p=15.24%	$\gamma = 1.19, \alpha = 236.22$ $\gamma = 0.84, \alpha = 1073.63, \mu = 525.95$ $a = 0$ $a = 500$	0.1608	101(0.17%)/-	0
Server APDU size	99163	Lognormal	$\alpha = 1.22, \mu = 4.55$	0.2304	163(0.16%)/-	0.05
Server IAT	99177	Normal p=71.51% Weibull p=7.49% Deter. p=2.15% Deter. p=12.27% Deter. p=6.58%	$\mu = 212.87, \sigma = 96.59$ $\gamma = 0.91, \alpha = 451.55, \mu = 419.96$ $a = 44$ $a = 218$ $a = 328$	0.1364	47(0.48%)/-	0.21

with one fit of the Weibull distribution coupled with several discrete values.

Server side APDU size has a good fit in Lognormal distribution with some discrete steps (usually at 37 B) for single player based categories (questing and trading). On the other side, we have noted that spikes occur in histograms around 7000 B, probably related to loading instances where significant data must be transported. We came to this conclusion as these spikes occur only in action categories which are related to instances (i.e., *Dungeons, Raiding, PvP combat*). While these APDUs have rather low probability, we took them into account in the modelling procedure as they carry significant amount of overall data (e.g., only 1.5% of those APDUs

Table 4.3: Network traffic model parameters for *PvP combat*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	66635	Deter. p=7.63% Deter. p=5.60% Deter. p=13.12% Deter. p=3.11% Deter. p=59.50% Deter. p=6.66% Deter. 4.38%	a= 6 a=10 a=14 a=19 a=35 a=51 a=58	0.1307	0/0	0.39
Client IAT	66631	Weibull p=78.4% Deter. p=20.18% Deter. p=:1.42%	$\gamma = 0.79, \alpha = 208.50$ $a = 0$ $a = 500$	0.0681	155(0.23%)/-	0.24
Server APDU size	71594	Weibull p=93.08% Largest Extreme Value p=0.73% Deter. p=6.19%	$\gamma = 0.92, \alpha = 538.59$ $\mu = 7754.99, \alpha = 394.83$ $a = 37$	0.0183	14(0.02%)/-	0
Server IAT	71594	Weibull p=83.32% Deter. p=4.13% Deter. p=4.11% Deter. p=8.44%	$\gamma = 1.71, \alpha = 193.26$ $a = 44$ $a = 200$ $a = 328$	0.1799	97(0.14%)/-	0.22

Table 4.4: Network traffic model parameters for *Dungeons*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	50460	Deter. p=4.57% Deter. p=8.00% Deter. p=16.28% Deter. p=4.04% Deter. p=8.28% Deter. p=55.70% Deter. p=3.13%	a=6 a=10 a=14 a=19 a=22 a=35 a=51	0.1048	55(0.04%)/0	0.27
Client IAT	50460	Weibull p=95.86% Deter. p=4.14%	$\gamma = 0.58, \alpha = 268.37$ $a = 500$	0.2038	221(0.44%)/-	0.33
Server APDU size	96035	Weibull p=99.15% Largest Extreme Value p=0.85%	$\gamma = 0.89, \alpha = 221.83$ $\mu = 7698.83, \alpha = 198.842$	0.0331	63(0.07%)/-	0.01
Server IAT	96056	Weibull p=78.35% Weibull p=2.58% Deter. p=3.06% Deter. p=9.55% Deter. p=6.46%	$\gamma = 2.28, \alpha = 231.3$ $\gamma = 0.79, \alpha = 344.14, \mu = 405.96$ $a = 44$ $a = 200$ $a = 328$	0.1443	32(0.03%)/-	0.16

carried more than 10% of all transmitted data). This is why for those categories we apply a mixture model of Weibull and Largest Extreme Value distribution.

Discrete steps for server side IATs have been observed at 44 ms, 200 ms, 218 ms and 328 ms. While the step at 200 ms can be explained with the TCP delayed ACK mechanism, the rest of the steps appear to be inherent to the WoW application protocol. Server IATs have, in general, been the most complex to model. We usually modelled them with the combination of two distributions (Weibull, Normal, Lognormal) and several discrete values.

Table 4.5: Network traffic model parameters for *Raiding*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	19136	Deter. p=3.81% Deter. p=4.35% Deter. p=12.15% Deter. p=20.18% Deter. p=3.63% Deter. p=6.81% Deter. p=45.53% Deter. p=3.54%	a=6 a=10 a=14 a=19 a=20 a=29 a=35 a=51	0.1022	19(0.1%)/0	0.27
Client IAT	19135	Weibull p=85.73% Deter. p=14.27%	$\gamma = 0.76, \alpha = 299.52$ $a = 0$	0.0898	65(0.34%)/-	0.27
Server APDU size	37801	Weibull p=98.97% Weibull p=1.03%	$\gamma = 0.86, \alpha = 941.79$ $\gamma = 0.91, \alpha = 1183.28, \mu = 7298.20$	0.0342	16(0.04%)/+	0.16
Server IAT	37801	Weibull p=84.39% Deter. p=9.55% Deter. p=6.06%	$\gamma = 1.99, \alpha = 188.92$ $a = 44$ $a = 200$	0.0660	6(0.02%)/-	0.03

4.5 Summary and outlook

In this section we have inspected the characteristics of the PSOVW in detail. Based on the identified metrics we have categorised the possible situations of the virtual world into five action categories: trading, questing, PvP combat, dungeons, and raiding. The performed categorisation is novel, and it is performed through detailed analysis of the application state of the game. Similar classifications in previous related work used for traffic measurements [2, 10] are not supported by an application level analysis and seem arbitrary. Based on identified action categories we have performed action specific measurements of network traffic. Through inspection of traffic characteristics we have validated the categorisation as defined action categories are distinctive on the network level. Finally, we present the network traffic models for each action category.

In the next chapter we describe measurements of behaviour in terms of defined action categories. Also, based on the results of those measurements, a behavioural model is proposed.

Chapter 5

User behaviour model

Looking at the real human life, it is evident that there is a potentially high degree of randomness in possible activities, movement, interactions, etc. Nevertheless, there are easily identifiable patterns in almost every person's life. These patterns can be found on several time-scales: from the daily routines of getting up, eating, working, to weekly patterns such as playing sports on weekends, and even to yearly patterns like going to the seaside in the summer. In the virtual world, the rules regarding patterns of behaviour also apply. From hourly patterns such as performing large group actions in the evening, when the highest number of players is available, to daily patterns, such as spending more time in the game during weekends. User behaviour patterns in MMORPGs can be used to estimate and predict both network traffic and computational load. A good estimation of traffic load is essential for planning the network aspect of a MMORPG.

Although it has been shown that traffic of online games has certain properties such as large, high periodic bursts of very small packets with predictable long term rates [139], due to the increasingly large number of users and different behaviour of those users, MMORPG traffic – especially in the upstream direction – is difficult to predict and model. We aim to model the MMORPG traffic not just on the packet level by describing the statistical properties of observed traffic, but on the source level, analysing the sources of the traffic further and in more detail. In order to create a source model of the MMORPG traffic, we need to understand not only the relations between generated traffic and the context/situation in the virtual world, but also the application level behaviour of the players. This approach requires a better insight into *what, when,* and *why* players do what they do in the virtual world. In previous sections we provided answers to the question *what* players do in the virtual world through categorization of virtual world state to action categories. In this section, we extend the *what* question to cover the whole playing session, including not just the bursts of activities, identified thus far, but also the time in-between actions, and personal activities such as voice communication and chat. We also answer the question *when*, by identifying player behaviour patterns, and the question *why*, through examining relationship between motivations and behaviour.

We study user behaviour on both single user session level, and service level. On a user level, topics of interest include length of a user session, actual set of actions performed during the session, and user motivation. On a service level, the topics of interest are the number of active users, arrival and departure rates. Also, we aim to understand and model both hourly and daily patterns of all listed parameters.

In this section we first describe the measurements and methodology, followed with the results regarding session characteristics and user behaviour. The next issue investigated is the relationship between player motivation and behaviour patterns. Based on obtained results, in the end we present the models of player behaviour.

5.1 Additional aspects of player behaviour

We introduce an additional action aspect of player behaviour – *Uncategorised* which indicates lack of any classifiable events in the virtual world (i.e., time played not belonging to either action category). We assume that uncategorised portions of the session consist of: 1) the time spent inactive, or Away From Keyboard (AFK), 2) waiting for some events to take place (e.g., waiting for the guild officers to pick the people for the raid), and 3) other actions not belonging to any other category. The latter include actions of continuous killing monsters for profit (commonly known as “grinding”), chasing achievements, exploring, etc.

Also, we identify *Communication* which indicates how much do players textually communicate between each other (i.e., chat). Communication can occur concurrently with other player action categories. Other categories are strictly distinct (e.g., player can be communicating and raiding at the same time, but can not be in PvP combat and dungeons simultaneously). In communication category, we only consider time periods in which the messages are sent through the inbuilt chat client. We do not consider the player as communicating when chat messages are just received, as the players can be "joined in" many chat channels, in which they may (or not) actually participate.

5.2 Measurements and methodology

We performed three measurement procedures in order to obtain all required data for behaviour modelling process:

- Set 1 - initial measurements, described in detail in [8];
- Set 2 - the largest set of measurements in which the modelling presented here is based, described in detail in [60]; and,
- Set 3 - measurements aimed to further explore what is actually done during parts of the session belonging to *Uncategorised*.

Set 1 - preliminary measurements considering the user behaviour were taken over the six weeks from November to December 2008 following the release of the second expansion for WoW "Wrath of the Lich King" (WotLK), on 13th November 2008. These measurements were done with the help of 11 volunteering WoW players. For the purpose of studying the player in-game behaviour, we developed a WoW add-on named World of Warcraft Session Activity Logger (WSA-Logger) [140], by using ACE3 framework [141] and Blizzard Entertainment's WoW Application Programming Interface [142].

Set 2 - the second set of measurements was done through a student project in which students needed to find at least five WoW player volunteers who agreed to install and use the WSA-Logger and participate in this research. To find player volunteers students referred to their colleagues who play WoW, or their acquaintances from the game since some of the students were playing WoW themselves. Online communities such as WoW Internet forums were another source of volunteering players. No personal data about the participating players, other than their age, was gathered. The described method of obtaining the player sample resulted in somewhat younger player sample compared to the general population of MMORPGs reported

Table 5.1: Players' sample age comparison

Age range	Our sample	Williams et al.
Teens 12-17	6.73%	6.45%
College-age 18-22	43.27%	12.40%
Young adult 23-29	38.46%	26.27%
Thirties 30-39	8.65%	36.39%
Forties 40-49	1.92%	12.40%
Fifty or older 50+	0.96%	4.80%

by Williams et al. [49], as shown in Table 5.1 (average age 24 compared to average age 33, respectively). Monitoring period for the second set of measurements was from May 5th, 2009 to June 21st, 2009, and the measurements have been performed in WoW version 3.x, with the total number of 104 participating players. While the first set of measurements was performed right after the launch of new game content (i.e., WotLK expansion) resulting in a non typical user behaviour, the second set was performed in the time frame in which there was no recent release of additional content in order to observe typical player behaviour. Three factors that affected the size of the player base were that players needed 1) to volunteer in order to participate, 2) to install the WSA-Logger, and 3) after collecting, submit the gathered data. However, the relatively small number of players involved was compensated with the high level of information that we managed to gather during the process, since the traces enabled a much more detailed examination of each player's behaviour. Information about the use of voice communication during the game session was also obtained through an additional questionnaire filled in by the volunteering players. The source of the add-on was made open so as to address players' concerns regarding privacy.

Set 3 - the final set of measurements was performed under the same terms as the second set (i.e., as a student assignment with the same rules). The goal of the measurements was to further investigate *Uncategorised* by tracking players location and movement patterns. The monitoring period for these measurements was from May 15th, 2011 to June 10th, 2011 in WoW version 4.x (The Cataclysm expansion) with 15 players participating in the measurements.

5.2.1 Motivational assessment and voice communication questionnaire

The *Motivations Assessment* designed by Yee [143] is a survey consisting of 39 multiple-choice questions with answers offered in form of 5-point Likert scale. This is an example of the question form used in the assessment: "*How important is it to you to be well-known in the game?*", with following possible answers: "1) *Not Important At All, 2) Slightly Important, 3) Somewhat Important, 4) Very Important, 5) Extremely Important*". At the end of the assessment, player's percentile ranks in the 10 motivation subcomponents against a sample of 3200 original respondents are shown in several graphs. Each participating player filled in the assessment and reported his/her percentile ranks on every motivational component and subcomponent (e.g., component *Achievement* and subcomponent *Mechanics*). The results of the motivational assessment, coupled with other behavioural measurements were used to validate the hypotheses regarding the relationship between players' motivation and their actual behaviour. As indicated by Yee [39], the descriptions for each motivational subcomponent emphasize what it means to score high on the specific subcomponent. Scoring low on these subcomponents reveals that a player is not interested in those motivators. Therefore, while testing our hypotheses we have looked only at the players with score rank over 80 out of 100 in a particular motivational subcomponent. Also, for the negative relations we have looked at the players who scored lower than 20. These limits were determined in several tests as setting the 90/10 limit resulted in too few players to perform the analysis on. Setting the bars to 70/30 and lower would be in contrast to the definition of the motivational components. Each of the hypothesis relates one motivational subcomponent to one action category. The algorithm for testing each hypothesis is as follows:

1) Extract the players with rank score over 80 and rank score below 20 in specific motivational subcomponent (N_{high}) and (N_{low});

2) For each of the extracted players compare the time spent in the target action category with the average value of the whole sample (these values are determined through the analysis of the data obtained with WSA-Logger);

3) Observe the number of players with higher than average time spent in the targeted action category in the player group with rank score over 80 (M_{high});

4) Observe the number of players with lower than average time spent in the targeted action category in the player group with rank score below 20 (M_{low});
5) Calculate the percentage of players conforming to the hypothesis in the following way $(M_{high} + M_{low})/(N_{high} + N_{low})$.

The questionnaire we used for the investigation of players' voice communication preferences consists of 5 multiple choice questions regarding the frequency of voice communication usage throughout action categories, with answers offered in form of 5-point Likert scale. The form of the questions conforms to the example: "*How often do you use voice communication while in Raiding instances?*" with following possible answers: "*1) Always, 2) Often, 3) Sometimes, 4)Rarely, 5) Never*". Also, we investigated the general use of VoIP while playing and popularity of common VoIP applications (i.e., "*Do you use a VoIP program for voice communication while playing, and if yes, which one?*") The full questionnaire can be found in Appendix A of this thesis.

5.2.2 Data gathering: WSA-Logger

The WSA-Logger is an add-on for WoW written in LUA programming language. It records the events fired by the WoW API when a certain action is performed in the virtual world. For example, the *Bank_Frame_Open* event is fired once the bank interface is displayed and the WSA-Logger notes the the event as trading. We have tracked only the events that could be classified as action specific (e.g., entering a battleground instance for PvP combat). As an output, the add-on generates a log file which consists of the chronological list of all events that were collected during the monitored period. Each event in the log file is represented by the date, time (hours, minutes, and seconds), day in the week, event name, and player action type. By analysing this event list, we can extract the data about player behaviour in terms of specific categories (i.e., retrace all relevant player actions). The complete list of events tracked is provided in Appendix B.

The add-on is designed not to interrupt or modify player gameplay in any time so it runs in the background with no indication of its activity. It does have a GUI which enables testing of proper functionality as indicated in Figure 5.1, where we can see the simple GUI, and testing

Figure 5.1: WSA-Logger GUI and testing screenshots

phase in which the add-on displays the triggered events on screen. Also, it is designed to be as lightweight as possible (in terms of memory usage).

As the game has been constantly evolving and changing, so has the WSA-Logger. As the new content is produced, it needs to be added into the add-on (e.g., new areas in the virtual world.). For the second set of measurements compared to the first version [8], the functionality of WSA-Logger has been extended in order to track written communication (text chat). WoW client has simple and effective chat mechanisms. There are several chat commands which enable communication amongst players. Communication command */say* is used to communicate with people who are nearby in the virtual world, */yell* is a similar command allowing the people on a larger virtual territory to see the message. Players joined in a guild can always communicate with each other, no matter what their position in the virtual world is, through a specific, guild-only channel. Players can also use private chat channels to talk to each other. In general, there are many channels and players can easily create new ones. In addition to text messages, there are "emotes", messages which are done by the virtual character (e.g., */wave* causes the character to wave). All of these types of outgoing messages are tracked. The downside of message tracking is that the log files generated by the add-on have significantly increased in size. The second modification was made in order to increase the precision of the add-on. The previous version tracked time based only on *GetGameTime()* method from WoW API, which only returned time in hours and minutes. By using an additional library *GameTime:Get*[144], we have increased the precision to one second. Compared to other work in this area, this

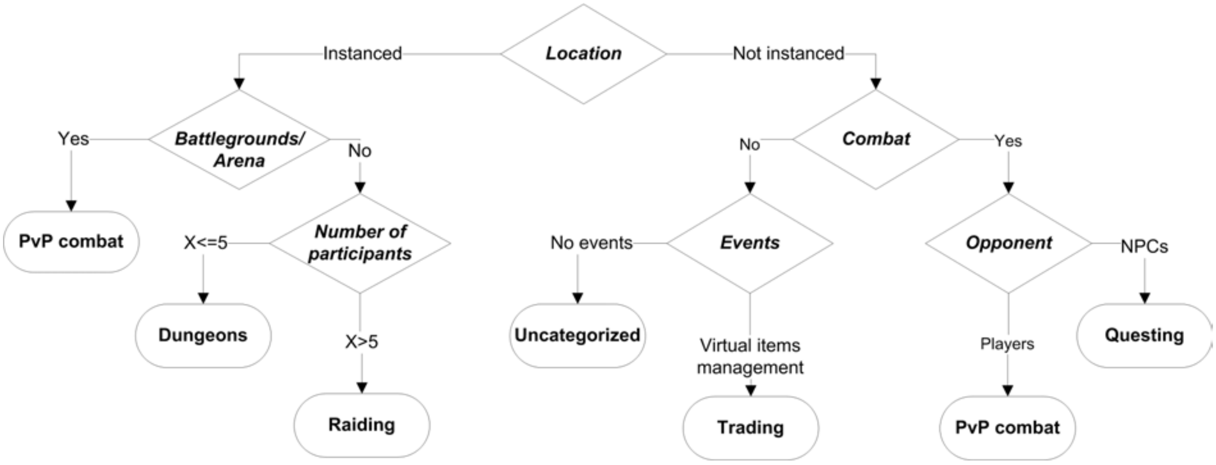


Figure 5.2: Determining the action category based on WoW API event

is currently, as of December 2011, the highest achieved event monitoring precision. The only disadvantage of the library is that for the very first minute of the session, seconds are not tracked. This resulted in larger size of log files, and higher memory usage of the add-on.

For the third set of measurements WSA-Logger was extended and improved in order to add a GUI display of gathered data, and reduce the load on the system. Also, the add-on was changed in a way to track movement patterns of the player in terms of accessed areas of the virtual world. Additionally. WSA-Logger was renamed wJournal due to the fact that the previous name reminded the players of malicious software used to steal information on player accounts commonly called “key loggers”. The latest version of wJournal can be obtained from [140].

The relationship between WoW API events and player action types was determined by using a decision tree, shown (simplified) in Figure 5.2. For some actions, the decision is more complex, while for others it’s rather simple. For example, when the *Auction_house_show* event is fired, once the auction interface (used for trading) is first displayed, the WSA-Logger notes the date, time, and player action type as “trading”. All mail, bank, trade, and profession related events in the WoW API (e.g., *Trade_accept_update*) are interpreted as “trading”, all quest related events (e.g., *Quest_accept_confirm*) are interpreted as “questing”. PvP related events are noted in several different ways: by using *Zone_changed_new_area* and checking is the active area instanced battleground or arena, through events that confirm that the player is located in a battleground or in arena (e.g., *Update_battlefield_score*), and also events (e.g., *Chat_msg_combat_honor_gain*) which indicate that there is active battle amongst players regardless of the location (i.e., world PvP combat). Dungeons and raids are only noted by using the *Zone_changed_new_area* event, which is fired whenever a new zone is loaded (e.g., when the player enters or leaves an instance), and a list of names of all currently active dungeon and raid instances (as this event may also be fired in some other occasions). This helped determine the time interval which the player spent within the instance. If the player participates in several instances in a sequence, then the total time (combined) is taken into account. All events indicating any form of the communication (e.g., *Chat_msg_say)* are registered and stored in the log file.

5.2.3 Data analysis and filtering

For the analysis of the log files provided by the add-on, we developed a log file parser in Java. The monitored session characteristics include overall session duration and player activity within a session (i.e., player behaviour in terms of action performed). As for the session duration, we define it as the time interval between the time when the player logs in to the game and the time of the corresponding logout from the game.

We define a "session segment" as a part of the playing session in which the player performs only actions from a specific category. Session segments of dungeons, raiding, and instanced PvP combat are labelled with their start and end time (there are specific events labelling entrance and exit from the instanced areas), and segments of questing, trading, communication, and non-instanced PvP combat are noted as a chain of events. So, one raiding segment is represented with the time between entrance and exit of the raiding instance. If a player enters a raiding instance subsequently, it is treated as one raiding segment. For non-instanced categories, such as trading, an action specific session segment is defined as the time between the first and the last event in the chain of events of the same category. Time between two events of different categories is labelled as belonging to the second category, as we assume that the change of context occurs when the player performs last action in the sequence of actions of one type. If the time between two events of a specific category is larger than the timeout period of 5 minutes, it is labelled as uncategorised.

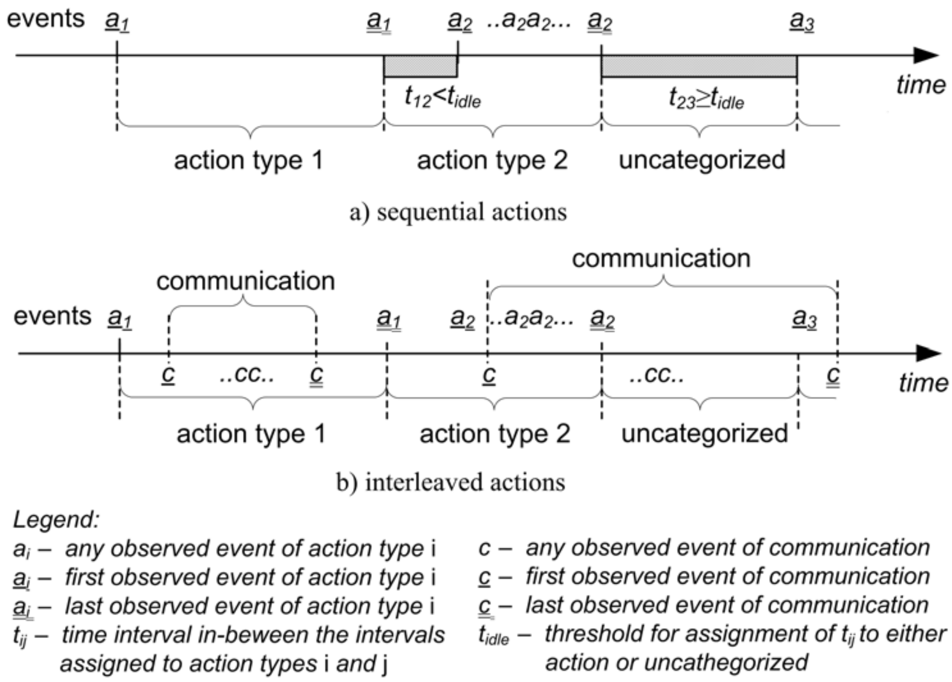


Figure 5.3: Determining the duration of a specific action within a session

Communication differs from the other categories as it happens concurrently with other actions and it is thus monitored in parallel with all other actions, by collecting outgoing messages. If the time between sending two messages is shorter than 5 minutes, it is added to the ongoing communicating session, and if it is longer, it is discarded as the given communication session is considered to end when no messages are sent for 5 minutes. Figure 5.3 illustrates how the duration of a specific action category within a session is determined. Part a) shows sequential actions. The following example illustrates the relationship between events in WoW API and action categories. Players enter the instance ($\underline{a_1}$) of dungeons category (action type 1 in the figure). After defeating the opponents and exiting the instance ($\underline{\underline{a_1}}$), players go into town to perform some trading actions (action type 2). They repair equipment ($\underline{a_2}$), sell unneeded items (a_2), and store the needed ones in the bank ($\underline{\underline{a_2}}$). The time period between exiting the instance and repairing is shorter than the timeout period, so it is assigned to the latest action category. After storing the items in the bank, the player rests for a while. The time between him starting the next action ($\underline{a_3}$) turns out to be longer than the timeout period and it is thus labelled as uncategorised. Part b) shows communication as an interleaved action. For example, while in a dungeon [$\underline{a_1}$–$\underline{\underline{a_1}}$], the players encounter a challenging NPC opponent and are discussing the

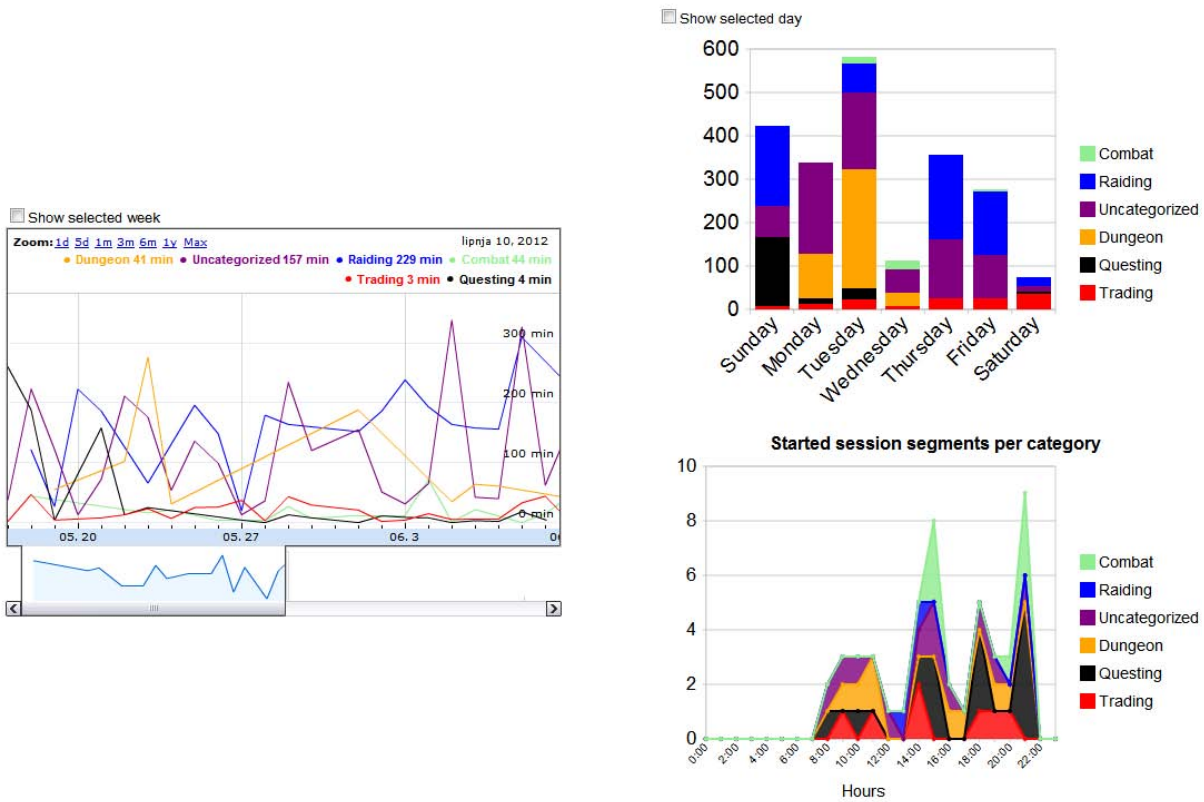


Figure 5.4: Tool for data filtering

tactics to defeat it [c–c]. This time period is assigned to communication while at the same time being a part of dungeons. Later, while in town, the players start to chat and that initiates the second communication session (another communication period in parallel to action type 2 and beyond).

After the logs were parsed, the data about each session segment, comprising of its start time, end time, player designation, type, and session number were stored into a MySQL database. For the purpose of visualization of the gathered data, an Ajax based web tool [145] was developed, which displays the data characteristics. This tool was used for visual data filtering, for easier identification of any anomalous behaviour and outliers in the data. Figure 5.4 shows the screenshot of one set of graphs showing player's session during a week, during days in the week and during one day.

5.3 Session characteristics and user behaviour patterns

The most important session characteristics that we study are session length, session segment length, probability of session segments, number of active users. Also, we look at daily and weekly patterns for listed parameters.

5.3.1 Session length

Lengths of MMORPG sessions have been studied thoroughly in the related work. The issue with research regarding session lengths is that significantly different results are reported even in the case of the same game (e.g., WoW). These differences stem from two main causes: 1) lack of a firm commonly agreed session definition, and 2) different measuring techniques.

As a first step in measuring session length we want to answer a question: "What is a session?". We identify two types of sessions in MMORPGs. Firstly, we identify *character session* as a duration of presence of specific character in the virtual world, and secondly we define *player session* as the time period between the player's login and the player's logout from the game. During one player session a player subsequently uses multiple characters. In short, one player session typically comprises multiple character sessions.

Most frequently used method of measuring session length is the polling technique. This technique is based on the ability of one character in the virtual world to obtain information about all other online characters. In WoW this is possible through the command "/who" which returns the list of active players. This enables creation of automatic scripts which run within the client of one player and gather data about all other active players. This technique measures the length of character sessions and it is applied in a number of works [15, 57, 54]. Additionally, this type of measurement approach can introduce an error, as sampling a player base at 10 minute intervals results in the shortest session time of 10 minutes.

The second approach is estimation of session duration from network traffic measurements [65, 59]. Traffic measurements can extract the length of the whole TCP connection for the player, resulting in obtaining player session lengths.

We use a third approach, in which the measurement process is taking place on the players' computers. This approach has an upper hand in measuring player behaviour as it can be studied in great detail. The downside is that it is very hard to obtain the players who are willing to participate in the research, as players actually need to perform some actions on their computers in order to participate. This results in a lower number of players observed. Our first set of measurements [8] was processed in a way that extracted character session lengths. The acquired data consists of 2753 individual character sessions. The mean session duration value is 54.92 minutes. Interestingly, about 24% of the sessions lasted 2 minutes or less, while the longest session lasted almost 10 hours (598 minutes). A high percentage of short sessions can be explained by phenomenon of “alts” (alt[ernative]s, or “secondary” characters) and their usage. Many players have one or more alts and use them as extensions of the main character’s capacity to store game items. Alts are also used for additional professions, due to the inherent limitation of the game which allows one character to learn at most two main professions (e.g., the main character can be an alchemist/herbalist, while the alt can be a leatherworker/skinner).

Our second set of measurements [60, 146] was processed in a way which extracted player session lengths. As mentioned before we define the (player) session length as the time passed between player login and his logout from the game. If the time between a given player’s logout and a new login is less than 5 minutes, we treat those two subsequent sessions as one session. This is done to alleviate the issue of disconnecting and alts, as in order to switch between characters the player needs to logout and login again. We want to look at the sessions of an individual player (person), not his/her specific character. We realized that if we would look at only the sessions of the specific character, it would lead to misinterpreting the session lengths by considering them shorter than they actually are. Namely, one player may have one or more alts which can be played regularly, or very often, used for storing and trading virtual items. Sessions on those types of alts are commonly very short, usually comprising of buying, selling, and sending the virtual items. Such sessions sometimes last only for a few minutes, and if taken separately they would severely skew the dataset towards shorter sessions. That is why only player sessions can correctly describe the time that players spend in virtual worlds of MMORPGs. In this way, we identified 5872 player sessions, which comprised 11775 character sessions.

We illustrate how significant can the difference resulting from different measurement techniques and different definition of the session be by using Figure 5.5. It should be taken into

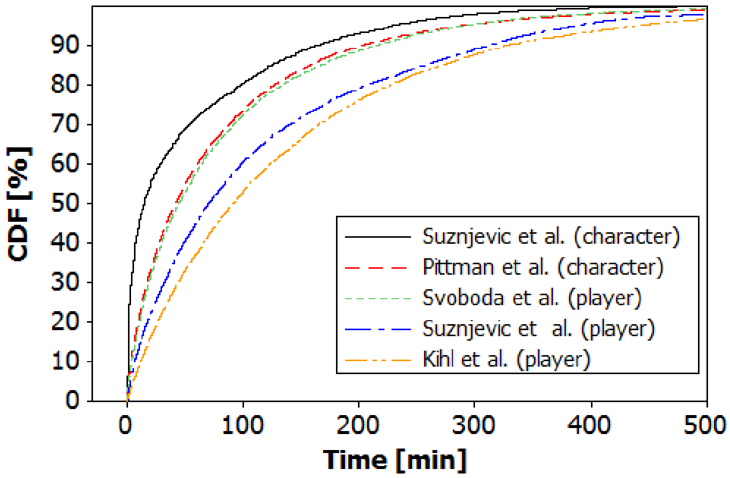


Figure 5.5: CDFs of session length from different measurement approaches

account that all of the measurements depicted refer to the same game (i.e., WoW). The shortest session duration is reported in our first work [8] as measurements had 1 minute precision and were character sessions (30% of the sessions reported were shorter than 5 minutes). It should be noted that the player sample on which measurement was performed was very small. Pitman measured character sessions [55] with a 15 minute sample time, while Svoboda [59] modeled player sessions based on a traffic trace of a mobile network which have shorter session times compared to wired networks. Longer session times have been reported in our work describing the set 2 of our behavioral measurements [60], as we measured player session with 1 second measurement precision and additional parsing to ignore disconnecting. The longest sessions have been reported in the work by Kihl et al. [65] in which authors report player session length based on the results of traffic measurements in the wired accsess network. It is clear that character based estimates of session duration are significantly shorter than the real session duration.

In Figure 5.6, the interval plot of the session duration across the hours of the day is depicted. The graph shows the mean values with the 95% cofidence intervals (i.e., meaning that there is a 95% chance that the true mean of the data is within those values). We measure the length of the session started in the given hour, regardless of whether it spanned several hours. As noted by Williams et al. [49], the average age of the MMORPG player is 33 so most of the players follow a daily routine of spending time at work or in school during the day, which results in their availability in the late afternoon/evening.

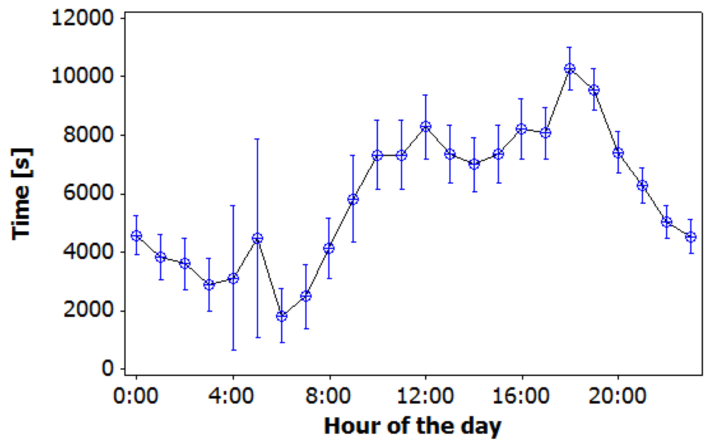


Figure 5.6: Interval plot of session lengths across hours of the day

5.3.2 Action category session segment probability

Probabilities of a specific action category segment appearing during the day are presented in Figure 5.7. The obvious trend is that group based activities (raiding and dungeons) have a raising trend from morning towards the evening, while questing as a main single player activity has a strong decreasing trend. Trading and PvP combat have more or less the same percentage of segments during the day. While probability of the trading segments is easily explicable, as trading interleaves with most of the other action categories and is always and easily accessible to players, the question remains why does PvP combat, as a group based action category have a stationary trend. The answer lays in WoW battlegrounds system. Players enlist for the battle and the game system automatically creates a player group consisting of random players which might not even use the same server. In this way battles are always accessible, and players do not have to organize themselves as they have to for raiding. This shows how sensitive player behaviour patterns are to game mechanics. It is expected of these patterns to change over time as the game designers expand the game with new mechanics. For example, a similar system to battlegrounds was introduced recently for the dungeons category (after our measurements were performed). We expect that due to this change, the dungeons will show the trend similar to that of PvP combat in the future. Another change has been implemented for PvP combat – rated battlegrounds in which organized (i.e., not randomly assembled) groups of players compete in order to achieve higher ranks on the player scoreboards. We assume that this change will result

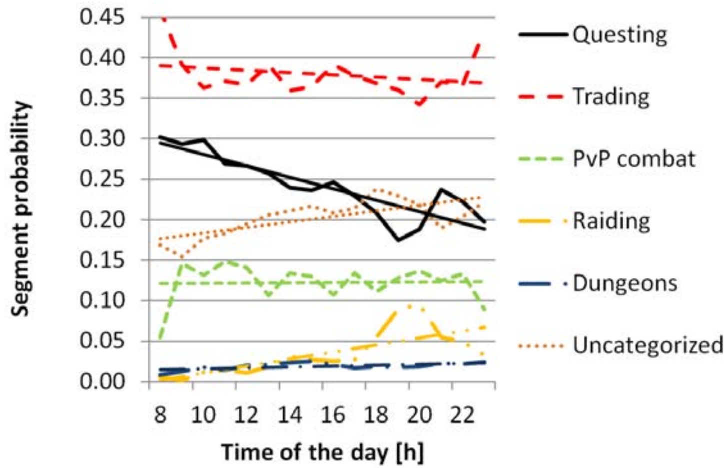
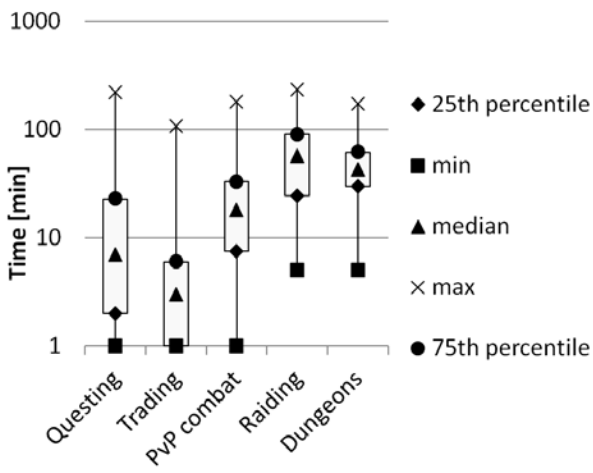


Figure 5.7: Probability of specific action category session segments though hours of the day

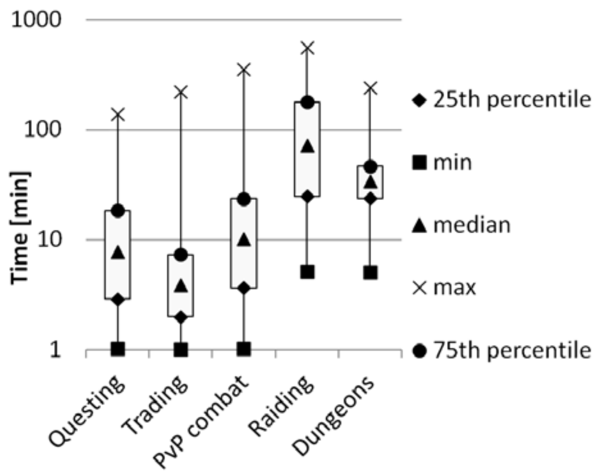
in PvP combat taking a trend similar to other group based categories. Uncategorised segments also have a raising trend towards the evening, which we assume is a result of more players being online and available for just chatting, as well as waiting for organized group activities.

5.3.3 Session segment length

Session segment is a part of the session belonging to only one action category. In our effort to better determine the time characteristics of a specific action category and decrease the effect of interleaving actions (e.g., handing in quests that yield honour points which triggers events belonging to questing *(Quest_Finished*) and PvP combat *(Honor_gained)*), we have added some limitations to the minimum duration for session segments of specific categories. Minimal values for dungeons and raiding were set to 5 minutes, as it is almost impossible to reach and defeat main NPC enemies in those instances in less time. Limits on other actions were set to one minute. Comparison of the 1st [8] and 2nd set of measurements [60, 146] is depicted in Figure 5.8. Dungeons tend to be shorter, probably because the game content is not new any more in the 2nd set of measurements, and players are generally more powerful (i.e., players have better equipment), so they can complete tasks in less time. On the other hand, raiding has longer duration. As players hit the level cap they can only improve through obtaining better equipment, first from dungeons and after that only through raiding, which results in longer raiding sessions. Also, game designers tend to add more raiding instances which are increas-



(a)



(b)

Figure 5.8: Statistics of session segment lengths for first and second set of measurements

ing in difficulty during the life of the game. The rest of the activities have similar or slightly shorter times. The same relations amongst categories remain with raiding as the longest activity, followed by dungeons and PvP combat.

5.3.4 Behaviour patterns

In this section we answer the question *when* players perform certain actions by studying patterns of their behaviour. Recorded play time with portions of each category across all players participating in the 1st set of measurements is shown in Figure 5.9. While there are evident similarities between some players, there are also significant differences (e.g., some have not engaged in PvP combat at all (3, 6, 7), while some others did very few dungeons (1, 7, 9)). Also, the time spent playing in the monitored period varies significantly amongst players. The differences of the player motivations and their connection to session behaviour in terms of action categories performed are studied in more detail in Section 5.4.

In the first set of measurements, we observed that some parts of the session were lacking any categorizable events, so we could not assign that time to any action category. Due to that, for the second measurement set we wanted to determine how much of the overall time played can be categorized into the defined action categories. As shown in Figure 5.10, we have successfully categorized more than 70% of the session time in the second measurement set. The results show

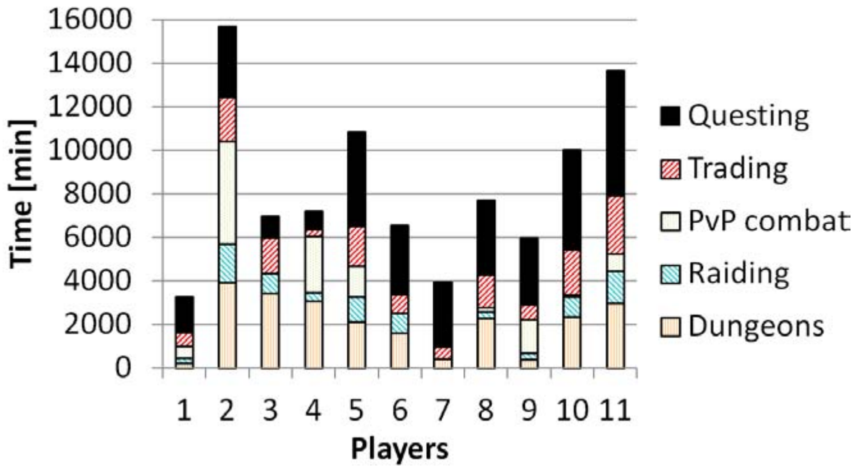


Figure 5.9: Percentage of time spent playing per category (1st set of measurements)

that players have spent most of their playtime in category raiding. We have also investigated communication in our 2nd set of measurements and it is shown that players are comunicating a significant portion of the overal time spent playing (over 17%).

We assume that time labelled uncategorised is spent chatting, AFK, preparing to set on some other actions, travelling, and exploring. In order to confirm these assumptions we have performed a third set of measurements which in addition to the player actions also noted their location. It was determined that players spent 73% of the uncategorized time in capital cities (e.g., in WoW that are “Stormwind”, “Orgrimmar”, “Dalaran” etc.). We assume this is due to the fact that the capital cities are sanctuaries with guards, so no hostile NPCs or players can attack an AFK player in the capital city. As uncategorised portions of the players’ behaviour are a significant part of overall played time, in order to create a good source based traffic model a traffic model of uncategorised is needed. The approach taken with other action categories is not applicable, as we can not give specific instructions to players on what to measure, since we are not entirely sure what players do during these parts of the session. Due to these limitations, we decided to use the traffic model of trading based on the results of these behavioural measurements. We chose this approach because trading is an action category with the lowest mobility and dynamics and uncategorised comprises time periods in which users are commonly AFK. Also, regarding the server side traffic, trading fits the best, due to the virtual world collocation of the players in these two action types (i.e., both trading and uncategorised are usually performed in capital cities).

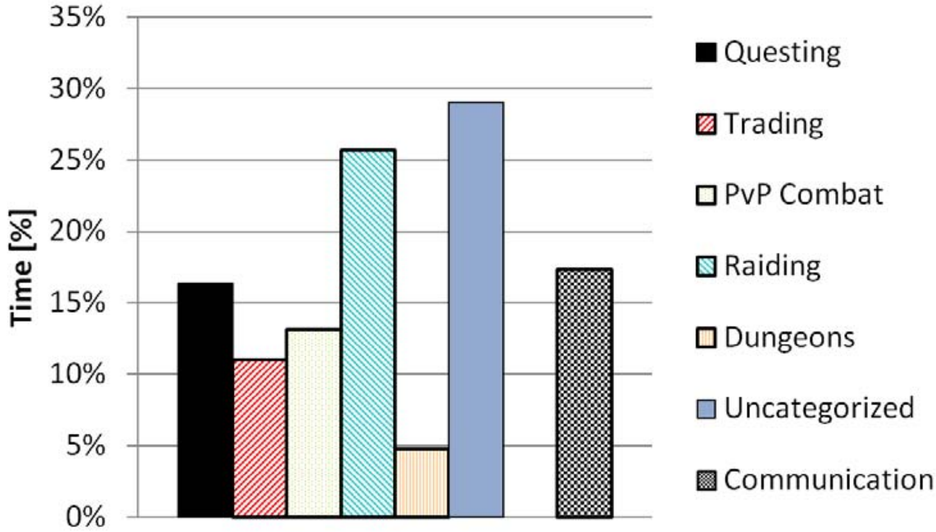


Figure 5.10: Percentage of time spent playing per category (2nd set of measurements)

As for the behaviour patterns, we try to observe hourly, and daily patterns, as well as to look at the overall behaviour of players in terms of amount of time played per category during the whole measurement period. Raiding shows the strongest patterns as it requires a highest number of specifically designated and organized people and is usually done by guilds. Also, raiding needs a group of 10 or 25 members (even 40 in original WoW without expansions) so that is how many people in the guild need to be present in order to start the raid. As opposed to PvP combat, in which large groups of players are entering battlegrounds with the help of the game mechanics, in raiding, players need to organize themselves to play. As shown in

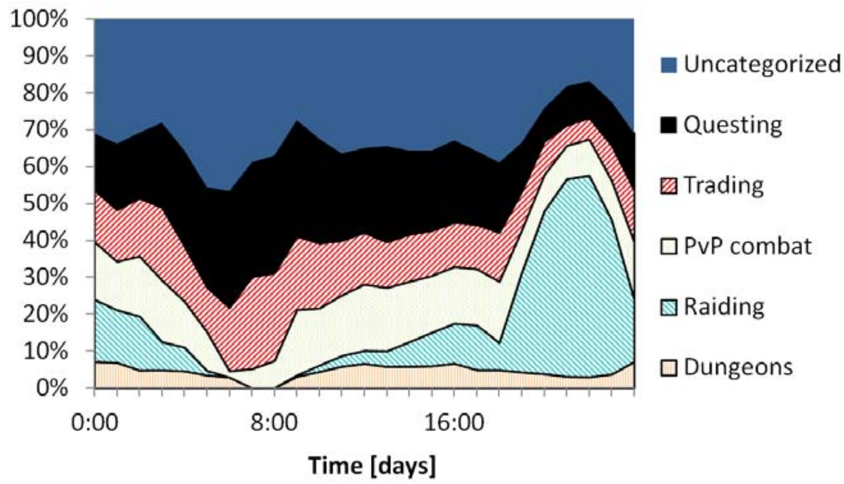


Figure 5.11: Proportion of action categories through hours of the day (2nd measurements)

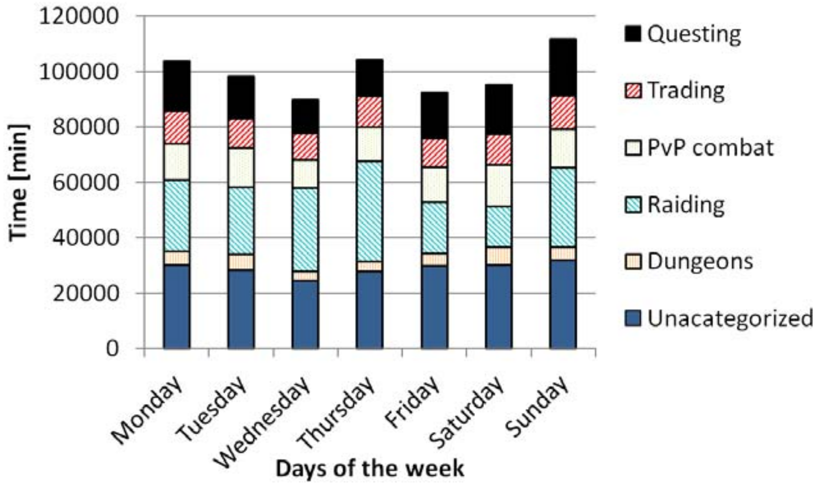


Figure 5.12: Proportion of action categories through days of the week (2nd measurements)

Figure 5.11, raiding shows a significant incline around 18:00 and starts to decline around 22:00, corresponding to the availability of players. This could be expected since the average player's age in our sample is 24, so it may be presumed that most of them work or attend classes during the day. The observed pattern has significant impact on the overall computational and network load, as raiding is the category with the highest requirements on both game servers and the network, as shown in the previous section.

While many previous works show that weekends have higher amount of time spent playing, we observed higher playtime only on Sunday, and not on Saturday (Figure 5.12). We assume that this is caused by a specific weekly raiding pattern. Thursdays have the highest raiding activity, which has lower values on Fridays and Saturdays. Most of the raiding guilds do not raid on Friday and Saturday which can be explained by the fact that Friday and Saturday nights are most often used for socializing in real-life, and especially by younger people. As our player sample is younger than the average player base, this phenomenon is even more emphasized. The rest of the activities are relatively evenly divided across the days of the week. The daily pattern over the week, other than the significant decrease of raiding on Fridays and Saturdays, is not strong.

Adding new content to the virtual world has significant impact on the course of player behaviour. This is easily demonstrated by comparing overall patterns of our first and second measurement in Figure 5.13. Figure 5.13a illustrates how much time players spent performing

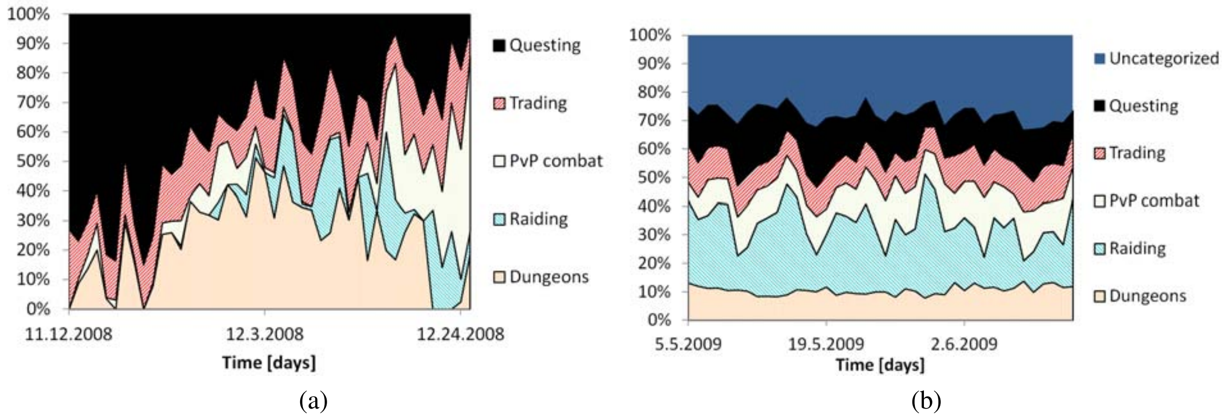


Figure 5.13: Overall proportion of action categories through the whole time of the measurem-nets

a particular action type, or, in other words, how "popular" a particular action type is in our first set of measurements. In this particular trace, the most popular activity (by far!) is questing. This, however, is an atypical case, for a simple reason: our measurements were taken over the six weeks in November - December 2008 following the release of the new expansion for WoW (WOTLK, on 13th November 2008). With the players eager to reach the required level for "high end" game play participation (raiding, dungeons, and rated PvP combat), questing provided them with the most efficient means to achieve it. In this trace, dungeons are the second most popular activity. It is interesting to observe how the share of questing in overall session time decreases over the time period when the measurements were taken, as shown in Fig. 15. It may be noted that, as time passes and more players reach the maximum level, the share of questing is being reduced. Further on, there is a huge increase in PvP combat activity around the 16th December, the date when another battle arena (Arena season 5) started. These results indicate a high interdependence of action type "popularity" and in-game conditions.

On the other side, in Figure 5.13b the whole period of second measurement is depicted. As previously mentioned the second measurement set was during a "stationary" period as related to the game release, where by stationary we mean a time period in which the players have, in general, become familiar with the current game content. As the new content is released every several months, most of the MMORPG lifespan is spent in the stationary mode. Figure 5.13b shows us that, other than weekly raiding pattern, other action categories have a rather stable percentage of game time.

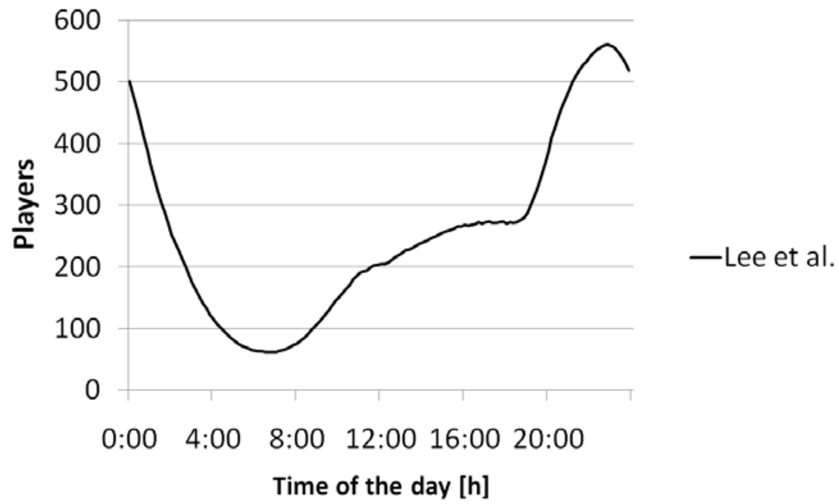


Figure 5.14: Number of active players across hours of the day

5.3.5 Player count

The virtual world of WoW is divided on many shards and the players are distributed across those shards. One shard comprising the entire virtual world of WoW is often referred to as just "server". We will study a number of active players on one such server. The number of active players on a WoW server has been studied in several related works [56], [58], [54], [55]. We do not repeat the measurement process of the player population during the course of the day, as this subject is well covered in the literature and there are publicly available datasets with information about the number of active players on different servers [54, 15].

In Figure 5.14 we can see the obvious hourly pattern of the number of active users through the hours of a day extracted from the dataset by Lee et al. [15]. There are two times as many active players in the evening then through the rest of the day, and if we compare the maximum value in the evening and the minimum value in the morning the difference is almost 7 times.

Besides an obvious hourly pattern of the number of users there is also a significant daily pattern as depicted in Figure 5.15. We can observe two groups, weekdays and weekend. Weekdays have a very similar number of users playing, on all days except Thursdays (weekly maintenance day on the monitored server). Maintenance means that virtual world is off-line, which means there are no active players in that period. Saturdays and Sundays have significantly more players, especially during the day, which again follows as a result of higher availability of an average player during the weekend.

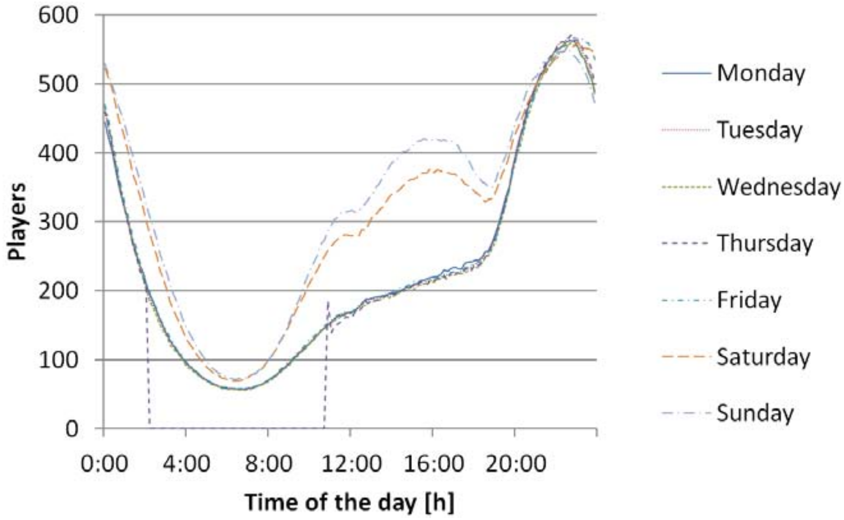


Figure 5.15: Number of active players across days of the week

5.4 User motivation – behaviour relationship

In this section, we provide the answer to the question *why* players do what they do in the virtual world. This is performed through inspecting relationship between previously defined motivational components and subcomponents defined by Yee [39] and player behaviour in terms of our action categories. Definitions of motivational components and subcomponents serve as a basis for our hypotheses about relations between player motivation and their actual in-game behaviour. The model of player motivations consists of 3 major motivational components which comprise a total of 10 subcomponents: *Achievement* (*Advancement, Mechanics, and Competition*), *Social* (*Socializing, Relationship, Teamwork*), and *Immersion* (*Discovery, Role-Playing, Customization, Escapism*), as described in detail in Section 3.1. The goal of this effort is to model the types of players based on the previously established psychological motivators. Based on a player type, behaviour of a single player can be modelled more precisely. Also, the mapping between psychological motivators established by a third party and action categories serves as another form of validation of the proposed action categories.

5.4.1 Motivation–behaviour hypotheses

As raiding and PvP combat are two action categories which yield the most in terms of character improvement at the maximum level, we assume that the players highly motivated by advancement will spend a greater (than average) amount of time performing them. Questing is the prime activity for obtaining experience, and therefore the levelling process, and it is positively associated with this motivational subcomponent in the early stages of the game in which the players have not reached the level limit. We do not assume such as association as the time when the measurements took place, since by then most of the players have already reached the level limit.

PvP combat is a highly complex activity. To perform well, players need to have a deep knowledge of not only the functions of their own character, but the functionalities of other players, and also to anticipate their moves. Therefore, we assume that it will be positively associated with the knowledge about mechanics of the game. Also, the highest competition is in the PvP combat action category, as players can actually be ranked in comparison with other players and this information is visible in the game, as well as through the player rankings on the game's web site, so we assume a positive dependency.

Socializing and relationship are motivational components which can be hardly mapped to our action categories. We assume that these kind of human interactions are mostly realized through communication between players. In this way, we foresee that amount of communication between players will be increased for those who score highly on these two motivational components. Teamwork is connected to a group effort, therefore the players more motivated by this subcomponent should perform more group based actions such as dungeons and raiding. As for the PvP combat category, it is difficult to assume the latter, as players often perform PvP combat "alone in the group". They join PvP battles through the help of the game mechanics, and often do not actually try to perform as a part of a group or in cooperation with others (e.g., a typical player entering a battleground in WoW is assigned to a random group of players).

Exploration in WoW is mostly done through questing, as some tasks encourage players to go to areas of the world they have not yet explored. Role-playing motivational subcomponent is more focused on role-playing in terms of acting out characters in unscripted situations such as in an improvisational theatre (e.g., player only speaks from the point of his/her character). This kind of role-playing is supported on dedicated role-playing (RP) servers. Customization and escapism cannot be mapped on our action categories, as players can escape their real-life problems by doing any of the defined actions, and customization is rather limited in WoW and it has not been taken into account. On the basis of given descriptions, the following nine hypotheses were formed:

- *H1. Advancement is positively associated with Raiding*
- *H2. Advancement is positively associated with PvP combat*
- *H3. Mechanics is positively associated with PvP combat*
- *H4. Competition is positively associated with PvP combat*
- *H5. Socializing is positively associated with Communication*
- *H6. Relationship is positively associated with Communication*
- *H7. Teamwork is positively associated with Dungeons*
- *H8. Teamwork is positively associated with Raiding*
- *H9. Discovery is positively associated with Questing*

5.4.2 Validation of motivation – behaviour hypotheses

Validation is performed on our second measurement set. By tracking text based communication we have recorded 456228 messages. Message sending is, in general, very bursty, as more than 50% of the messages have the time period between sending two subsequent messages of 20 seconds or less. According to the player survey results, voice communication is used by more than 94% of the participating players. Players were also asked which program they use for voice communication (multiple answers were allowed). The most popular program in the observed player group is Ventrilo (75%), followed by Team Speak (21%). Approximately 11% of players use Skype. Also, it is interesting to note that only 16% of the players use the VoIP client inbuilt into WoW, the Voice Chat. This indicates that the Blizzard's VoIP solution is not well accepted among the players, although it has been deployed in late September of 2007, almost a year and a half before our measurements were taken in May/June 2009. One of the disadvantages of the inbuilt voice chat is that it is disabled when a player exits the WoW client,

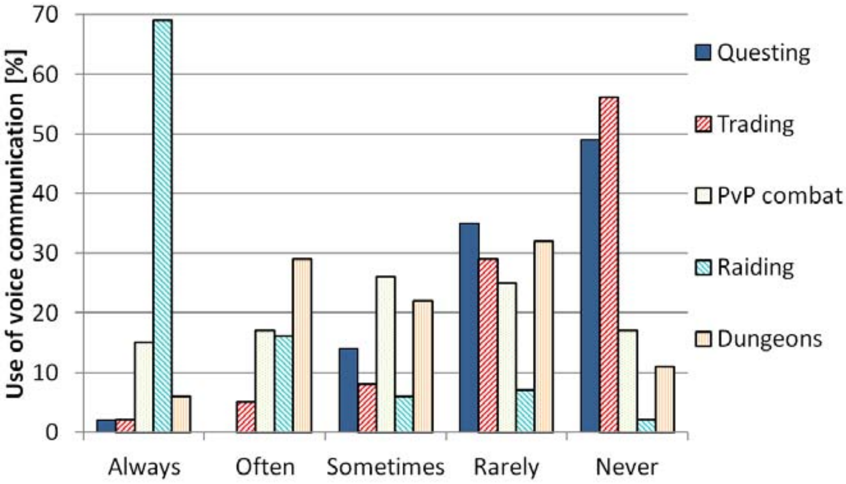


Figure 5.16: Usage of voice communication per action category

so if a player is disconnected from the game server during combat, the voice communication is interrupted as well. For players, it is often useful to maintain communication even when they are disconnected from the game server. Hence, most players tend to use independent programs, even at an additional cost. In our survey, the players were asked how often they use voice communication depending on the action category. The reported frequency of use of voice communication across action categories is shown in Figure 5.16. We can see that almost 70% of players "always" use voice communication while raiding, followed by PvP combat with almost 15%. Many raiding guilds have voice communication as mandatory, which may be one of the reasons for such high use of it while raiding. Players that participate in highly organized PvP combat (i.e., arena combat, or premade groups for battlegrounds) also often use voice communication for better coordination, which explains why PvP combat has second highest voice communication usage. Questing and trading are simple and not group-based activities, which explains the fact that almost half of players "never" use voice communication in these situations.

In Figure 5.17, the statistical distribution of motivational subcomponents derived from the survey is shown.

It is worth noticing that subcomponents of achievement are the strongest motivators. Only teamwork out of all other subcomponents, is close to advancement, mechanics, and competition. Subcomponents of immersion are the least significant in our player sample. The age of the game

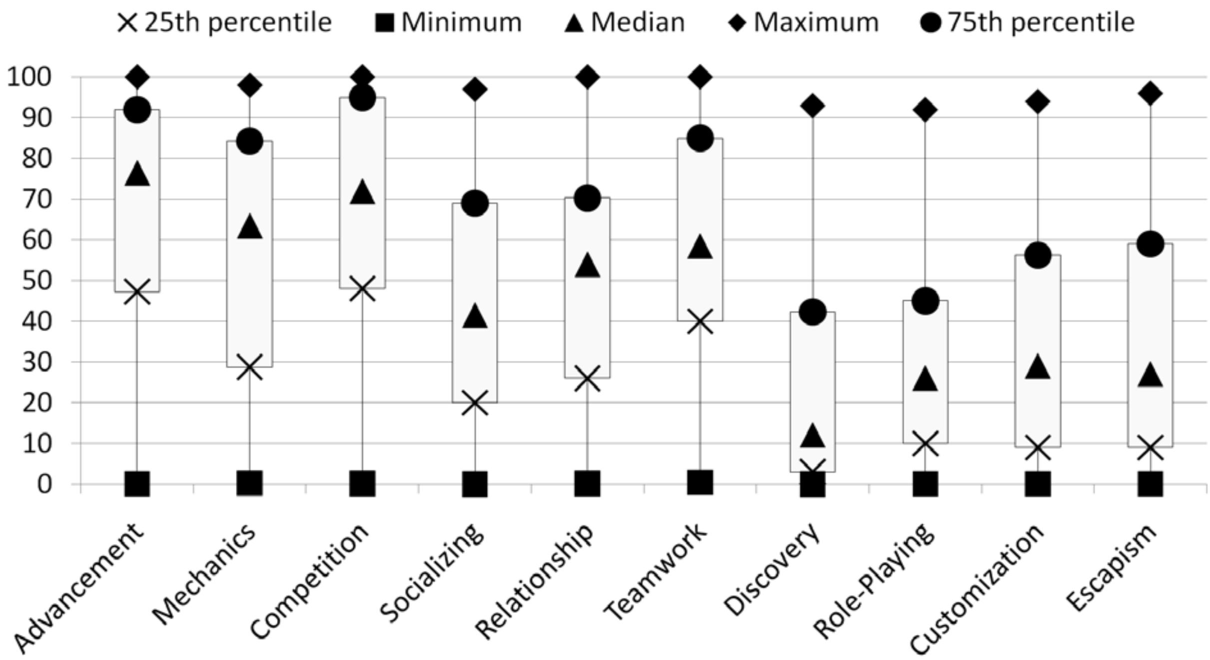


Figure 5.17: Statistics of the motivational subcomponents

might have an influence here, as the game was, at the time when the measurements were taken, almost four and a half years old, and most of the players were quite familiar with it (we can see the discovery subcomponent is the lowest of all). Low amount of possible customizations may be the reason for low score of the customizing subcomponent. In our sample, players mostly played on PvP servers (92), then on Normal servers (23), and only 4 of the players played on RP servers, which may account for low ranks on role-playing motivational subcomponent. Values shown in Figure 5.9 are representing average percentage of time spent in particular category which are referenced when the player behaviour is tested (e.g., while testing H1 we first separate the players who scored over 80 in advancement and then compare their time spent in raiding category with the reference value which is 26%, also we separate the players who scored below 20 in advancement, and test whether they spent less time than the reference value in raiding category). Results of the data analysis across all hypotheses are shown in Table 5.2. In first column, the percentage of players who conformed with the hypothesis (only players with extreme rank score in specific motivational component (i.e., over 80 and below 20) were tested). The second column contains Pearson's r coefficient, and the third column contains Kendall's $\tau - b$ coefficient derived from the whole player sample. Strength of the association is

Hypothesis	Conforming players	Pearson's r	Kendall's $\tau - b$	Association strength
H1	61%	0.1079	0.0610	Weakly positive
H2	37%	-0.0335	-0.0008	Weakly negative
H3	38%	-0.1029	-0.0839	Weakly negative
H4	42%	0.2084	0.1631	Inconclusive
H5	56%	0.1315	0.0951	Weakly positive
H6	68%	0.1675	0.1160	Positive
H7	17%	-1617	-0.1028	Strongly negative
H8	77%	0.1684	0.1072	Positive
H9	62%	0.1231	0.0995	Weakly positive

Table 5.2: Statistical data about player's conforming to hypotheses

determined based on these parameters. Most of the presented correlation coefficients are rather weak, but confirm the trends shown by the percentages.

As it can be observed from Table 5.2, both hypotheses H1 and H2 are not strongly supported by the data. We tried to determine the relationship between players highly motivated by advancement who have increased time spent in raiding and those who have increased time in PvP combat. It has been shown that those two player pools are very distinct, meaning that players are either focused on "Player versus Environment" or "Player versus Player" aspect of the game. As high as 89% of players who have high advancement component have increased playtime in either raiding or in PvP combat. The hypotheses H3 approved to be false as it is proved that increased knowledge of the game mechanics is not positively connected to the PvP combat category. H4 is only in which the statistics suggest the opposite relationship, while the percentage of the players does not confirm the hypothesis, the correlation coefficient suggests the different. The hypothesis H5 is weakly positive while H6 has a stronger positive association. This means that the socializing component has an influence on the amount of time players spent communicating, but it is not that significant. The amount of time players spend communicating by voice has not been measured, but only the percentage of players who use voice communication. These results showed that 100% of players who had increased socializing subcomponent used voice communication while playing, though the number of players in the sample who used voiced communication was also very high (94%). The explanation to very different results of H7 and H8 can be in the actual teamwork requirements of a small group and a large group activities within WoW. Dungeons represent a content which is used in the process of achieving the maximum level. As players reach the maximum level and make significant progress in the power of their characters in terms of character equipment (e.g., weapons and armour), dungeons tend to become very easy, even trivial. This results in teamwork not being required in dungeons. On the other hand, raiding instances are made very complex to grant the best rewards. We showed that players most often use voice communication while raiding, when the cooperation between them has to be very good. This is proven by the fact that only a portion of players can complete raiding instances. The results have showed that H9 hypothesis has a weakly positive association, with 62% of the players conforming to it.

5.4.3 Individual player modelling

While the data obtained is identifying the trends in the relationship between motivations and player behaviour, we do not perform modelling of the player (in form of explicit transformation function from player motivations to behavioural categories). The modelling of the player needs to take into account other various factors such as player experience in the game or the player "age" in the game, real-life characteristics of the player such as education, job, other psychological factors, etc. Also, the player sample of 104 players is too small for such complex modelling.

Modelling of specific player type is interesting as it player's preferences have significant impact on how the players behaves, and the mix of player types on each shard (or server) of the virtual world can be different. For sharded MMORPGs like WoW, each of the shards is a complete replica of the virtual world. Each of these virtual worlds have their reputation amongst players and attract players with different goals. While a certain number of players chose the shard on which to play randomly, some players will choose a shard which community is focused towards the aspect of the game which is the most interesting for the player, for example PvP combat or raiding. This in return has an influence on both server load, as well as on the characteristics of the generated traffic of different shards. In WoW there is already a division of servers on "PvP", "Normal", and "Role playing" servers based on the player's preferences.

While not numerically defining the new model, we have proven that the differences between players based on their motivations exist. We use the average player in the behaviour simulation, but we acknowledge the differences between specific players by creating player types with different characteristics. These player types can be enabled in order to be used in our simulation of player behaviour (i.e., some players have higher preferences for PvP combat while lesser for dungeons and raiding). The values of those characteristics are not derived from the real data, and are used as "placeholder" values which can be replaced in the simulation once a satisfying model of player motivation–behaviour relationship has been created. Once such models are created differences in traffic between different shards can be simulated based on the types of players which are focused on that shard.

5.5 Behaviour modelling

We perform modelling on two levels: single player and single shard (server). For a single player, we model player session duration (i.e., including multiple characters in one playing session), duration of action specific session segments, and probability of switching between segments (e.g., what is the chance that after questing a player will perform a trading action). Models for the single player are created through the hours of the day in order to capture dominant hourly pattern. As the daily pattern is not as strong, the only parameter modified by the coefficients for different days are lengths of session segments.

For the single server, we model the number of active players through initial values obtained from the dataset and modify those by arrival and departure processes. It should be noted that the dataset contains information regarding characters not players, so these arrival and departure processes are based on characters which results with a character session length, not player session length, in the simulation on the level of one server. The number of players is modelled for both hours of the day and days of the week in order to capture hourly and daily trends.

In order to identify the underlying distribution of an empirical dataset, we used the tool Minitab (www.minitab.com) which has both the least squares and maximum likelihood (MLE) method for estimation of parameters of the distribution. We used both methods, in order to confirm the results, and the results displayed here are obtained by using MLE.

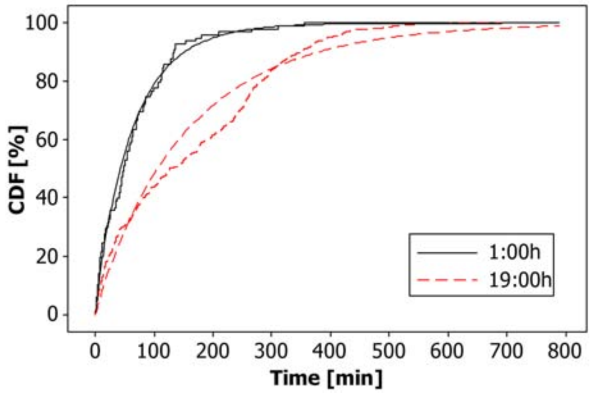


Figure 5.18: CDFs with Weibull fits of the session length started at specific hour (19:00-20:00 and 1:00-2:00)

5.5.1 Session Length

Previous works [55, 59], identified that session lengths conform to Weibull distribution, and we confirm those results as Weibull distribution proves to be the best fit for the session length. Sessions started in the evening, however do not conform as well to the Weibull distribution as those started later in the night. We also prove that session lengths are heavily dependent on the time of the day. In Figure 5.18, the Cumulative Distribution Functions (CDFs) of the sessions started between 1:00-2:00 and 19:00-20:00 are shown and it can be observed that the sessions started in the evening are much longer than those started early in the morning.

Parameters of the model are listed in Table 5.3. The hour marks the hour in which the session is started (e.g., 0:00 marks the sessions started between 0:00 and 1:00). The models are based on a precision of one second. All values, but the values for the sessions started in the hour 5:00-6:00, are estimated from the dataset. For the period 5:00-6:00, due to the lack of measurements, we estimate the parameters from the data from 4:00-5:00 and 6:00-7:00.

In order to calculate the values of the distribution we use inverse transformation function for the Weibull distribution. For sessions started between 19:00 and 20:00, the expression is as follows:

$$f(x) = 155.7 * (-ln(x)^{(1/0.9414)}) \tag{5.1}$$

where x is uniformly distributed random variable $0 \leq x \leq 1$.

Table 5.3: Model of session length across hours of the day (parameters of Weibull distribution)

Hour	0:00	1:00	2:00	3:00	4:00	5:00	6:00	7:00
Shape	0.97	0.87	1.00	0.95	0.83	0.82	0.81	1.06
Scale	4536.95	3749.16	3484.79	2757.72	2402.32	2066.07	1729.81	2240.09
Hour	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00
Shape	0.81	0.92	0.92	0.88	0.92	0.88	0.89	0.97
Scale	3426.77	5075.91	6627.83	6103.50	7547.86	6502.32	6185.00	6623.49
Hour	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
Shape	0.78	0.80	0.86	0.83	0.87	1.00	0.94	0.90
Scale	7131.57	6942.89	9935.93	9499.96	7153.49	6498.66	5039.38	4303.22

5.5.2 Session segment probability

In order to model the player behaviour throughout the day we construct a first order Markov chain for each hour of the day as we assume that the next action is only dependent on the previous action (e.g., after a dungeon or a raid, it is typical for a player to go sell items he/she does not need, or upgrade the newly obtained ones). In Figure 5.19, transition probabilities between states for hour 19:00-20:00 are depicted. Each action category is modelled as a single state of the Markov chain and it is marked with its starting letter (Q–Questing, P–PvP Combat, R–Raiding, D–Dungeons, U–Uncategorised, and T–Trading). Transitions with probabilities

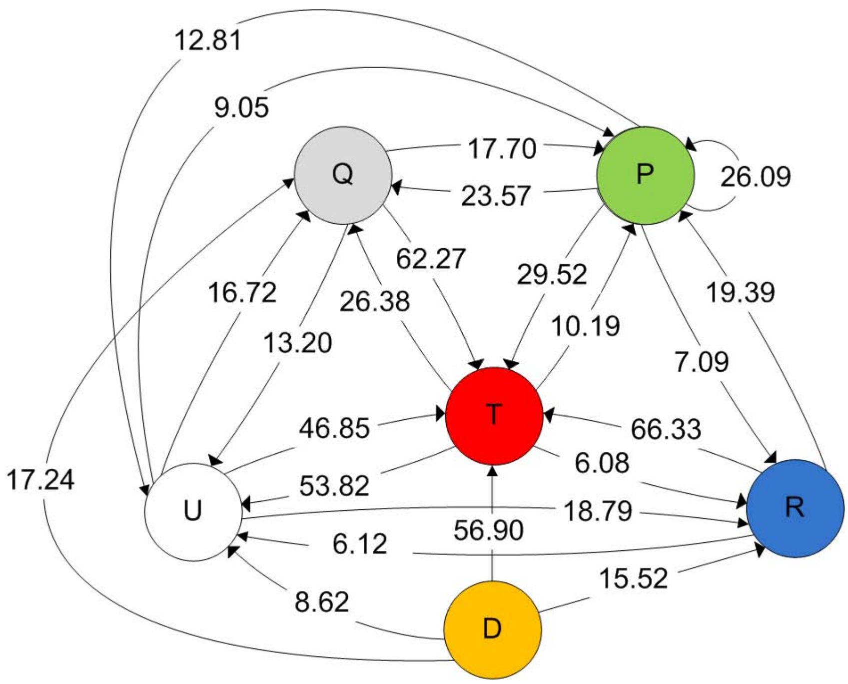
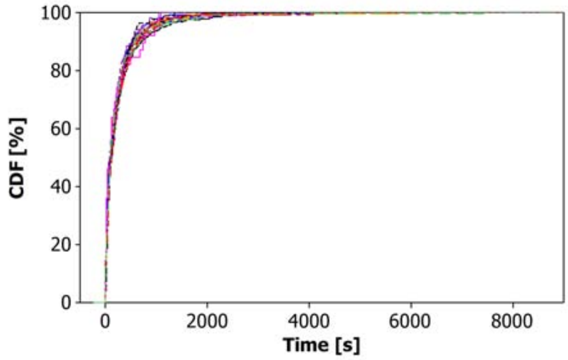
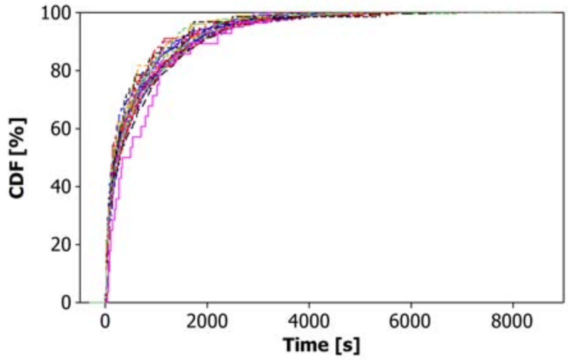


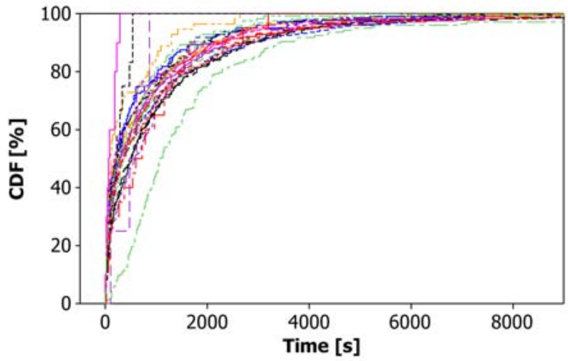
Figure 5.19: Transition probabilities between states in hour 19:00-20:00 (Q–Questing, P–PvP Combat, R–Raiding, D–Dungeons, U–Uncategorised, and T–Trading)



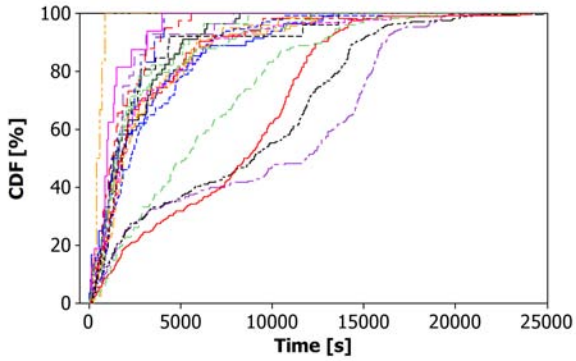
(a) Trading



(b) Questing



(c) PvP Combat



(d) Raiding

Figure 5.20: CDFs of the action specific segments during hours of the day

lower than 2% are not shown in order to simplify the figure. Parameters of the chain across all hours of the day are listed in the Appendix C. The detailed description of the notation of the model through an XML file is given in Section 6.1.2.

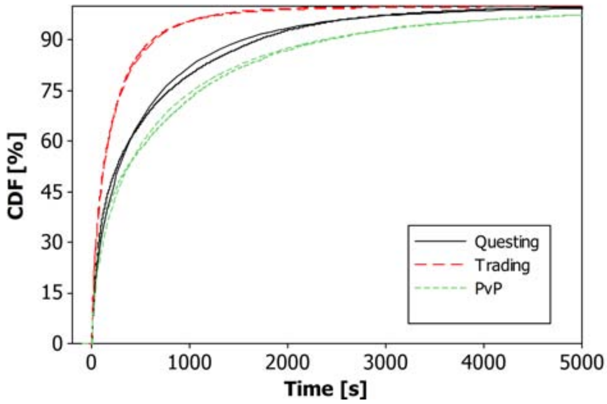
5.5.3 Session segment length

Through inspecting CDFs of hour specific segments durations depicted in Figure 5.20, we conclude that questing and trading segment durations do not significantly dependent of the time of the day (i.e., differences between values of the CDF for different parts of the day were within 10%). PvP Combat has more differentiated CDFs, similar to dungeons, but still the differences amongst hours of the day are rather small (i.e., different values of CDF within 20%). Based on this information we decided to model session segments of those four action categories as

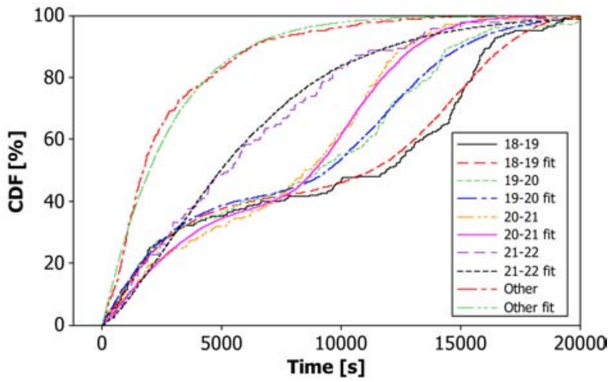
Table 5.4: Session segment duration models

Action category	Fit	Shape	Location	Scale	AD	P-value
Trading	Weibull	0.69	-	190.98	14.70	<0.01
Questing	Weibull	0.66	-	440.01	59.93	<0.01
PvP Com.	Weibull	0.61	-	603.57	28.23	<0.01
Dungeons	Largest Extreme Value	-	1321.42	1116.00	4.80	<0.01
Uncategorised	3 - parameter Weibull	0.79	300.96	567.06	11.87	<0.05

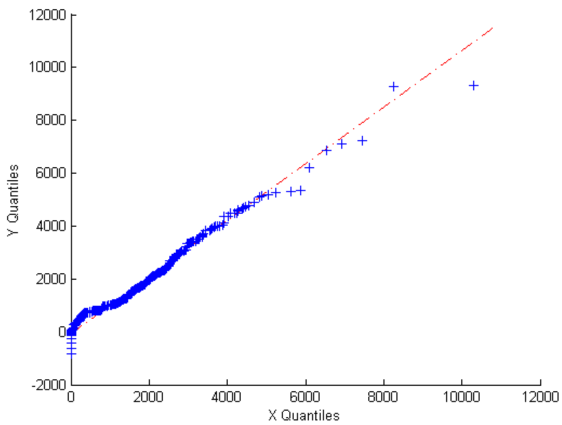
independent of the time of the day.



(a) Questing, Trading, and PvP Combat



(b) Raiding



(c) Dungeons

Figure 5.21: Goodness of fit of session segment length across categories

The segment duration models are provided in Table 5.4 with the values of goodness of fit tests (Anderson Darling statistic and the P-value). The P-values are showing that even the best fits are not closely following the empirical distribution but it should be noted that these tests are biased for large messy datasets [147]. Dungeons were modelled with Largest Extreme Value

Table 5.5: Raiding segment duration models for different hours of the day

Hour	Data	Fit	Location	Shape	Scale	AD	P-value
18-19	42%	Weibull	-	1.175	2513	0.405	>0.250
	58%	Weibull	-	6.110	15306.5	1.722	<0.010
19-20	44%	Weibull	-	1.121	2583.76	0.663	0.085
	56%	Lognormal	9.4173	-	0.227	1.016	0.011
20-21	42%	Weibull	-	1.279	3303.08	1.18	<0.010
	58%	Lognormal	9.287	-	0.19762	0.768	0.045
21-22	100%	Weibull	-	1.368	6476.70	0.625	0.102
22-18	100%	Weibull	-	1.046	2786.5	5.209	<0.010

distribution. This distribution also results in negative values which were transformed into 0. This fit is depicted with a Q-Q plot presented in Figure 5.21c. On the other hand, raiding shows significant differences in the CDF based on the time of the day (Figure 5.20). It is noted that the longest sessions are in the evening, when players can afford long uninterrupted time periods (e.g., some raiding sessions span few hours). We decided to model raiding segments' duration with five different models based on the time, one for the hours 18:00-19:00, 19:00-20:00, 20:00-21:00, 21:00-22:00, and another model for the remainder of the day. Also, those segments proved to be harder to model, so mixture modelling is applied with a limit at 7500 seconds. Models for raiding segment duration are depicted in Table 5.5. As can be seen p-values are higher, than for other action categories, but this is mostly due to the fact that the raiding session segment datasets are smaller because they are split across hours. Goodness of fit for the models of raiding are depicted in Figure 5.21b by plotting the CDF of the empirical data against the CDF of the date derived from the model. As it can be seen, models are closely following the empirical data.

5.5.4 Number of players

In our modelling process, we aim to model only the number of players on a single shard or a single WoW server. The number of active players on several different WoW servers has been studied in several works [56], [58],[15],[54], [55] over very long periods (up to over 1000 days). Several datasets from listed measurements are publicly available, so we do not repeat the measurement process, but use the datasets described [55] and [15]. The following differences between these datasets should be noted: 1) they were taken on two different servers of WoW, 2) measurements have been done in different time frames, and 3) while Pittman's datasets comprises data about both available player factions on the server, Lee's datasets contains information about only one.

We model the player number through the initial value combined with arrival and departure processes for increasing and decreasing the current user number. We model the arrival of new players and departure of leaving players as a *Homogeneous Poisson Process* (HPP). This modelling is based on results of Chen et al. [50] who stated that arrival process can be modelled as HPP for each 1-hour interval. In other words, the same rate for the arrival/departure process is applied during 1 hour intervals leading to 24 rates for each process for one day. While their results are obtained based on an analysis of a different game – ShenZhou Online, the authors stated that the results are applicable for other MMORPGs. Additionally, beside 1-hour intervals, we investigate arrival and departure processes in 10 minute and 15 minute intervals for Lee's and Pittman's dataset respectively, as their sampling frequencies enable this. We wanted to observe whether the lowering of the duration of the time interval, on which constant HPP rate is defined, makes the simulation of player number significantly better. Figure 5.22 shows Lee's and Pittman's results for number of users over a 24-hour cycle, also with our own simulation results. As it can be observed from Figure 5.22, both simulations slightly underestimate the number of users. Also, simulation results with 1 hour intervals are not significantly worse than those with 10 minute and 15 minute intervals. All parameters (i.e., starting number of users, and a 24-extent one-dimensional arrays for arrival/departure rates) are calculated for each day of the week in order to capture both hourly and daily patterns.

This modelling procedure is static, meaning that it is not adaptable to the variable number of users because of its dependence upon arrival and departure rates. In order to enable simulations with a significantly different number of players, we apply a normalization technique in an effort to obtain the "shape" of the curve describing the number of users characteristic for a WoW server over a 24-hour time cycle. We apply normalization methodology for prediction of number of players described in detail in [148]. The algorithm is described as follows. Our training set consists of the number of users over a period of D days and nights (i.e., around 1100). Let v_td be the number of active users at time t on cycle d. For each 24 hour cycle, d,

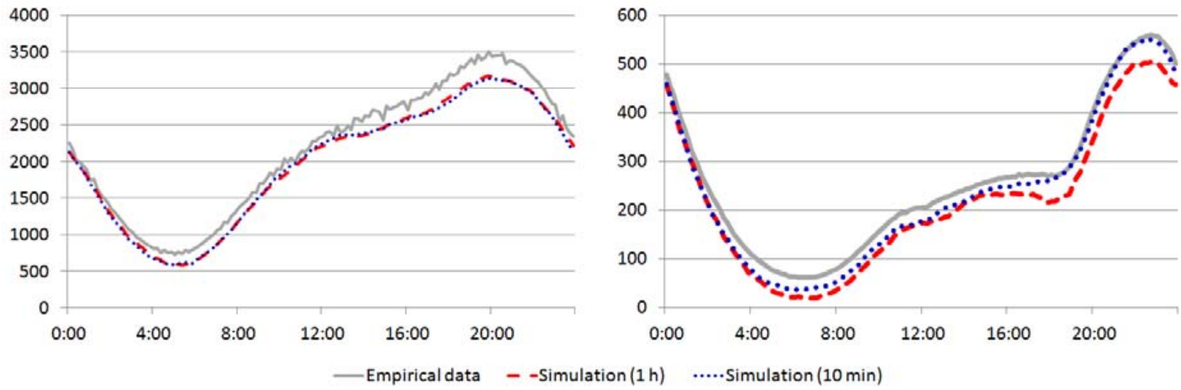


Figure 5.22: Empirical and simulated (HPP) number of users over a 24-hour cycle (left – Pittman's dataset, right – Lee's dataset)

number of users at T time–points is observed, which results in following training set:

$$V_d = [v_{1d}, ..., v_{Td}], d = 1, ..., D. \tag{5.2}$$

We use a slight modification as we normalize the player number with the average number expected in the day in order to be able to start the simulation at any time in the day.

$$V_d^N = \frac{V_d}{avg(V_d)} = \frac{v_{1d}}{avg(V_d)}, ..., \frac{v_{Td}}{avg(V_d)} = [v_{1d}^N, v_{2d}^N, ..., v_{Td}^N] \tag{5.3}$$

For each time–point t we compute the 80 percent quantile.

$$q_t^{80} = 80\%quantile(v_{t1}^n, ..., v_{tD}^n), t = 1, ..., T \tag{5.4}$$

In this way we can obtain the predicted number of users at each time t regardless of the average number of users in the day. Based on the difference between current number of users and predicted number of users, in the next step we calculate parameters of HPP, an arrival rate (if an increase in player number is predicted), or a departure rate (if a decrease in number of players is predicted). This approach allows us to make simulations of a different number of users but still capture the characteristic curve which depicts hourly pattern of the number of users. We find this important, because number of users can differ significantly between games, and as we can see in Figure 5.22, even between different servers of the same game (i.e., depicted datasets are from two different WoW servers). It should be noted though, that Lee's dataset comprises only information about one out of two factions in WoW so those numbers should be doubled.

5.6 Summary and outlook

In this chapter we have presented the measurements and methodology through which behavioural data was gathered. Additional action categories Uncategorised and Communication have been introduced. We identified characteristics of player behaviour and especially noted behavioural patterns. Also, we have explored relationship between players' psychological motivations and their in-game behaviour. Based on the data obtained through measurements a player behaviour model has been created. In the next chapter we use this model to run the traffic generation procedure in order to implement a source model of MMORPG network traffic.

Chapter 6

User behaviour based traffic generator

In our effort to better model the traffic of MMORPGs we are taking a source based approach. Modelling the sources of traffic is, in other words, modelling the network behaviour of the applications running on the endpoints.

Our approach for modelling the sources of network traffic is based on application level user behaviour. Behaviour on the application level is categorised, and application generated traffic of each of the defined categories is modelled. Through this we achieve source-based traffic generation.

In order to realize a complete source-level traffic model, models of different aspects of player behaviour (i.e., number of players, transition from one category to another, duration of action specific session segments, and durations of sessions) need to be integrated with traffic models. In order to achieve this integration, a functional architecture is proposed.

The proposed functional architecture is implemented in User Behaviour Based Network Traffic Generator (UrBBaN-Gen). The goal of UrBBaN-Gen is to generate realistic traffic of a MMORPG, as a complex IP service, on both client and server side and based on user behaviour. Through modification of the service parameters, UrBBaN-Gen allows simple testing of various user behaviour scenarios and their impact on the network. While UrBBaN-Gen has been developed to generate WoW network traffic, functional architecture and implementation are service independent. New services may be added through new user behaviour models

In order to confirm that our service model describes the traffic of WoW better than existing models, we perform several validation procedures with UrBBaN-Gen. First, we compare the generated traffic per action category versus the model parameters and empirical validation traces, at the level of a single flow. Next, the results of our model are compared with existing WoW traffic model [59]. Next, the results of the behavioural simulation are compared with the empirical measurements. Finally, we compare the load per player, obtained from the aggregated traffic based on the simulation of the whole service (i.e., one WoW server), with the results of empirical measurements taken in an access network in Sweden [65].

6.1 Architecture

The proposed functional architecture works on two levels as depicted in Figure 6.1: 1) application level (behaviour simulation), and 2) network level (traffic generation). Also, in terms of scale it works for: 1) single user, and 2) multiple users. At the traffic level, the main function regarding single user is to generate a single flow with certain properties. Providing and controlling resources for generation of high number of traffic flows is the main functionality of the traffic level for multiple user. In order to reduce the hardware costs, for provision of the needed resources we use the virtualization. As the Figure 6.1 suggests, the main functionality of the application level for a single user is behaviour simulation through transition between action categories. Providing accurate number of active users is the main task of the application level for multiple users.

The behaviour of a single user during the course of the MMORPG session can be described as follows: a user will perform actions from one category for some time, followed by actions of another category, and so on until the end of the session. We use a first order Markov chain for modelling the decision process of switching between separate action categories in order to simulate a single user behaviour within one session. We also model player session length and action specific session segment length for the simulation of a single user.

For multiple users, two approaches to simulation are used.

- The first approach uses the definition of an initial number of users (at the start of the day i.e., 0:00), and modifies the user number through arrival and departure processes which are modelled as HPP. It should be noted that the values defined for the starting number of users, arrival and departure processes are based on the datasets of character probing

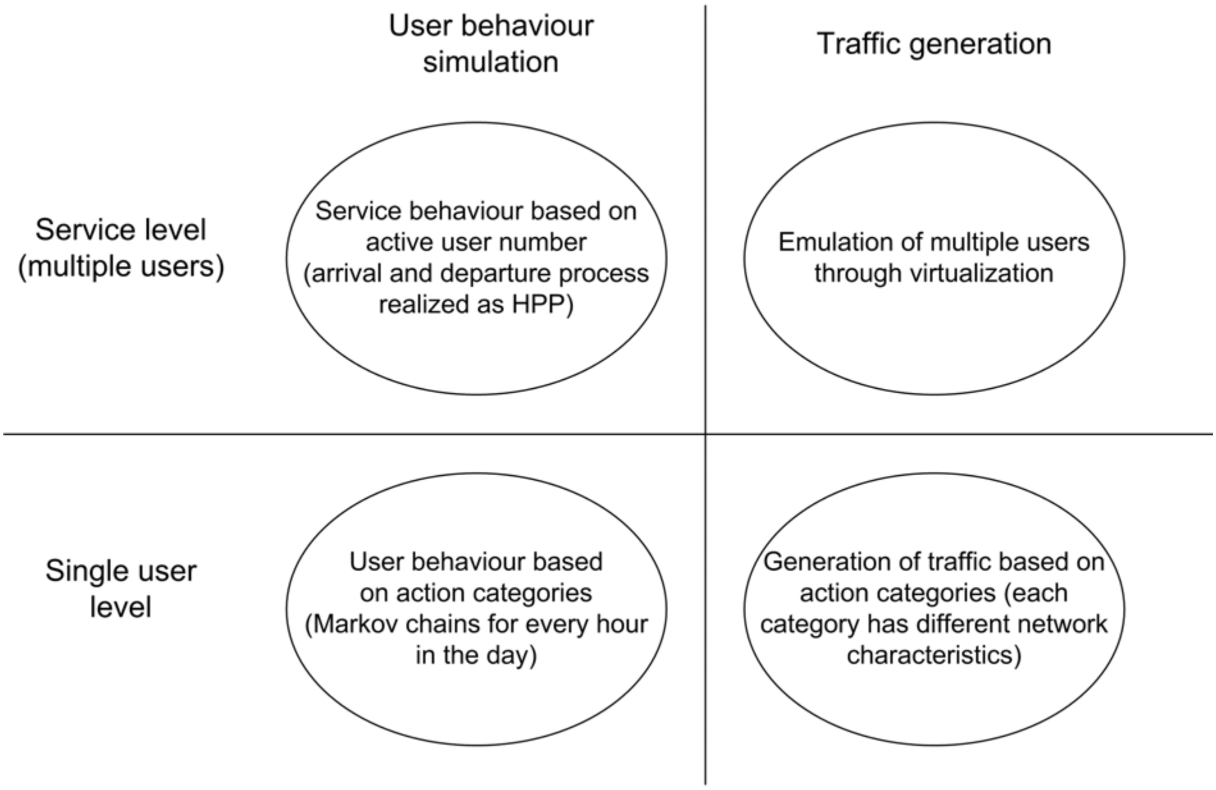


Figure 6.1: Levels of the functional architecture

measurements (as there is no easy way to distinguish which character belongs to which player). Therefore, the session lengths in the service simulation are character based. This is not an issue as the simulation of the service is focused on the load which is preserved regardless of the character/player session lengths.

- The second approach is based on normalizing the patterns of the number of users to a list of coefficients which describes "the shape of the pattern". Through multiplying the coefficients with expected average number of active users we can estimate the number of users at any point in the day. For example, this approach enables us to simulate how the number of active users would behave if the average number of users would double, without having to measure new values of arrival and departure rates.

For the network traffic generation of a single user the traffic models of player action categories are implemented in the traffic generator.

In order to simulate a large number of users (number of users in one shard in WoW is measured in thousands) distributed over the network, multiple traffic generators need to be

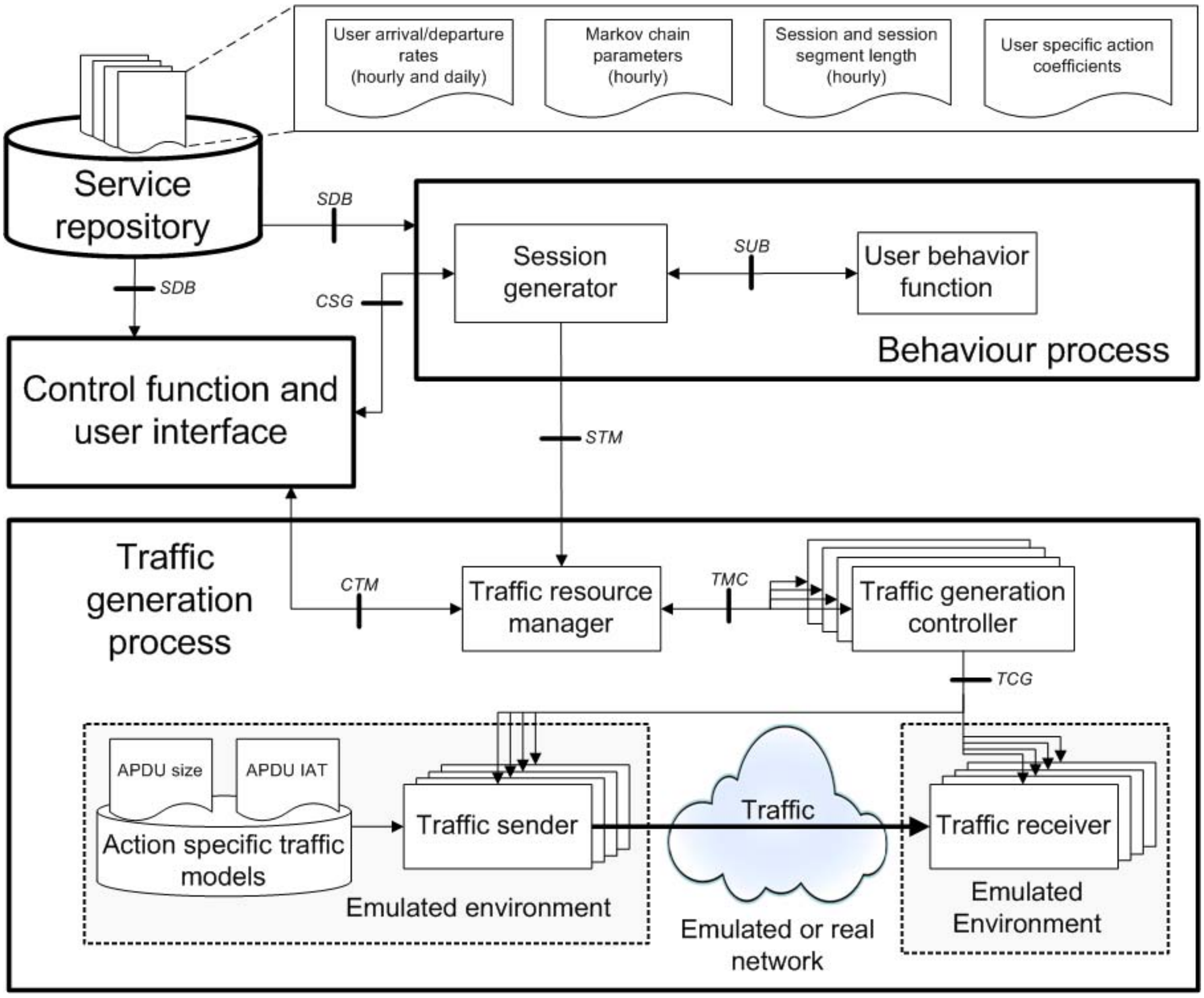


Figure 6.2: Architecture of UrBBaN-Gen functionality

used, and traffic generators in the simulated network environment need to be distributed as well. To reduce the hardware requirements, we apply the concept of virtualization. Through this approach we create as many users as possible on as few PCs as possible.

The components of the source model are realized as specific functional entities in the functional architecture which is depicted in Figure 6.2. All listed functional elements and interfaces are explained in detail in the following sections.

The main components of the architecture are:

- *Control function and user interface* for controlling the parameters of the simulation;
- *Service repository* containing input data which fully describe the simulated service;
- *Behaviour process* in charge of simulating user behaviour; and

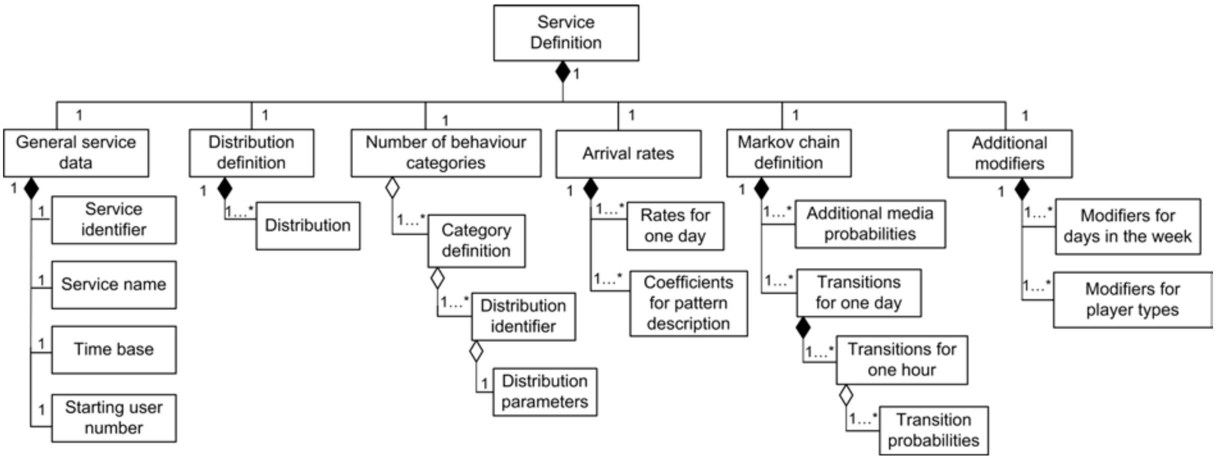


Figure 6.3: Service definition structure

- *Traffic generation process* which transforms the player behaviour to network traffic.

6.1.1 Control function and user interface

The task of the control function is to control other components of the functional architecture. Through the user interface, parameters of the simulation can be changed, and parameters of the models can be adjusted.

The control information is transferred over two interfaces: 1) *Control – Session Generation (CSG)* to the behaviour process, and 2) *Control – Traffic Management (CTM)* to the traffic generation process. Through the CSG interface behaviour simulation parameters are adjusted, and commands for simulation management are transferred (e.g, start simulation). The CTM interface is used to control (i.e., to enable or disable) the traffic generation process.

6.1.2 Service repository

Structure of the service definition is depicted in Figure 6.3 and described in detail in the remainder of this section.

General service data comprises unique service identifier, service name, time base, and starting user number. Time base represents the time frame for which the parameters are defined. In other words, all parameters can be defined for the periods of one day or one week, depending on the goal of service definition – to capture only hourly or also daily patterns. This time frame should not be confused with the granularity of the parameters which can be much more precise (e.g., rates for arrival process are defined for 10 minute intervals based on the characteristics of the measurements).

Distribution definition contains the identifier and the name of each mathematical distribution used in the simulation.

Number of behaviour categories states how many action categories the specific service has. It comprises definition of each category which consists of identifier of the distribution and distribution parameters. One category can be defined with multiple distributions for different hours of the day and for each day of the week.

Arrival rates comprise coefficients for pattern description, and arrival and departure rates. Both coefficients describing the shape of the pattern, and arrival and departure rates are constant over a time interval which can be adjusted (e.g., for 10 minute intervals, 144 values must be defined to fully describe one day (6 per hour times 24 hour = 144), while for 1 hour intervals only 12 values must be listed).

Markov chain definition contains probabilities for transition between specific behaviour categories. They can be defined for each hour of the day and for each day of the week. Also, here are located the probabilities for additional media flows per action category which are defined as time independent.

The last element are additional modifiers which can modify the duration of specific segment either for days of the week, either for specific types of players (i.e., some users types can have preference for certain behaviour category over another).

As previously stated the service repository contains behavioural definitions of supported services. The service definition of our MMORPG service contains:

- *A list of coefficients for describing the shape of the pattern*, for the purpose of simulation with different starting number of user which is in (defined in tag Coefficients for pattern description). These coefficients are multiplied by the average estimated number of users in order to obtain the number of users at a certain point in the day. Values in the list are defined (at minimum) for every hour of the day. To allow daily variations, this list can be defined for each day of the week.

- *Initial number of users and lists with user arrival and departure rates*, on which the arrival and departure processes realized as Homogeneous Poisson Process, are modelled (defined in tags Starting user number and Arrival rates). These rates are defined (at minimum) for every hour of the day for each day of the week. Rates can be defined in higher resolution as well (e.g., for every ten minutes' interval).

- *Markov chain parameters defining transition probabilities* between specific player action categories depending on the hour of the day (defined in tag Markov chain parameters). These parameters could be defined for each day of the week as well. In our application for MMORPGs, only hourly patterns are modelled.

- *Probabilities of adding additional media flows*, depending on the session segment in our case, VoIP communication (defined in tag Additional media probabilities).

- *Session and session segment length models* are also defined for each hour of the day (defined in tag Category definition). As well as for the Markov chain parameters, there is an option to define parameters for each day of the week.

- *User specific action coefficients* contain user preferences towards certain action type (defined in tag Modifiers for player types). For example, for our defined categories, some users might prefer PvP combat to raiding.

6.1.3 Behaviour process

The behaviour process encompasses two functions, namely, *Session generator* and *User behaviour*.

Session generator function handles the number of active users and the length of their presence in the system through arrival and departure processes. It takes input from the service definition through the *Service definition – Behaviour process (SDB)* interface. Depending on the type of simulation chosen, this input is either in the form of arrival and departure rates and starting number of users, or in the form of list of coefficients which represent the normalized pattern of the number of active users. Over the *Session generator – User behaviour (SUB)* interface, the session generator transfers requests for initiation of new users and leaving of the user from the system to user behaviour function, as well as any modifications of the parameters of the behaviour model defined through user interface. Over the same interface it receives data about user behaviour in terms of next of type and duration of next session segment for each specific user.

Through the *Session generator – Traffic manager (STM)* interface, the session generator sends the information regarding user behaviour (i.e, identifier of each user, session segment type and duration) received from user behaviour function, as well as, departure of any existing user from the system. Also, other parameters of the simulation are sent (i.e., starting number of active users). The output sent to traffic resource manager consists of action category performed and the length of the action.

User behaviour function simulates the user behaviour in terms of switching from one action category to another, and assigning the lengths of each session segment. The inputs for this function are Markov chain parameters, distributions of session and segment lengths, user specific coefficients, and probabilities of adding new media flows which are received over the SDB interface. This function primarily communicates with the session generator over the the SUB interface, as described previously.

6.1.4 Traffic generation process

The traffic generation process uses multiple distributed senders and receivers, with a central point being the *Traffic resource manager* process which runs multiple *Traffic generation controller* processes based on the scale of the simulation. *Traffic sender* and *Traffic receiver* are functions dedicated for actual crafting and sending the packets on the network link according to traffic models of action categories.

Traffic resource manager function manages traffic generation and assigns the resources for nodes which participate in the traffic generation (i.e., range of IP addresses and ports). It takes into account how long each resource is taken and when it should be released so it can be reassigned. For example, once a traffic generator has been assigned to generate a flow; after the flow has finished, this traffic generator can be assigned to a new flow. Also, as we decided to use virtualization in order to reduce the hardware requirements, this function controls various virtualization parameters.

Traffic resource manager communicates with *Traffic generation controller* over the interface *Traffic resource manager – Traffic generation controller (TMC)*. Over TMC interface, several types of requests are sent. The request for creating and restarting a virtual node consists of the node type (sender or receiver) and virtualized node IP address. Request for initiation of a new flow comprises IP addresses of the sender and the receiver, type of the session segment, duration of the session segment, as well as the next free port on which the receiver will listen. Because of the distributed architecture, the communication between *Traffic resource manager* and *Traffic generation controller* needs to be reliable, so it is performed over TCP.

If more than one *Traffic generation controller* is used, one instance is replicated on each PC used in the simulation. It directly initiates creation and restarting of virtualized traffic senders and receivers. Over the *Traffic generation controller – Traffic generator (TCG)* interface, requests for new flows are sent. One instance of *Traffic generation controller* controls multiple virtualized nodes which are hosted on the same PC.

Traffic sender and receiver are sending and receiving network traffic according to the parameters received over the TCG interface. These entities are in charge of crafting packets according to the defined models of APDU sizes, and sending them according to IAT models. Thus, traffic models for each action category need to be built in the traffic generator. For each new service to be added and simulated, the traffic models need to be defined inside traffic sender.

6.2 Implementation

In this section we describe the software implementation of the models and the proposed architecture in the User Behaviour Based Network Traffic Generator (UrBBaN-Gen). The implementation of the architecture is realized through three main software modules: 1) *User behaviour simulator*, 2) *Distributed traffic generation control system*, and 3) *Traffic generator*.

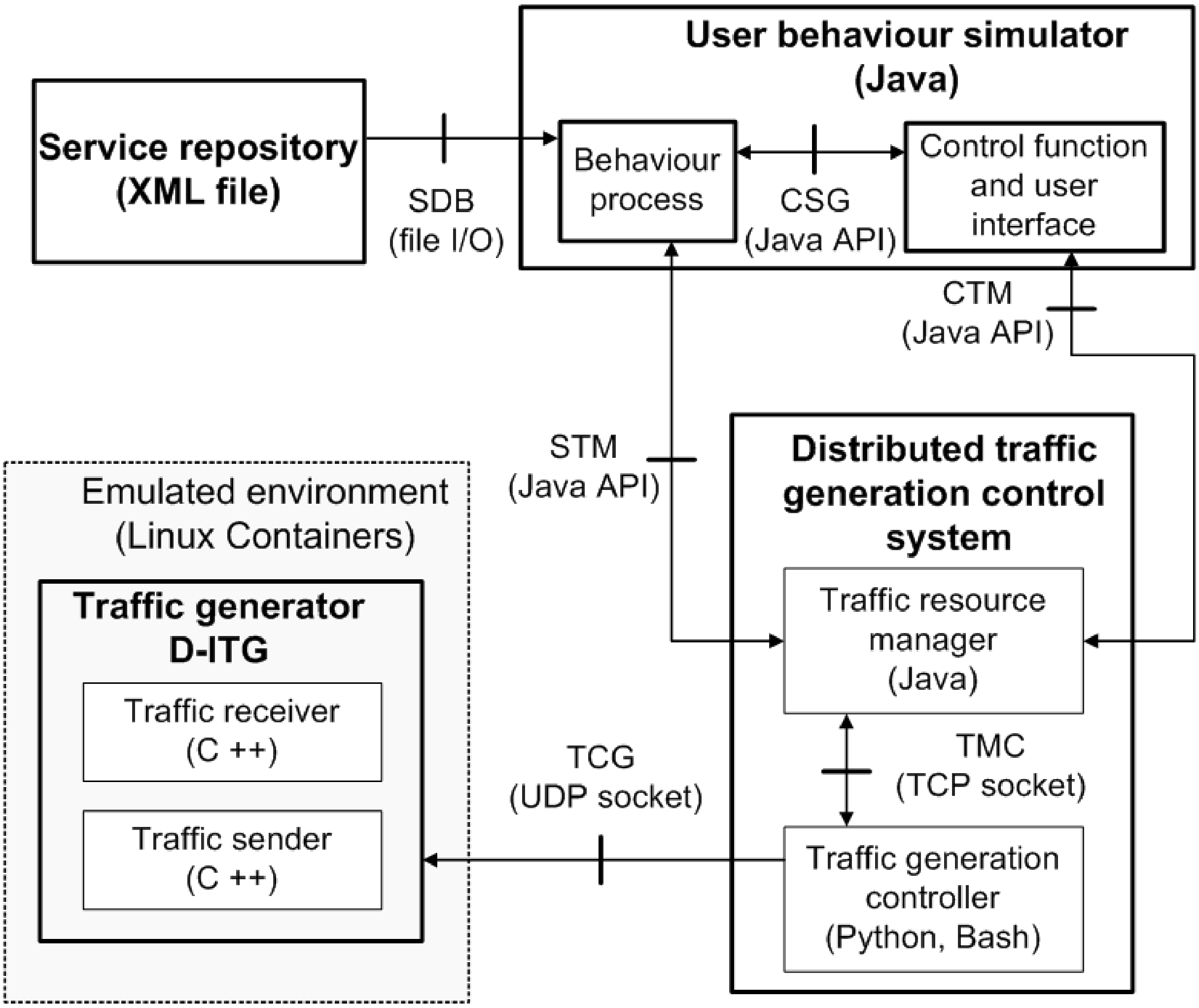


Figure 6.4: Mapping of functional architecture to implementation modules

In Figure 6.4, mapping of functions to implementation modules and the implementation of the interfaces is shown.

1) User behaviour simulator comprises the *Behaviour process*, as well as the *Control function and user interface*. The SDB interface is realized as a file input/output.

2) Distributed traffic generation control system is implemented as Java application (Session generator function) connected through Java API (CSG and STM interfaces) to User behaviour simulator, and as Python and Bash scripts (Traffic generation controller).

3) As for the Traffic generator, we implement our models in Distributed Internet Traffic Generator (D-ITG) developed at Universita' degli Studi di Napoli "Federico II" (Italy) [149]. Commands to D-ITG senders are issued over UDP port 8998 (interface TCG).

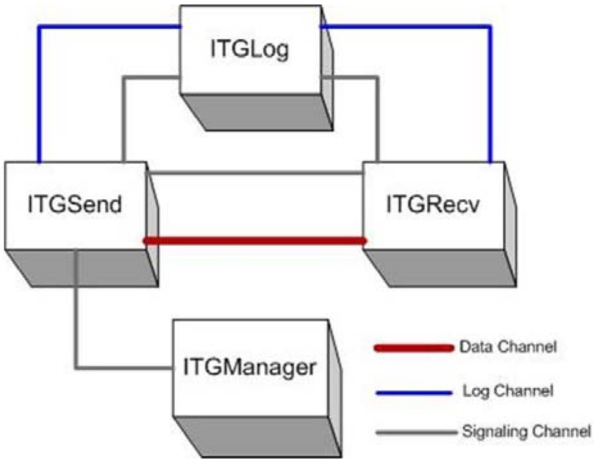


Figure 6.5: D-ITG architecture [1]

6.2.1 Traffic generator (modified D-ITG)

Here we briefly describe the architecture of D-ITG, and its main functionalities, while more information can be found in publications related to D-ITG [150, 151, 1]. Out of the listed generators in Section 3.6, D-ITG was chosen as the solution mainly due to its good scalability, many possible distributions to which the packet size and packet IAT could be modelled, and activity of the D-ITG authors and community.

D-ITG traffic generation is based on traffic flows defined with two parameters – packet size and inter-arrival time between packets. D-ITG offers an array of possibilities for creating different characteristics of the generated traffic through manipulation of parameters of several supported random distributions, as well with a number of protocols from transport and application layer.

The architecture of the D-ITG is depicted in Figure 6.5. D-ITG consists of four main modules: 1) sender (ITGSend), 2) receiver (ITGRecv), 3) logger (ITGLog), and 4) manager (ITGMan).

ITGSend is a system component for creating traffic flows. ITGSend can create multiple traffic flows if run in daemon mode. Packet sizes or packet IATs can be defined through parameters of the following distributions: Exponential, Pareto, Cauchy, Normal, Poisson, on/off, Gama, and Weibull. Supported protocols of the transport layer are TCP, UDP, DCCP (Datagram Congestion Control Protocol), and SCTP (Stream Control Transmission Protocol). Supported protocols of the application layer are DNS, VoIP, and Telnet, as well as the traffic corresponding to active play of two network games, Counter Strike and Quake III Arena.

ITGRecv serves as the packet receiver. It can receive multiple flows from different ITGSend modules.

ITGLog module creates the log file describing the traffic flow generated. It can run locally, or on a remote computer in order to lower the load on the computer performing the traffic generation. ITGLog is used together with ITGDec module which serves as a decoder of created log files. Through decoding ITGDec can calculate parameters such as bitrate, delay, jitter during the whole course of traffic generation. Also, D-ITG supports display of the data resulting from the analysis of the log file through ITGPlot script.

Additionally, D-ITG comes with an API called ITGAPI. Through ITGAPI, remote control of traffic generation is enabled. Remote module can control ITGSender in daemon mode through sending commands over UDP socket on port 8998 on which ITGSender listens.

We used source code of D-ITG version 2.7.0 Beta2, and compiled it on Ubuntu version 10.10. Modifications were made on the D-ITG sender component. We modelled each of the action categories as a new application protocol. New models of application protocols were implemented according to several existing models for generating traffic with on-line game characteristics. New cases of application layer protocols were added in the *ITGSend.cpp* class of the sender component. Also, distributions of APDUs and IATs for each of new models were specified in the *traffic.cpp* class. Header files *ITG.h* and *traffic.h* along with class *ITG.cpp* were modified to include new models.

We noticed that D-ITG could not generate packets with small payload, which were frequent in our models. To enable such packets to be generated, restrictions on minimum payload size – defined in class *ITG.cpp* and *ITGSend.cpp* – were removed. Precisely, the value of minimum payload size in case of generating traffic flow with payload log type option set to "no information is sent in the payload packet" has been set to zero. Therefore, during our tests traffic flows were always generated with option *-p* set to value 2.

In the initial implementation of the models [100], we realized our models on UDP transport protocol as the realization over TCP did not yield satisfying goodness of fit. The problems with TCP versions were resolved in our later efforts through disabling the delayed ACK on receiving machine and Nagle algorithm in D-ITG sender.

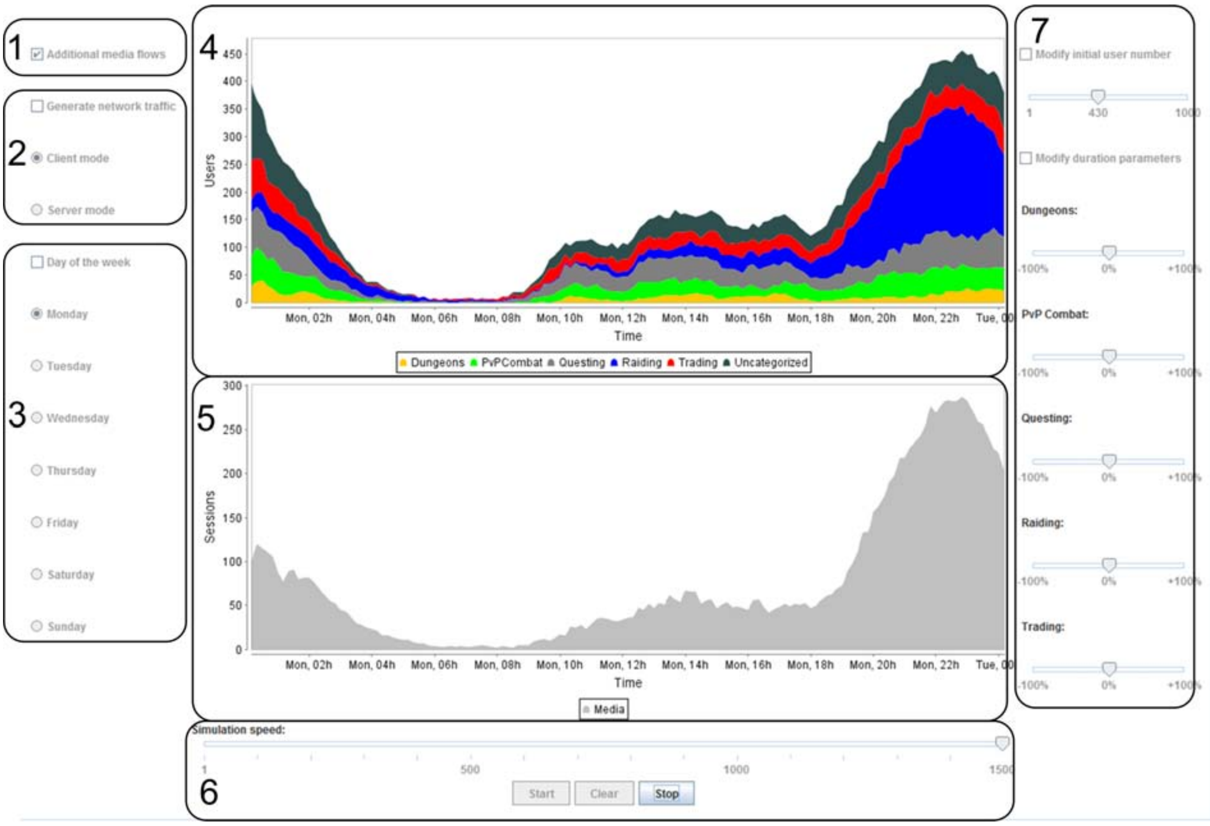


Figure 6.6: Graphical user interface of the user behaviour simulator

6.2.2 User behaviour simulator

User behaviour simulator is an implementation of the *User behaviour process* (including *Session Generator* and *User behaviour function*) and *Control function and user interface* developed in Java. The Service repository uses a file system to store service definitions as EXtensible Markup Language (XML) files. User behaviour simulator also includes the SDB interface (file I/O), and a Java XML parser.

In this section we refer to a “MMORPG user” as “player” and to “user of simulation” as “administrator” in order to avoid confusion. Control over the whole simulation and traffic generation procedure is performed through simulator GUI which is depicted in Figure 6.6.

The GUI consists out of seven areas as marked in Figure 6.6. Area 1 contains the checkbox which enables simulation of additional media flows (in the case of MMORPGs, players using VoIP communication). In area 2 an administrator can decide whether to enable traffic generation, and if it is enabled, whether to generate server or client side traffic. Area 3 serves for the control over the time frame of the simulation (i.e., the whole week, or specific day of the week). UI also comprises the panel for displaying the simulation results in form of dynamic graphs. The upper graph (area 4) shows the number of active players in each action category, and the lower graph (area 5) depicts the number of active players which are using VoIP. Both graphs are dynamically changing through the course of the simulation. Speed of the simulation can be controlled through a slider in the area 6, which also contains the buttons for starting and stopping the simulation, as well as button for clearing the graphs. This enables simulation to be run in real time, or to be accelerated up to 1500 times. Parameters of the model, specified in the service definition XML file, can be modified through multiple sliders in area 7 (i.e., administrator can modify the values of initial player count and the session segment duration of each action category). Through manipulation of these parameters, behaviour scenarios different from the one specified in the service definition can be easily simulated.

Each player in the simulation is represented by an individual thread, which “decides” the sequence of his/her activities during a session as defined by the model of session segments probabilities (i.e., Markov chain). Each player thread also generates the duration of each segment based on the distribution for specific action category.

As previously stated, the administrator has the option to run just the simulation without generating traffic, or to run the simulation along with the network traffic generation. The control information is sent over the CTM interface. If generation of network traffic is enabled, simulator sends the data into the system for traffic generation over the STM interface.

Three log files are created at the end of the simulation:

- 1) log file with the duration of player sessions;
- 2) log file containing the number of active players in the system (taken every ten minutes in the simulated time); and
- 3) log file which records the simulation and contains the data about simulation progress including timestamps of events marking arrival and departure of a player, as well as timestamps of each player’s transition to a new action category.

After the logs are created, information such as the average number of players in the simulation or the share of action categories can be analysed independently of the simulator.

6.2.3 Distributed traffic generation control system (DTGCS)

The task of the Distributed traffic generation control system is to control the distributed traffic generation, according to the input from the User behaviour simulator. The traffic generation can be performed on multiple PCs, depending on the scale of the simulation. Each PC participating in the simulation, may host multiple virtualized nodes. For the virtualization of instances of D-ITG senders and D-ITG receivers we use Linux Container (LXC) technology [152]. In each LXC, only one instance of D-ITG sender or D-ITG receiver is run. Senders are run in daemon mode so they can receive multiple flow requests. All LXCs are connected to a Linux bridge [153]. Beside virtual interfaces of the LXC containers, the the network card interface can also be added to a bridge so the traffic from virtualized nodes can be sent to the real network. A Linux bridge is a way to connect two Ethernet segments together in a protocol independent way. Packets are forwarded at Layer 2, based on Ethernet address, rather than IP address.

The system for control of the distributed virtualization consists of a Java module, and a set of Python and Bash scripts which implement the functionalities of *Traffic resource manager* and *Traffic generation controller* respectively. *Traffic resource manager* function is realized as a Java module and is connected to User behaviour simulator through Java API. The module performs the following actions:

- reads the input parameters of the simulation from a properties file (i.e., IP addresses of all PCs which are used for traffic generation (commonly one PC for receiving the the traffic and multiple ones for sending the traffic), and information regarding IP addresses of the virtualized nodes);
- takes care about initialization of the virtualized nodes (i.e., estimates the number of virtualized nodes needed for the simulation, and sends requests for initiation of each node);
- sends requests for the initiation of a new flow based on the arrival of a new session segment;
- handles the distribution of free ports on receivers (i.e., stores the list of free ports as well as busy ports, it “knows” which port is busy and for how long); and

- restarts the virtual nodes that reach their flow limit (around 150 flows for senders, and 2000 for receivers).

The TMC interface is realized as a TCP socket.

Traffic generation controller is replicated on each PC participating in the simulation. It comprises several Python and shell scripts for various purposes:

- *initializeNetwork.sh* is a Bash script for configuring the network elements needed for the simulation (i.e., creating and configuring the Linux bridge, as well as, adding the hardware network interface in the Linux bridge so the traffic generated in virtualized nodes can enter real network;

- *behaviourListener.py* is a Python script for receiving the requests from traffic resource manager (over a TCP socket), parsing those requests, and starting other scripts according to the received command;

- *sender.sh* is a Bash script for creating a virtualized node running D-ITG sender and joining the node to the Linux bridge;

- *receiver.sh* is a Bash script for creating a virtualized node running D-ITG receiver and joining the node to the Linux bridge;

- *itgManagerImpl.py* is a Python script for sending commands to D-ITG senders in daemon mode over UDP port 8998 which is reliable as those UDP packets are transferred only through the kernel of the system, so there is no possibility of a packet loss; and

- *killLxc.sh* is a Bash script for destroying an LXC container.

The activity diagram of the implementation of *Traffic generation controller* is depict in Figure 6.7. Scripts *initializeNetwork.sh* and *behaviourListener.py* are manually started. While *initializeNetwork.sh* ends after it configures the network (i.e., in few seconds), *behaviourListener.py* is constantly active and listens for requests initiated by the Java segment of DTGCS. After the request has been received, it has been handled in a new sub-process. Based on the request type, different actions are taken. The script *behaviourListener.py* runs *sender.sh* or *receiver.sh*, if it needs to initiate sender or receiver respectfully. In case that an existing LXC needs to be

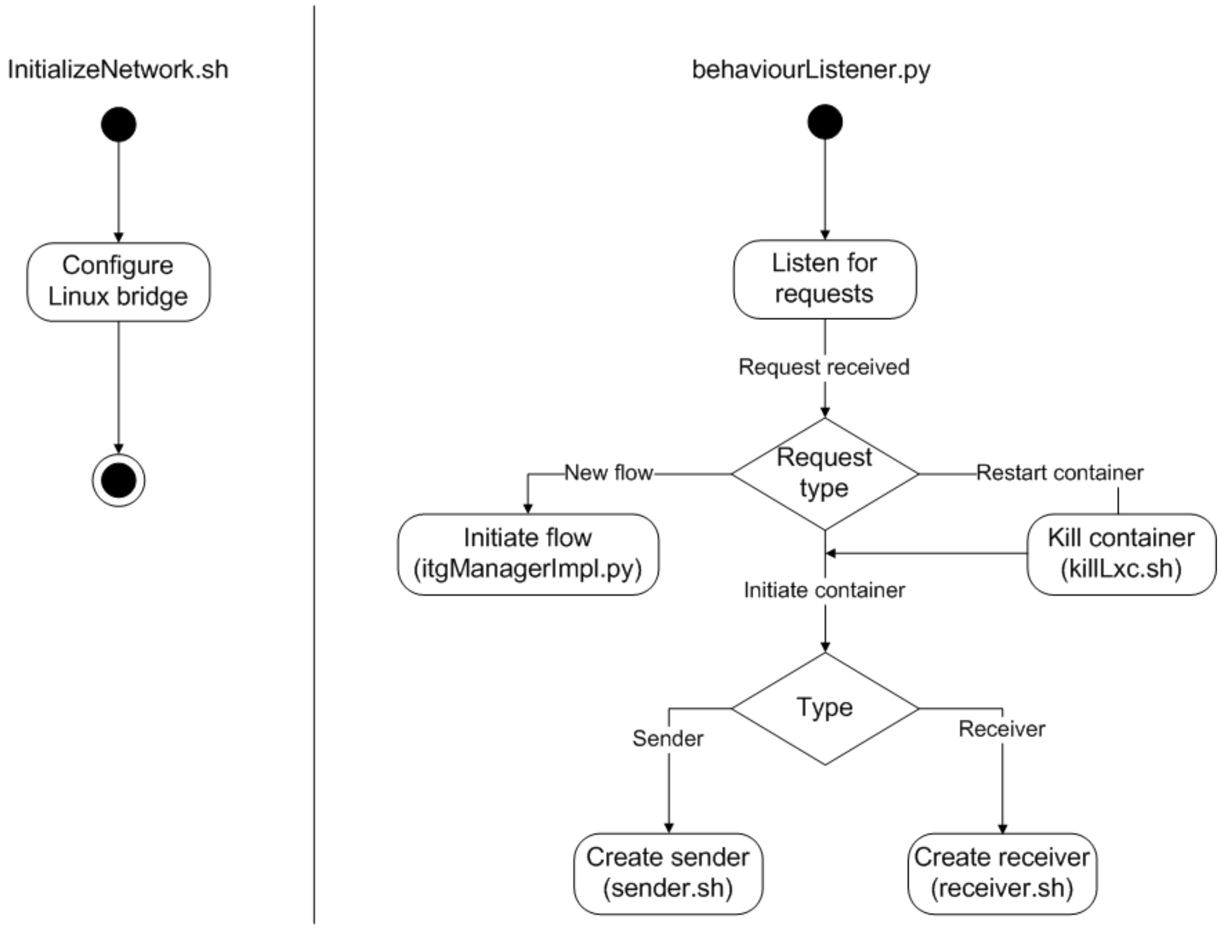


Figure 6.7: Activity diagram of the implementation of *Traffic generation controller*

restarted, first it is destroyed through *killLxc.sh*, and afterwards a new LXC is initiated. If the request was carrying information regarding a new flow which needs to be started, that information is passed to *itgManagerImpl.py* which transfers it over UDP port 8998 to appropriate D-ITG sender.

The deployment of UrBBaN-Gen in a laboratory testbed is illustrated in Figure 6.8. The setup uses 4 PCs, with following configuration:

- PC1: Dual Core @2.53GHz, 4GB RAM: Windows 7, Java;
- PC2: Quad Core @3.3 GHz, 4GB RAM: Ubuntu 11.10, LXC, Linux bridge, Python;
- PC3: Quad Core @3.3 GHz, 4GB RAM: Ubuntu 11.10, LXC, Linux bridge, Python; and
- PC4: Quad Core @3.3 GHz, 4GB RAM: Ubuntu 11.10, LXC, Linux bridge, Python.

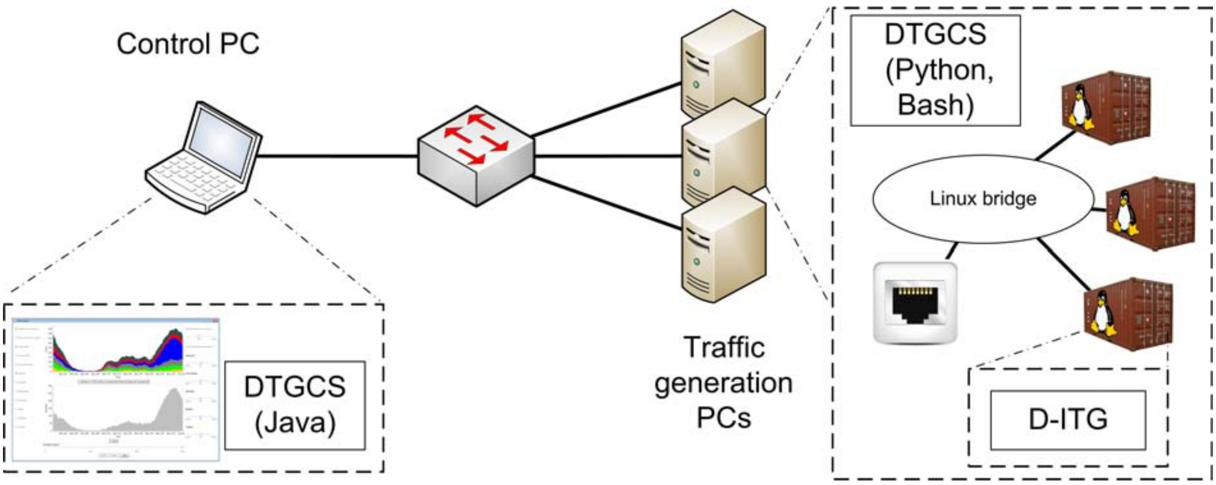


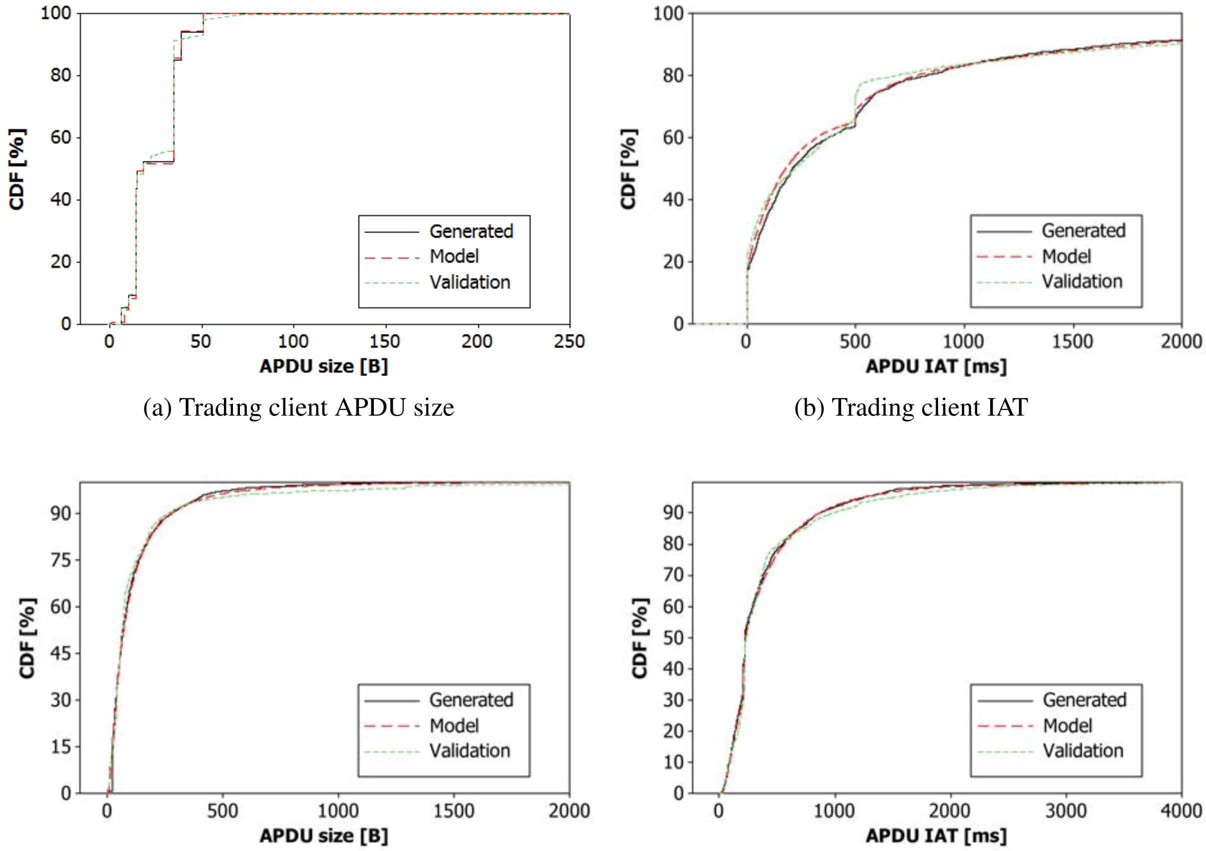
Figure 6.8: UrBBaN-Gen laboratory testbed setup

User behaviour simulator and Java part of DTGCS are collocated on a control PC (PC1). Multiple PCs can be used for traffic generation (here PC2, PC3, and PC4), and each one of them holds Pyhton and Bash part of the DTGCS. Also, on each PC used for traffic generation a Linux bridge is created, and a hardware network interface is added to it. As LXCs are created, they are also joined to Linux bridge. In each of the created LXC, an instance of D-ITG sender or D-ITG receiver is run. Traffic generated in any LXC container holding D-ITG sender goes through Linux bridge and hardware network interface into the real network. From real network it again enters the Linux bridge on another PC, and to the LXC container which holds D-ITG receiver.

6.3 Validation of proposed source model of network traffic

The goal of this thesis is to create a model which describes the traffic of MMORPGs in a more realistic fashion than previous models. In order to demonstrate that this goal has been achieved, we compare results of our models with previous models of WoW network traffic. We validate our models by comparing the generated synthetic traffic traces generated by UrBBan-Gen with real traces captured in the network. For validation of behaviour models, we compare the results of the behaviour simulation with real measurements performed by Lee et al. [15]. The real traces we perform validation on, are the second half of our original traffic measurements, as the first half was used to obtain the parameters of the model. Also, we test the model against results obtained through measurements in access network in Sweeden by Kihl et al. [65].

6.3.1 Validation on the level of a single flow



(a) Trading client APDU size

(b) Trading client IAT

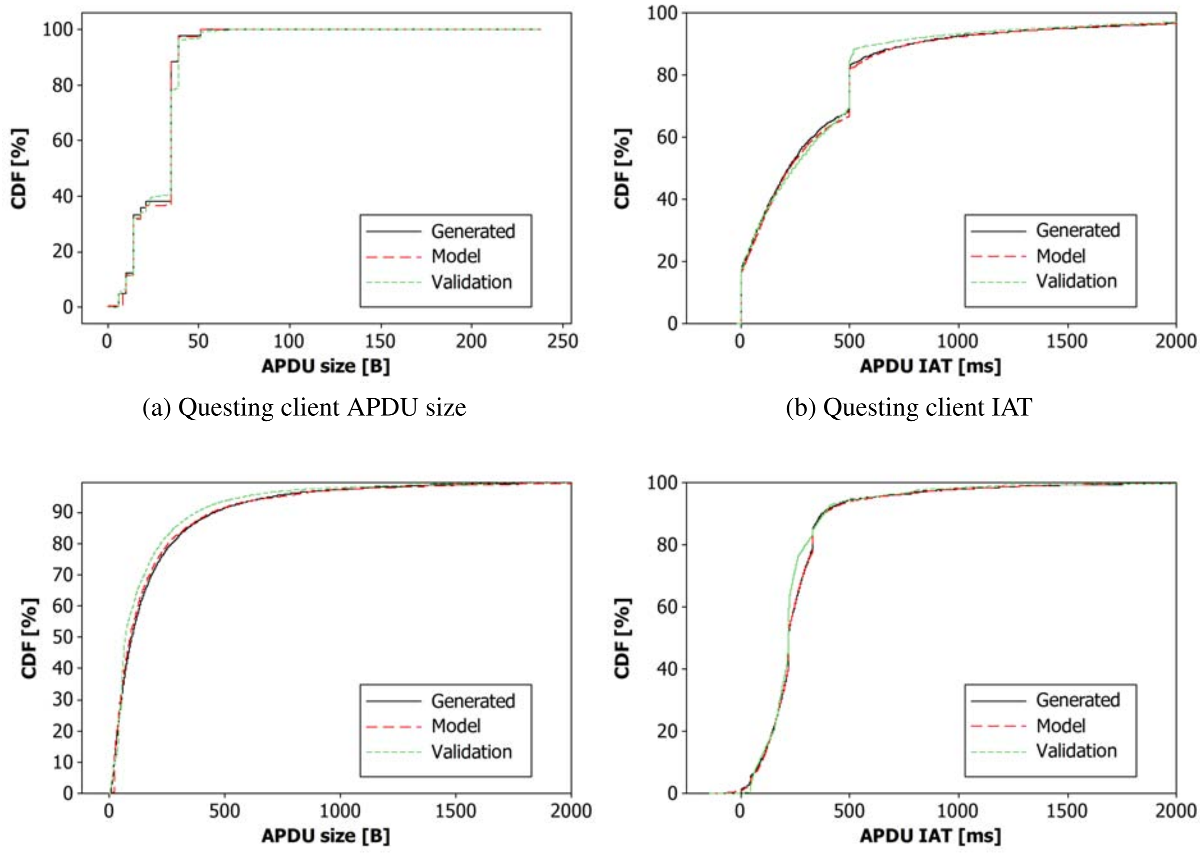
(c) Trading server APDU size

(d) Trading server APDU IAT

Figure 6.9: CDF of trading client APDU size

In order to validate that the generated traffic has the required characteristics, we have performed comparison between the parameters of the analytical model, generated traffic, and the traffic from the validation traces. We perform this through plotting CDFs of APDU sizes and IATS of the model, generated traffic, and validation traffic obtained from validation traces. In this way we can see whether the model captures the general trend in comparison with validation traces, how closely generated traffic follows the parameters set by the model (i.e., are there any discrepancies due to implementation), and in the end, how closely does the generated traffic corresponds to the real traffic (i.e., validation traffic).

Figures 6.9 – 6.13 depict each of the modelled parameters (client APDU size, client APDU IAT, server APDU size, and server APDU IAT), for each of the defined action categories (trad-



(a) Questing client APDU size

(b) Questing client IAT

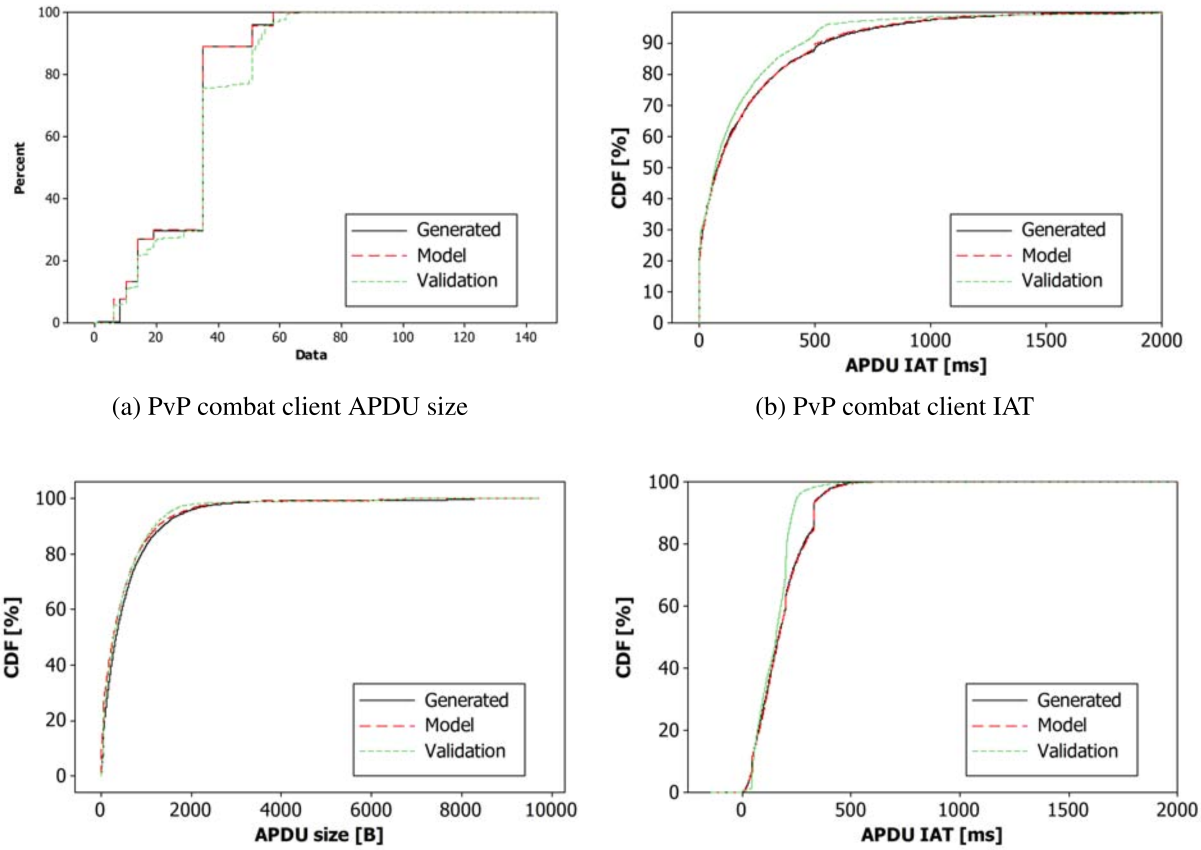
(c) Questing server APDU size

(d) Questing server APDU IAT

Figure 6.10: CDFs of questing parameters

ing, questing, PvP combat, dungeons, and raiding). Figure 6.9 shows the properties of the trading category. It can be observed that all parameters have very good fits. All parameters of the generated traffic are very closely following the model which demonstrates that implementation of the traffic generator is very good. APDU size of the client traffic is closely following the model, but due to the discrete steps nature of the model there are some discrepancies. Also, the model slightly underestimates the value of 35 B while it overestimates the values of 39 B. As for the client side IATs, the model underestimates the spike at 500 ms. Both server side validation APDU sizes and IATs follows the model closely, except for the tail which is slightly underestimated by the model.

Figure 6.10 shows the properties of the questing category. It can be observed that all parameters have very good fits with only minor discrepancies in the client IAT. APDU size of the validation client traffic shows almost the same curve as the model, with only minor differences



(a) PvP combat client APDU size

(b) PvP combat client IAT

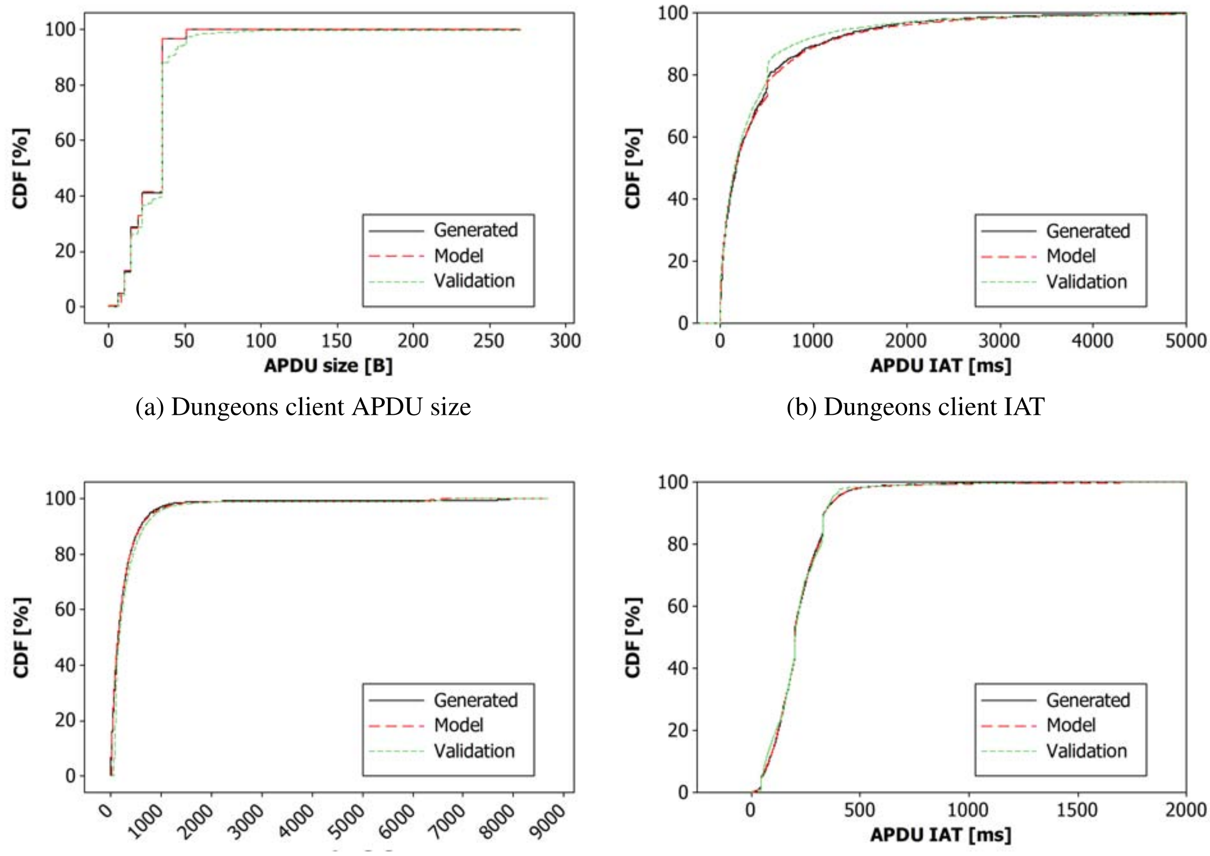
(c) PvP combat server APDU size

(d) PvP combat server APDU IAT

Figure 6.11: CDFs of PvP combat parameters

due to the discrete steps nature of the model. As for the client side IATs, the model underestimates the spike at 500 ms with respect to validation traces while for the remainder of the data the curves almost coincide. Server side validation APDU sizes show that model is a bit underestimating the lower part of the dataset up to 400 B. Server side validation IATs show discrepancies with respect to the model only on a short period between 300 and 400 ms.

Figure 6.11 shows the properties of the PvP combat category. While, in general, all parameters have good fits, larger discrepancies exist in comparison with previous categories, especially in both client and server IATs. All parameters of the generated traffic are very closely following the model. APDU size of the validation client traffic shows that model is overestimating the value of 35 B and underestimating higher values. As for the client side IATs, the model underestimates the lower part of the dataset up to 500 ms with respect to validation traces. Server side validation APDU sizes shows closely coinciding curve with the model. Model of the server



(a) Dungeons client APDU size

(b) Dungeons client IAT

(c) Dungeons server APDU size

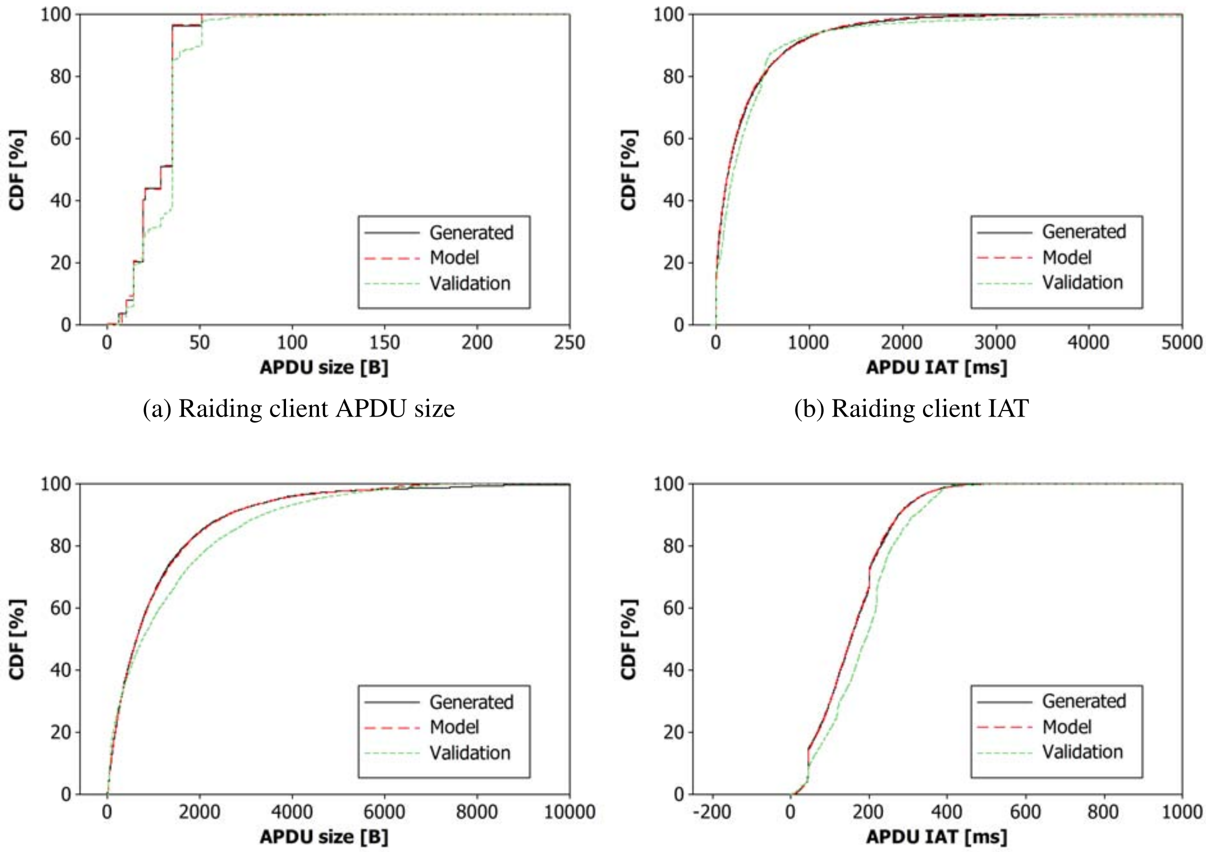
(d) Dungeons server APDU IAT

Figure 6.12: CDFs of dungeons parameters

side IATs overestimates the tail of the distribution, which is, in general, shorter than previous categories.

Figure 6.12 shows the properties of the dungeons category. All parameters of the generated traffic are very closely following the model. APDU size of the validation client traffic shows that model is slightly overestimating the value of 35 B and underestimating higher values. As for the client side IATs, the model underestimates the spike at 500 ms with respect to validation traces. Server side validation APDU sizes shows closely coinciding curve with the model, as well as the server side IATs. In general, models of dungeons category are the best fits in comparison with other categories.

Figure 6.13 shows the properties of the raiding category. The fits of this category are, in overall, not as good as in other categories. Again, all parameters of the generated traffic are very closely following the model, but there are more discrepancies between model and the



(a) Raiding client APDU size

(b) Raiding client IAT

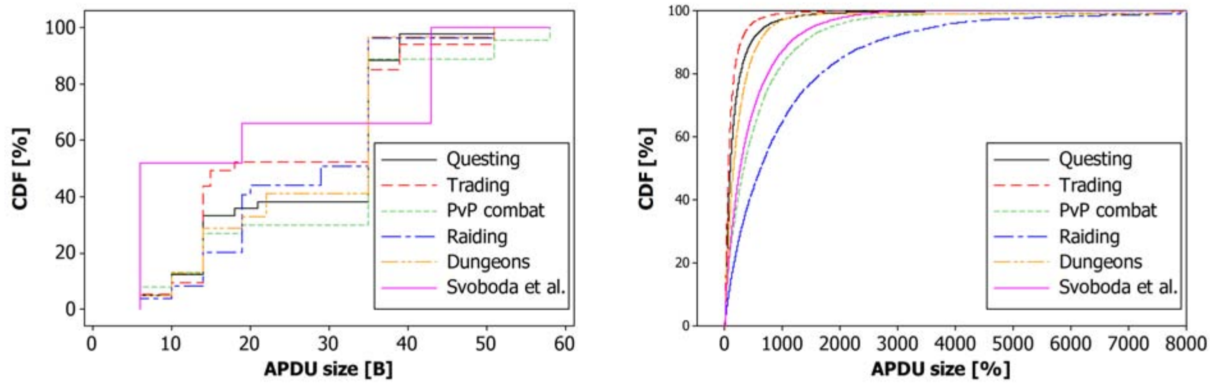
(c) Raiding server APDU size

(d) Raiding server APDU IAT

Figure 6.13: CDFs of raiding parameters

validation traces. One of the causes might be that a relatively small amount of raiding traces was available for the modelling and validation procedure (only 10 traces). Client APDU sizes of the validation traffic shows that model is underestimating values of 19 B and overestimating value of 51 B. As for the client side IATs, the model overestimates the lower part of the distribution up to 500 ms and underestimates the tail. Server side validation APDU sizes shows that the model is, in general, underestimating the size of the APDU. The largest discrepancies are shown for the server IAT where, where the model is underestimating the IAT times.

In general, client side IATs have a spike at 500 ms which indicates that some keep–alive mechanism is active. These spikes are more emphasized for slow paced actions (i.e., trading and questing), while they are almost non existent on the most dynamic categories (i.e., PvP combat and raiding). Client side APDU sizes though all categories show discrete steps. Server APDU sizes, on the other hand show curves, while the server side IAT proved to be complex



(a) Client (b) Server

Figure 6.14: CDFs of APDU size of action categories and model by Svoboda et al.

and difficult to model properly. In general, the models of all action categories are very good fits when compared with validation traces. It should be noted that, in order to achieve this level of goodness of fit, models needed to be very complex. All models are realized as mixture models (i.e., composition of multiple distributions).

In order to confirm the hypothesis that our WoW network traffic model is improved in comparison with existing models, we perform comparison with previously published models by Svoboda et al. [59] and Park et al. [2]. Svoboda et al. create one model for the general network traffic of WoW, while Park et al. create several models for their behaviour classes: Hunting the NPCs, No play, Combat with players, and Moving, but do not provide any reasoning for such classification. Both models are described in more detail in Chapter 3. In Figure 6.14 we can see the comparison between our and Svoboda's model of APDU sizes for client and server traffic. Model by Svoboda et al. seems like an average of categories on the server side, but this means that it is significantly overestimating some behaviours (i.e., trading, questing, dungeons) and underestimating others (i.e., raiding). On the client side, the differences are more significant, as none of the values Svoboda's model list are amongst the most frequent values of our model. As for the APDU IATs Svoboda et al. model both the client and server APDU IATs with a single function, while it can be observed from the validation traces that the differences are significant, not only between client and server traffic, but also amongst different action categories, and that our model successfully captures those differences. Thus, we conclude that the model proposed in this thesis captures the properties of the traffic in more detail.

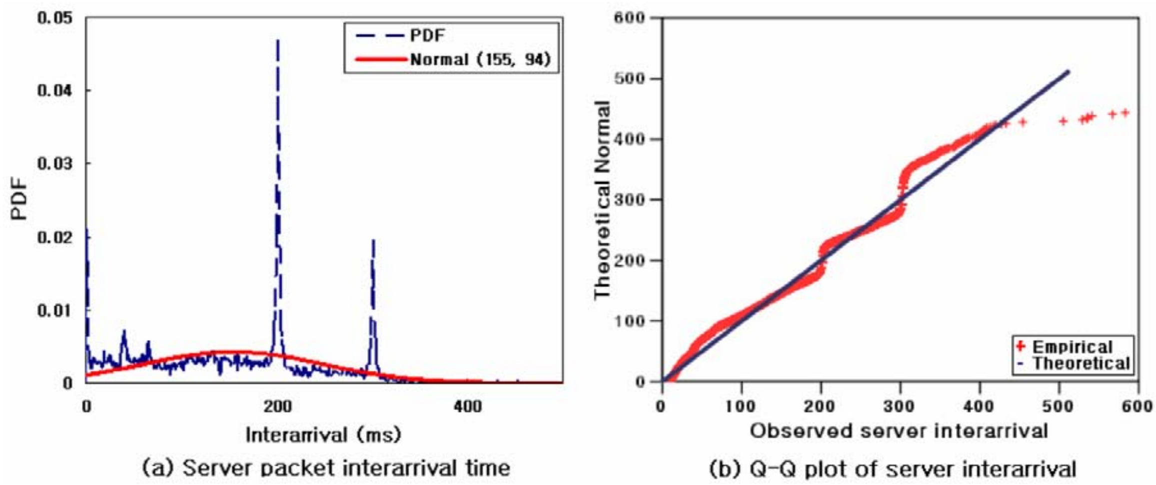


Figure 6.15: QQ plot and PDF of the server IAT model by Park et al.[2]

As Park et al. have not published the parameters of their model, we can not create an explicit comparison. Some conclusions regarding the fit of the model can be drawn as the authors clearly state quote: “However the models of the packet size and interarrival time of the games don’t exactly correspond to the empirical distributions. Through the Q-Q plots, we confirm that empirical data is identical to the model on some intervals or points in fig. 4” [2]. We also present their graphs regarding the fit of the IAT times in Figure 6.15. Additionally, the authors do not model the APDU, but only packet sizes observed on the network. Therefore, we conclude that our model captures the properties of the WoW network traffic better.

6.3.2 Validation of of the model through inspection of aggregated traffic

We do not posses our own measurements of the aggregated traffic, as the traffic measurements of aggregated traffic can be performed either on the game operator side, or in the core network. A measurement of WoW traffic was performed in the core network of Sweden [65]. We gratefully acknowledge the help of authors Andreas Aurelius and Maria Kihl who disclosed some segments of the dataset acquired in these measurements. The disclosed dataset contains number of active players in the monitored network, sum of inbound (server), and sum of outbound (client) traffic. The period which dataset describes is one month (i.e., March of 2010). The state of the network is reported each five minutes resulting in 8928 entries in the disclosed dataset.

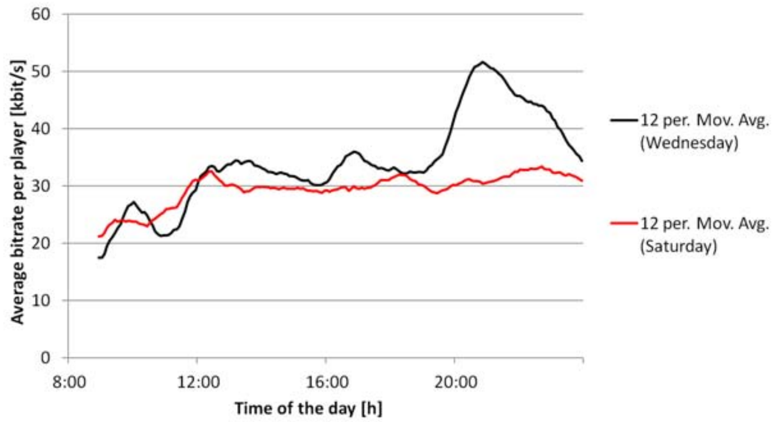


Figure 6.16: Average empirical load per player on Wednesday and Saturday

The highest number of active players at a single time was 70.

The validation of the aggregated traffic is performed through inspection of the average load per player during the hours of the day, and days of the week in the disclosed dataset, and comparison with the simulation results of the behaviour driven simulation. For the purpose of brevity, further in this section we will refer to "empirical results" as those extracted from the dataset of measurements performed in the access network in Sweden [65], while "simulation results" are those based on an average values of five simulations performed with our UrBBaN-Gen.

The hypothesis under test is: The average network load per player varies due to different player behaviour patterns, and these variations can be captured through use of our model. We test this hypothesis through inspection how does the load per player vary in the empirical results and compare it to simulation results. Also, we illustrate how would average network load looks with the use of Svoboda's model.

Analysis of the empirical results of number of active players and their aggregated (i.e., server and client) bandwidth usage shows that there are significant differences in average load generated by a single player, both in hours of the day, and days of the week. These variations in the load can go up to six times in a single day. As for different days of the week, there is less variation.

In Figure 6.16 the average values of bandwidth usage per active player of each hour for Wednesday and Saturday are displayed, based on empirical results. We display only the time

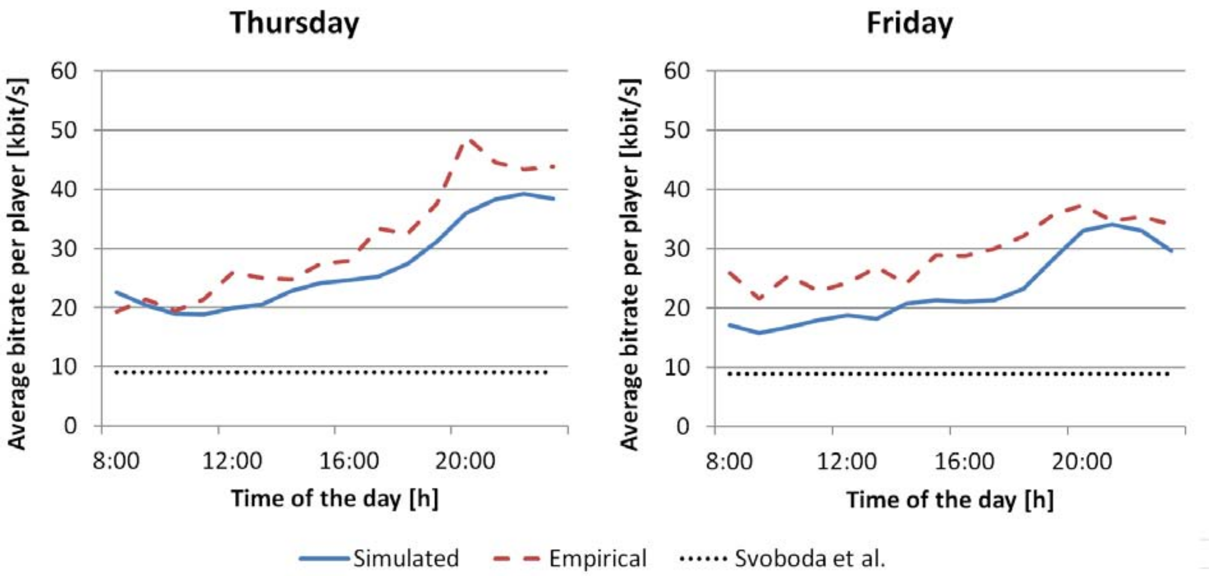


Figure 6.17: Average load per player simulated and empirical

from 8:00 to 0:00, as late in the night and early in the evening number of active players is rather small. It is evident that there is a significant difference, especially in the evening. This level of discrepancy can not stem from the difference in player numbers, as it was shown earlier (in Chapter 5) that there are more players over the weekend (Figure 5.15), but only from different player behaviour. This assumption can be confirmed through inspection of Figure 5.11 in which it can be observed that raiding, as the most demanding action category, is much more often performed on Wednesday than on Sunday. Also, from Figure 5.12 it can be seen that raiding is dominantly performed in the evening, thus we conclude that ***the hypothesis that player behaviour has significant influence on both average bitrate of a single flow and consequently, the aggregate traffic, is confirmed***.

In order to validate how good does our model capture the variation of network load per player, in Figure 6.17 we present the average load per player from empirical results, results obtained from our simulation, and estimate based on Svoboda's model (i.e., static load). We perform this comparison for another two days of the week, for Friday (another day with lower share of group based actions and especially raiding) and Thursday (a day with high share of raiding) as can be seen in Figure 5.11.

When inspecting Figure 6.17, the first notion is that, again, the higher load is seen on the day with higher share of raiding (Thursday). The second notion is that the simulated traffic load

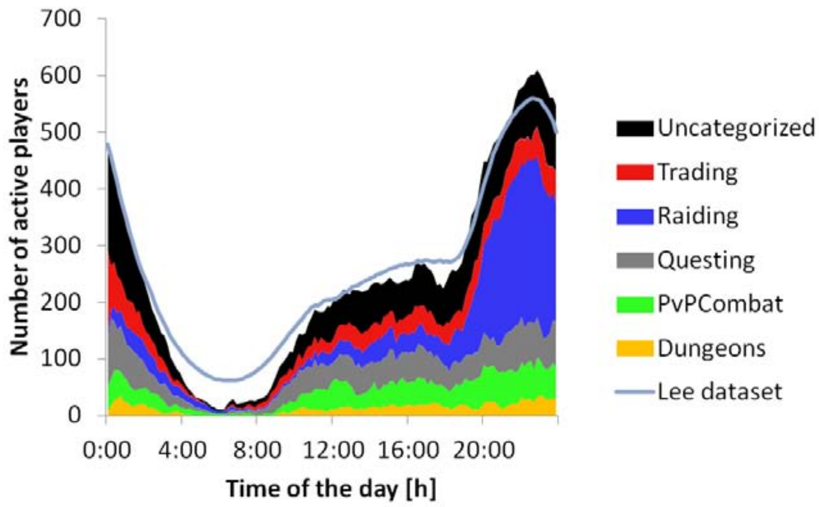


Figure 6.18: Active players per action category (simulated)

is, in general, lower than the load from the real network, but it is higher than the load estimated by Svoboda's model. This discrepancy can be caused by the following factors: a) assumption of uncategorised category having the lowest requirements, and b) evolution of the game (i.e., the traffic measurements on which the model is based and those from the core network are taken in different versions of the game). While there are differences between the simulated and empirical data, it is evident (Figure 6.17) that the general trends of increase and decrease of the load per player are preserved. ***This means that, although traffic properties of the game can change due to game updates, the trends caused by player behaviour are preserved***. Results presented in Figure 6.17 confirm the second part of the presented hypothesis – ***our model captures the variation of average network load per player, and therefore outperforms previous models***.

6.3.3 Validation of behaviour simulation

In order to validate the results of the player behaviour simulation regarding number of players, a comparison with the dataset by Lee et al. [15] is performed. The comparison approach is to plot an average number of users in the dataset obtained through the measurements over the results of one simulation (in Figure 6.18). The simulation tends to slightly underestimate the number of users in the morning (i.e., 3:00-10:00), but captures the main trends such as high decline in the late night, slow increase of players in the morning, and a steep increase in the evening.

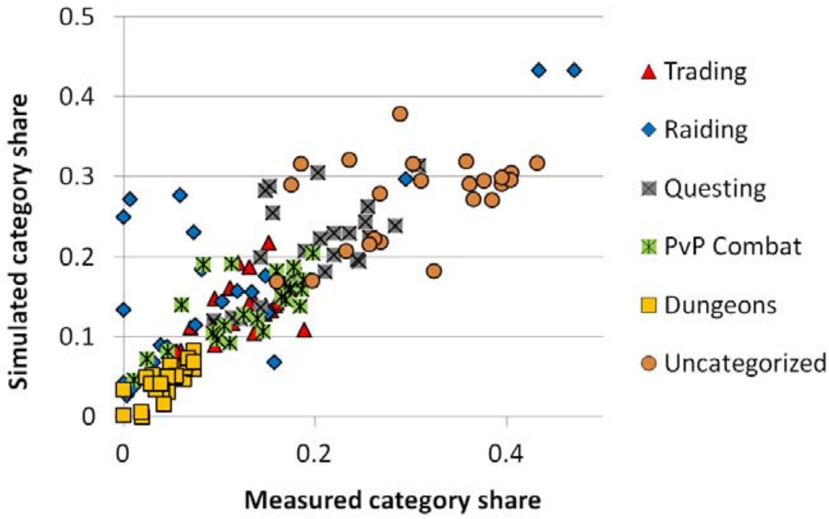


Figure 6.19: Simulated vs measured share of action categories

In order to give a more precise numerical estimation of the goodness of fit of behavioural simulation, a comparison of average results of five simulations and the measurement data is performed. The relationship between percentages of players in each action category during the hours of the day are depicted in a scatter plot chart displayed in Figure 6.19. If simulation would describe the measurement data perfectly, all of the points would form the diagonal. Figure 6.19 shows that slight discrepancies exist, but in general, the diagonal trend is clearly visible. Differences exist, especially in the questing category. Simulation captures dungeons patterns most closely, overestimates raiding in early morning periods, and slightly disperses uncategorised.

6.4 Summary and outlook

In this chapter we have presented the functional architecture through which we realize our source based network traffic model for MMORPGs (using WoW as a case study). We described the functional elements of the architecture in detail, as well as the interfaces between them. We have presented the software implementation of the proposed architecture in User Behaviour Based Network Traffic Generator (UrBBaN-Gen). Finally, we have validated our source model through inspection of synthetic and real traffic, and have confirmed that our model describes the network traffic of MMORPGs better than existing models.

Conclusion

As global game traffic is estimated to grow with an annual rate of 37% in the period from 2009 to 2014 [6], there is an evident need for models of game traffic in order to efficiently test and plan the newtork. MMORPGs are one of the fastest growing game generes, with millions of users which require very high QoS. The main research problem of this thesis is related to variable characteristics of network traffic of MMORPGs. The main goal of the thesis is to create a MMORPG source based network traffic model, based on the application level user behaviour. The model is created as such to capture both behaviour on the level of a single user and the behaviour on the service level (i.e., multiple users).

Virtual worlds of MMORPGs are very large and complex, with many possible actions which players can perform. Therefore, in order to study, measure, and model player behaviour, all possible player behaviours need to be classified in a limited number of categories. Currently, some categorisations exist in the related work [2, 10], but the presented categories are not explained nor validated, and in general, seem arbitrary. Therefore, there is a need for a ***classification of user actions in the virtual world of MMORPGs***, which is, together with ***characterization of associated network traffic*** of each category, the first contribution of this thesis.

The first step for categorization of possible situations in the virtual world is to identify metrics which characterize such a situation. We propose initial categorization of user application level behaviour based on player motivation and game mechanics with focus on player progression. The initial categorization is consolidated based on the values of the parameters of the identified metrics for each proposed category, and finally five user behaviour action categories are identified: Trading, Questing, Player versus Player combat, Dungeons, and Raiding. The answer to whether the identified categories are distinctive enough is obtained though inspection of the characteristics of network traffic of each category. Finally, a network traffic model is created consisting of APDU sizes and APDU IATs for both client and server traffic of each action category.

The source of the MMORPG network traffic is the player. Therefore, in order to create a source model of network traffic, we must model the player and his/her behaviour. ***User behaviour model based on categories of user actions, motivational parameters, and identified behavioural patterns on application level*** is the second contribution of this thesis. The model is created on the base of three sets of action specific, behavioural measurements on real MMORPG players. On a single player level, the behaviour model is described with distributions of session length and action specific session segment length, and probabilities of switching between action categories (i.e., first order Markov chain). All of the parameters are inspected for each hour of the day, and each day of the week in order to capture patterns of player behaviour. On the level of multiple players, the behaviour is described with the number of active players at the start of the session, and a list of values comprising rates of the arrival and departure process. Certain relations between observed patterns of player behaviour and player motivation are confirmed, but not modelled, due to the lack of measurement data.

The final step is the integration of the models of network traffic and player behaviour through an ***architecture and implementation of traffic generator***, which is together with ***the verification of the model through comparison of synthetic and real traffic***, the third novelty of this thesis. The created implementation is named User Behaviour Based Network Traffic Generator (UrBBaN-Gen). UrBBaN-Gen is implemented in Java, Python, and Bash scripts, and comprises three modules: User behaviour simulator, Distributed traffic generation control system, and Traffic generator for which D-ITG is used. Synthetic traffic created by UrBBaN-Gen is tested and compared with real traffic, and results show that characteristics of the synthetic traffic closely follow those of real traffic. Also, our model is compared with the existing models of WoW network traffic, and the advantages of our model are illustrated.

The results of this thesis may be applied for:

- Testing of network equipment and protocols, not only for MMORPG, but for other complex IP services;
- Planning of MMORPGs infrastructure (both network and server);

- Testing of virtual world partitioning mechanisms;
- Development of player churn algorithms;
- Dynamic load balancing for servers; and
- Dynamic capacity allocation in networks.

Future work is aimed at further testing and scalability enhancement of UrBBaN-Gen to enable its use for network testing. We consider that this tool can create realistic traffic conditions of any complex IP service, not only MMORPGs. We also aim to investigate additional applications of behavioural simulation for dynamic load balancing mechanisms for both server and network capacity in cloud based services. Specifically, we aim to study possibilities of providing better QoS levels based on user behaviour using technologies such as OpenFlow.

Appendix A. Questionnaire regarding use of voice communication

1. What programs do you use for voice communication (multiple answers)? Offered answers:

    Skype

    TeamSpeak

    Ventrilo

    WoW Voice Chat

    Other (which one?)

    Not using voice communication

2. How often do you use voice communication while performing actions related to trading and crafting virtual items?

    Never

    Rarely

    Sometimes

    Often

    Always

3. How often do you use voice communication while performing actions related to solving tasks given by NPCs (questing)?

    Never

    Rarely

    Sometimes

    Often

Always

4. How often do you use voice communication while performing actions related to combat between players?

Never

Rarely

Sometimes

Often

Always

5. How often do you use voice communication while performing actions related to fighting in small groups versus NPCs in instances (dungeons)?

Never

Rarely

Sometimes

Often

Always

5. How often do you use voice communication while performing actions related to fighting in large groups versus NPCs in instances (raiding)?

Never

Rarely

Sometimes

Often

Always

Appendix B. List of WoW API events across categories

Trading

AUCTION_HOUSE_SHOW

AUCTION_HOUSE_CLOSED

BANKFRAME_CLOSED

BANKFRAME_OPENED

GUILDBANKFRAME_CLOSED

GUILDBANKFRAME_OPENED

BIND_ENCHANT

REPLACE_ENCHANT

MAIL_CLOSED

MAIL_SHOW

TRADE_CLOSED

TRADE_SHOW

CRAFT_SHOW

CRAFT_CLOSE

TRADE_SKILL_SHOW

TRADE_SKILL_CLOSE

MAIL_SEND_SUCCESS

TRADE_SKILL_UPDATE

MERCHANT_CLOSED

MERCHANT_SHOW

Questing

QUEST_ACCEPT_CONFIRM

QUEST_COMPLETE

QUEST_FINISHED

QUEST_WATCH_UPDATE

QUEST_ITEM_UPDATE

PvP combat

DUEL_INBOUNDS

DUEL_FINISHED

DUEL_OUTOFBOUNDS

DUEL_REQUESTED

BATTLEFIELDS_SHOW

BATTLEFIELDS_CLOSED

CHAT_MSG_BG_SYSTEM_ALLIANCE

CHAT_MSG_BG_SYSTEM_HORDE

CHAT_MSG_BG_SYSTEM_NEUTRAL

CHAT_MSG_COMBAT_HONOR_GAIN

UPDATE_BATTLEFIELD_SCORE

UPDATE_BATTLEFIELD_STATUS

ZONE_CHANGED_NEW_AREA + area names (Warsong Gulch, Arathi Basin, Alterac Valley, Eye of the Storm,Strand of the Ancients)

Dungeons

ZONE_CHANGED_NEW_AREA + area names (Ragefire Chasm, Wailing Caverns, The Deadmines, Shadowfang Keep, Blackfathom Deeps, THe Stockade, Razorfen Kraul, Gnomeregan, Scarlet Monastery, Razorfen Downs, Uldaman, Zul'Farrak, Maraudon, The Temple of Atal'Hakkar, Blackrock Depths, Warpwood Quarter, Blackrock Spire, Dire Maul:West, Dire Maul:North, Stratholme, Scholomance, Upper Blackrock Spire, Dire Maul, Stratholme, Scholomance, Hellfire Ramparts, The Blood Furnace, The Slave Pens, The Underbog, Mana-Tombs, Auchenai Crypts, Escape from Durnholde Keep, Sethekk Halls, The Steamvault, Shadow Labyrinth, The Shattered Halls, Black Morass, The Mechanar, The Botanica, The Arcatraz, Magisters' Terrace, Utgarde Keep, The Nexus, Azjol-Nerub, Ahn'kahet: The Old Kingdom, Drak'Tharon Keep, The Violet Hold, Gundrak, Halls of Stone, Halls of Lightning, The Oculus, Culling of Stratholme, Utgarde Pinnacle)

Raiding

ZONE_CHANGED_NEW_AREA + area names (Ulduar, Vault of Archavon, The Eye of Eternity, The Obsidian Sanctum, Obsidian Sanctum, Naxxramas, Sunwell Plateau, Black Temple, Battle for Mount Hyjal, Tempest Keep, Serpentshrine Cavern, Zul'Aman, Magtheridon's Lair, Gruul's Lair, Karazhan, Ahn'Qiraj, Ruins of Ahn'Qiraj, Blackwing Lair, Molten Core, Onyxia's Lair, Zul'Gurub)

Communication

CHAT_MSG_SAY

CHAT_MSG_CHANNEL

CHAT_MSG_EMOTE

CHAT_MSG_GUILD

CHAT_MSG_OFFICER

CHAT_MSG_PARTY

CHAT_MSG_RAID_LEADER

CHAT_MSG_TEXT_EMOTE

CHAT_MSG_WHISPER_INFORM

CHAT_MSG_RAID

CHAT_MSG_YELL

Session start

PLAYER_LOGIN

Session end

PLAYER_QUITING

PLAYER_LOGOUT

Appendix C. The parameters of the behaviour model

There parameters of the Markov chain are presented in the Table 6.1 where the action category stated in the column is the starting state, while the resulting state is in the row. For example, for hour 0:00-1:00 the probability from going from raiding to trading is 0.83 and not 0.02.

Table 6.1: Parameters of the Markov chain across hours of the day

Time		Questing	Trading	PvP combat	Raiding	Dungeons	Uncategorized	Start
0:00-1:00	**Questing**	0.05	0.32	0.30	0.08	0.28	0.20	0.13
	Trading	0.69	0.04	0.39	0.83	0.58	0.63	0.67
	PvP combat	0.11	0.09	0.18	0.04	0.04	0.07	0.07
	Raiding	0.01	0.02	0.02	0.00	0.04	0.04	0.02
	Dungeons	0.03	0.02	0.02	0.00	0.00	0.03	0.01
	Uncategorized	0.12	0.52	0.09	0.05	0.06	0.02	0.09
1:00-2:00	**Questing**	0.04	0.38	0.25	0.00	0.23	0.19	0.20
	Trading	0.75	0.01	0.41	0.86	0.45	0.64	0.62
	PvP combat	0.08	0.09	0.22	0.01	0.05	0.07	0.06
	Raiding	0.00	0.03	0.01	0.01	0.11	0.02	0.02
	Dungeons	0.03	0.02	0.02	0.05	0.00	0.04	0.03
	Uncategorized	0.11	0.47	0.10	0.06	0.16	0.04	0.08
2:00-3:00	**Questing**	0.05	0.34	0.22	0.00	0.08	0.16	0.10
	Trading	0.70	0.02	0.45	0.93	0.67	0.65	0.72
	PvP combat	0.15	0.11	0.21	0.05	0.04	0.13	0.04
	Raiding	0.00	0.04	0.01	0.00	0.04	0.02	0.03
	Dungeons	0.00	0.02	0.01	0.00	0.04	0.01	0.03
	Uncategorized	0.09	0.46	0.11	0.02	0.13	0.03	0.08
3:00-4:00	**Questing**	0.10	0.45	0.46	0.00	0.22	0.30	0.13
	Trading	0.64	0.12	0.19	0.92	0.56	0.45	0.75
	PvP combat	0.16	0.08	0.23	0.08	0.00	0.16	0.03
	Raiding	0.01	0.02	0.00	0.00	0.11	0.03	0.03
	Dungeons	0.02	0.01	0.01	0.00	0.00	0.02	0.00
	Uncategorized	0.08	0.32	0.11	0.00	0.11	0.04	0.08
4:00-5:00	**Questing**	0.11	0.39	0.36	0.17	0.00	0.29	0.22
	Trading	0.68	0.03	0.29	0.75	1.00	0.53	0.61
	PvP combat	0.14	0.12	0.25	0.08	0.00	0.14	0.06
	Raiding	0.01	0.03	0.02	0.00	0.00	0.00	0.00
	Dungeons	0.00	0.00	0.00	0.00	0.00	0.02	0.00
	Uncategorized	0.06	0.42	0.08	0.00	0.00	0.03	0.11

Continued on next page

Table 6.1 – continued from previous page

Time		Questing	Trading	PvP combat	Raiding	Dungeons	Uncategorized	Start
5:00-6:00	**Questing**	0.00	0.50	0.11	0.50	0.00	0.31	0.50
	Trading	0.65	0.03	0.44	0.50	1.00	0.54	0.42
	PvP combat	0.16	0.09	0.22	0.00	0.00	0.00	0.00
	Raiding	0.03	0.00	0.00	0.00	0.00	0.00	0.00
	Dungeons	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Uncategorized	0.16	0.38	0.22	0.00	0.00	0.15	0.08
6:00-7:00	**Questing**	0.00	0.34	0.50	0.00	0.00	0.36	0.25
	Trading	0.81	0.02	0.50	1.00	0.00	0.60	0.69
	PvP combat	0.00	0.10	0.00	0.00	0.00	0.00	0.00
	Raiding	0.00	0.00	0.00	0.00	0.00	0.00	0.06
	Dungeons	0.04	0.00	0.00	0.00	0.00	0.00	0.00
	Uncategorized	0.15	0.54	0.00	0.00	1.00	0.04	0.00
7:00-8:00	**Questing**	0.04	0.61	0.60	0.00	0.00	0.33	0.18
	Trading	0.67	0.02	0.20	0.00	0.00	0.60	0.75
	PvP combat	0.12	0.03	0.20	0.00	0.00	0.07	0.00
	Raiding	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Dungeons	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Uncategorized	0.18	0.34	0.00	0.00	0.00	0.00	0.07
8:00-9:00	**Questing**	0.03	0.57	0.53	0.00	0.00	0.35	0.15
	Trading	0.71	0.06	0.18	1.00	0.00	0.57	0.75
	PvP combat	0.10	0.02	0.12	0.00	0.00	0.09	0.02
	Raiding	0.00	0.01	0.00	0.00	0.00	0.00	0.01
	Dungeons	0.02	0.00	0.00	0.00	0.00	0.00	0.01
	Uncategorized	0.13	0.34	0.18	0.00	0.00	0.00	0.06
9:00-10:00	**Questing**	0.05	0.47	0.35	0.00	0.38	0.40	0.22
	Trading	0.66	0.10	0.24	1.00	0.25	0.46	0.62
	PvP combat	0.18	0.10	0.36	0.00	0.13	0.07	0.06
	Raiding	0.00	0.00	0.00	0.00	0.00	0.01	0.00
	Dungeons	0.02	0.01	0.00	0.00	0.00	0.04	0.01
	Uncategorized	0.10	0.32	0.05	0.00	0.25	0.02	0.09
10:00-11:00	**Questing**	0.08	0.49	0.35	0.50	0.30	0.30	0.23
	Trading	0.60	0.06	0.17	0.17	0.30	0.47	0.64
	PvP combat	0.15	0.06	0.39	0.00	0.00	0.13	0.06
	Raiding	0.01	0.00	0.00	0.00	0.20	0.04	0.01
	Dungeons	0.02	0.02	0.00	0.17	0.00	0.03	0.00
	Uncategorized	0.14	0.37	0.08	0.17	0.20	0.03	0.06
11:00-12:00	**Questing**	0.05	0.43	0.31	0.17	0.37	0.30	0.21
	Trading	0.61	0.04	0.25	0.56	0.48	0.50	0.66
	PvP combat	0.20	0.11	0.35	0.17	0.00	0.10	0.04
	Raiding	0.01	0.01	0.01	0.00	0.04	0.04	0.01
	Dungeons	0.02	0.01	0.00	0.00	0.00	0.03	0.00
	Uncategorized	0.11	0.40	0.08	0.11	0.11	0.04	0.08
12:00-13:00	**Questing**	0.06	0.43	0.33	0.20	0.30	0.23	0.20
	Trading	0.64	0.04	0.19	0.44	0.48	0.55	0.60
	PvP combat	0.17	0.08	0.36	0.20	0.00	0.11	0.09
	Raiding	0.00	0.01	0.01	0.00	0.03	0.03	0.01
	Dungeons	0.02	0.02	0.01	0.04	0.03	0.04	0.01
	Uncategorized	0.11	0.42	0.11	0.12	0.18	0.04	0.08
13:00-14:00	**Questing**	0.03	0.42	0.36	0.10	0.30	0.25	0.19
	Trading	0.69	0.04	0.32	0.74	0.47	0.50	0.63
	PvP combat	0.11	0.09	0.18	0.06	0.02	0.13	0.06
	Raiding	0.01	0.02	0.03	0.00	0.04	0.03	0.01
	Dungeons	0.02	0.02	0.00	0.03	0.04	0.03	0.02
	Uncategorized	0.14	0.41	0.11	0.06	0.13	0.05	0.09
14:00-15:00	**Questing**	0.04	0.38	0.32	0.02	0.22	0.23	0.21
	Trading	0.61	0.04	0.27	0.66	0.44	0.51	0.57
	PvP combat	0.17	0.09	0.30	0.05	0.02	0.11	0.09
	Raiding	0.01	0.03	0.01	0.02	0.27	0.06	0.02
	Dungeons	0.02	0.01	0.00	0.20	0.00	0.05	0.01
	Uncategorized	0.14	0.45	0.10	0.05	0.06	0.05	0.10

Continued on next page

Table 6.1 – continued from previous page

Time		Questing	Trading	PvP combat	Raiding	Dungeons	Uncategorized	Start
15:00-16:00	**Questing**	0.04	0.36	0.33	0.08	0.15	0.23	0.23
	Trading	0.61	0.04	0.25	0.63	0.49	0.53	0.60
	PvP combat	0.19	0.09	0.26	0.03	0.07	0.10	0.06
	Raiding	0.01	0.02	0.02	0.00	0.18	0.04	0.03
	Dungeons	0.02	0.01	0.01	0.15	0.03	0.06	0.02
	Uncategorized	0.14	0.47	0.14	0.11	0.08	0.04	0.07
16:00-17:00	**Questing**	0.03	0.38	0.34	0.09	0.25	0.26	0.19
	Trading	0.69	0.03	0.31	0.82	0.51	0.52	0.66
	PvP combat	0.14	0.09	0.22	0.05	0.06	0.08	0.06
	Raiding	0.01	0.03	0.01	0.00	0.06	0.04	0.02
	Dungeons	0.03	0.01	0.02	0.00	0.01	0.05	0.01
	Uncategorized	0.11	0.46	0.09	0.05	0.10	0.04	0.06
17:00-18:00	**Questing**	0.04	0.36	0.26	0.06	0.28	0.24	0.20
	Trading	0.62	0.02	0.34	0.75	0.49	0.54	0.62
	PvP combat	0.16	0.11	0.27	0.07	0.04	0.11	0.08
	Raiding	0.01	0.03	0.02	0.00	0.13	0.04	0.02
	Dungeons	0.02	0.01	0.01	0.02	0.00	0.03	0.01
	Uncategorized	0.15	0.46	0.09	0.09	0.06	0.04	0.07
18:00-19:00	**Questing**	0.05	0.30	0.33	0.04	0.19	0.18	0.16
	Trading	0.59	0.07	0.23	0.78	0.42	0.54	0.60
	PvP combat	0.16	0.07	0.28	0.05	0.04	0.09	0.07
	Raiding	0.03	0.05	0.03	0.00	0.12	0.11	0.05
	Dungeons	0.02	0.01	0.01	0.03	0.01	0.04	0.01
	Uncategorized	0.15	0.50	0.12	0.10	0.21	0.04	0.12
19:00-20:00	**Questing**	0.03	0.26	0.24	0.03	0.17	0.17	0.13
	Trading	0.62	0.03	0.30	0.66	0.57	0.47	0.62
	PvP combat	0.18	0.10	0.26	0.19	0.02	0.09	0.08
	Raiding	0.02	0.06	0.07	0.01	0.16	0.19	0.10
	Dungeons	0.02	0.01	0.01	0.04	0.00	0.04	0.01
	Uncategorized	0.13	0.54	0.13	0.06	0.09	0.05	0.06
20:00-21:00	**Questing**	0.07	0.27	0.24	0.05	0.22	0.16	0.17
	Trading	0.58	0.02	0.28	0.60	0.54	0.48	0.56
	PvP combat	0.16	0.10	0.24	0.27	0.04	0.12	0.09
	Raiding	0.01	0.10	0.10	0.01	0.13	0.17	0.08
	Dungeons	0.02	0.02	0.00	0.02	0.00	0.03	0.01
	Uncategorized	0.15	0.49	0.13	0.05	0.06	0.05	0.08
21:00-22:00	**Questing**	0.04	0.39	0.26	0.03	0.39	0.24	0.20
	Trading	0.65	0.04	0.29	0.69	0.45	0.46	0.59
	PvP combat	0.14	0.10	0.20	0.20	0.03	0.11	0.11
	Raiding	0.01	0.06	0.12	0.00	0.04	0.09	0.05
	Dungeons	0.03	0.02	0.02	0.02	0.00	0.04	0.00
	Uncategorized	0.13	0.40	0.12	0.07	0.09	0.06	0.05
22:00-23:00	**Questing**	0.05	0.36	0.27	0.04	0.19	0.23	0.20
	Trading	0.63	0.01	0.26	0.73	0.61	0.51	0.54
	PvP combat	0.15	0.10	0.27	0.16	0.00	0.11	0.09
	Raiding	0.01	0.05	0.10	0.00	0.07	0.06	0.07
	Dungeons	0.02	0.02	0.02	0.01	0.00	0.05	0.01
	Uncategorized	0.13	0.46	0.09	0.06	0.14	0.04	0.08
23:00-24:00	**Questing**	0.02	0.33	0.30	0.05	0.18	0.18	0.19
	Trading	0.72	0.04	0.45	0.87	0.54	0.56	0.57
	PvP combat	0.10	0.10	0.08	0.03	0.03	0.11	0.07
	Raiding	0.01	0.04	0.03	0.01	0.10	0.05	0.04
	Dungeons	0.02	0.01	0.01	0.00	0.00	0.07	0.03
	Uncategorized	0.13	0.48	0.13	0.04	0.15	0.04	0.09

Bibliography

[1] A. Botta, A. Dainotti, and A. Pescapè, "Multi-protocol and Multi-platform Traffic Generation and Measurement," in *INFOCOM 2007, 26th IEEE International Conference on Computer Communications, Demonstration Session*, 2007.

[2] H. Park, T. Kim, and S. Kim, "Network traffic analysis and modeling for games," in *Internet and Network Economics*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, pp. 1056–1065.

[3] Entertainment Software Association. (2011, May) Essential facts about the computer and video game industry. [Online]. Available: http://www.theesa.com/facts/pdfs/ESA_EF_2011.pdf

[4] NewZoo, "National Gamers Survey 2010 | USA," Tech. Rep., June 2010. [Online]. Available: http://www.newzoo.com/press/NationalGamersSurvey2010_Summary_US.pdf

[5] KPMG, "The video games market in China: Moving Online," Tech. Rep., July 2008. [Online]. Available: http://www.kpmg.com/CN/en/IssuesAndInsights/ArticlesPublications/Documents/video_games_china_0712.pdf

[6] Cisco Systems. (2011, June) Cisco Visual Networking Index: Forecast and methodology, 2010-2015. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[7] Virgin Media. (2010, Nov.) World of Warcraft latency issues. [Online]. Available: http://community.virginmedia.com/t5/Broadband-down-your-phone-line/World-of-Warcraft-Latency-Issues/m-p/161857

[8] M. Suznjevic, O. Dobrijevic, and M. Matijasevic, "MMORPG player actions: Network performance, session patterns and latency requirements analysis," *Multimedia Tools and Applications*, vol. 45, no. 1-3, pp. 191–241, 2009.

[9] M. Suznjevic, M. Matijasevic, and O. Dobrijevic, "Action specific Massive Multiplayer Online Role Playing Games traffic analysis: Case study of World of Warcraft," in *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, 2008, pp. 106–107.

[10] X. Wang, H. Kim, A. V. Vasilakos, T. T. Kwon, Y. Choi, S. Choi, and H. Jang, "Measurement and Analysis of World of Warcraft in Mobile WiMAX networks," in *Proceedings of the 8th Workshop on Network and System Support for Games*, 2009, p. 6.

[11] K.-T. Chen and C.-L. Lei, "Network game design: Hints and implications of player interaction," in *Proceedings of the 5th ACM SIGCOMM Workshop on Network and System Support for Games*, 2006, pp. 1–9.

[12] Wikipedia, the free encyclopedia. (2011, June) Massively multiplayer online role-playing game. [Online]. Available: http://en.wikipedia.org/wiki/Massively_multiplayer_online_role-playing_game

[13] S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*. Addison-Wesley Professional, 1999.

[14] E. Castronova, *Synthetic Worlds: The Business and Culture of Online Games*. University Chicago Press, 2006.

[15] Y.-T. Lee, K.-T. Chen, Y.-M. Cheng, and C.-L. Lei, "World of Warcraft avatar history dataset," in *Proceedings of the second annual ACM conference on Multimedia systems*, 2011, pp. 123–128.

[16] B. Drain. (2011, Jan.) World of tanks hits 74,536 peak concurrent users on one server. [Online]. Available: http://massively.joystiq.com/2011/01/06/world-of-tanks-hits-74-536-peak-concurrent-users-on-one-server/

[17] J. Wallis. (2008, June) n00b world reorder. [Online]. Available: http://www.spaaace.com/cope/?p=111

[18] D. Massey. (2007, Feb.) Lord of the Rings online: Ask Turbine #2: Exclusive interview with Jeff Anderson. [Online]. Available: http://www.warcry.com/articles/view/interviews/askturbine/4647-Lord-Of-The-Rings-Online-Ask-Turbine-2-/Exclusive-Interview-with-Jeff-Anderson

[19] Wikipedia, the free encyclopedia. (2011, June) Dungeons and dragons. [Online]. Available: http://en.wikipedia.org/wiki/Dungeons_and_Dragons

[20] T. Pace, "Can an Orc catch a cab in Stormwind?: Cybertype preference in the World of Warcraft character creation interface," in *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, 2008, pp. 2493–2502.

[21] D. Grundy, " The Effect of Dependence on Chi-Squared and Empiric Distribution Tests of Fit," *Games and Culture*, vol. 3, no. 2, pp. 225–247, 2008.

[22] N. Ducheneaut, N. Yee, E. Nickell, and R. Moore, "The life and death of online gaming communities: a look at guilds in World of Warcraft," in *CHI '07: Proceedings of the SIGCHI Conference on Human factors in computing systems*, 2007, pp. 839–848.

[23] Wikipedia, the free encyclopedia. (2011, June) History of massively multiplayer online games. [Online]. Available: http://en.wikipedia.org/wiki/History_of_massively_multiplayer_online_role-playing_games

[24] L. Achterbosch, R. Pierce, and G. Simmons, "Massively multiplayer online role-playing games: the past, present, and future," *Computer Entertainment*, vol. 5, no. 4, pp. 1–33, 2007.

[25] S. Bozcan, "A tool for network simulation of massively multiplayer online games," Master thesis, Middle East Technical University, Universiteler Mahallesi, Dumlupinar Bulvari No:1, 06800 Cankaya Ankara, Turkey, Sept. 2008.

[26] S. Butts. (2007, Jan.) Ultima online: Kingdom reborn. [Online]. Available: http://pc.ign.com/articles/759/759391p1.html

[27] B. Drain. (2010, June) Eve pushes over the 60,000 peak concurrent user mark. [Online]. Available: http://massively.joystiq.com/2010/06/07/eve-pushes-over-the-60-000-peak-concurrent-user-mark/

[28] Activision Blizzard. (2010, Nov.) World of Warcraft subscriber base reaches 12 million worldwide. [Online]. Available: http://us.blizzard.com/en-us/company/press/pressreleases.html?101007

[29] Wikipedia, the free encyclopedia. (2011, June) Runescape. [Online]. Available: http://en.wikipedia.org/wiki/RuneScape

[30] L. Fan, P. Trinder, and H. Taylor, "Design issues for peer to peer massively multiplayer online games," *Int. J. Adv. Media Commun.*, vol. 4, pp. 108–125, 2010.

[31] G. Huang, M. Ye, and L. Cheng, "Modeling system performance in mmorpg," in *Global Telecommunications Conference Workshops*, 2004, pp. 512 – 518.

[32] WoWWiki, the free Warcraft universe info source. (2011, June) Wow. [Online]. Available: http://www.wowwiki.com/System_requirements/

[33] R. Prodan and V. Naea, "Prediction-based real-time resource provisioning for massively multiplayer online games," *Future Generation Computer Systems*, vol. 25, pp. 785–793, 2009.

[34] R. D. P. McFarlane, "Network software architectures for real-time massively-multiplayer online games," Master thesis, School of Computer Science, McGill University, Montreal, Quebec, Canada, Feb. 2005.

[35] J. Muller, J. H. Metzen, A. Ploss, M. Schellmann, and S. Gorlatch, “Rokkatan: Scaling an rts game design to the massively multiplayer realm,” *Future Generation Computer Systems*, vol. 4, 2006.

[36] N. Yee, “The demographics, motivations and Derived Experiences of Users of Massively-Multiuser Online Graphical Environments,” in *PRESENCE: Teleoperators and Virtual Envireonments*, 2006, pp. 309–329.

[37] R. Bartle, “Hearts, clubs, diamonds, spades: Players who suit MUDs,” *The Journal of Virtual Environments*, vol. 1, 1996.

[38] R. Bartle, *Designing Virtual Worlds*. New Riders, 2003.

[39] N. Yee, “Experimental Motives for Playing Online Games,” *Journal of CyberPsychology and Behavior*, vol. 9, no. 6, pp. 772–775, 2007.

[40] R. van Meurs, “How to play the game: a study on MUD player types and their real life personality traits,” Leisure Studies, Tilburg University, Tilburg University, 5000 LE Tilburg, Netherlands, 2007. [Online]. Available: http://www.rofv.nl/survey/

[41] F. You, J. Wang, Z. Zheng, J. Jung, C. Che, X. Luo, and P. Tam, “RA-model: a taxonomy model of player activities in mobile MMORPGs,” in *Mobility ’07: Proceedings of the 4th International Conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*, 2007, pp. 704–707.

[42] E. M. I. Koivisto and C. Wenninger, “Enhancing player experience in MMORPGs with mobile features,” in *In International Digital Games Research Association Conference*, 2005.

[43] S. Wu. (2007, June) Virtual goods: the next big business model. [Online]. Available: http://techcrunch.com/2007/06/20/virtual-goods-the-next-big-business-model/

[44] E. Caoili. (2010, Apr.) Over 140,000 Users Queue For WoW's Virtual Items. [Online]. Available: http://www.gamasutra.com/view/news/28129/Over_140000_Users_Queue_For_WoWs_Virtual_Items.php

[45] B. Choi, I. Lee, K. Lee, S. Jung, S. Park, and J. Kim, "The Effects of Users' Motivation on their Perception to Trading Systems of Digital Content Accessories: Focusing on Trading Items in Online Games," in *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, 2007, pp. 161–161.

[46] I. V. Geel. (2010, Nov.) Mmodata charts v3.1. [Online]. Available: http://mmodata.net/

[47] N. Yee. (2009, Mar.) The daedalus project - survey study of mmo players. [Online]. Available: http://www.nickyee.com/daedalus/

[48] N. Yee, "Maps of Digital Desires: Exploring the Topography of Gender and Play in Online Games," in *Beyond Barbie and Mortal Kombat: New Perspectives on Gender and Gaming*. Cambridge: MIT Press, 2008, pp. 83–96.

[49] D. Williams, N. Yee, and S. E. Caplan, "Who plays, how much, and why? Debunking the stereotypical gamer profile," *Journal of Computer-Mediated communication*, vol. 13, no. 1, pp. 993–1018, Nov. 2008.

[50] K.-T. Chen, P. Huang, and C.-L. Lei, "Game Traffic Analysis: An MMORPG Perspective," *Computer Networks*, vol. 51, no. 3, pp. 19–24, 2007.

[51] K.-T. Chen, P. Huang, G.-S. Wang, C.-Y. Huang, and C.-L. Lei, "On the sensitivity of online game playing time to network QoS," in *Proceedings of the IEEE International Conference on Computer Communications INFOCOM'06*, 2006, pp. 1 – 12.

[52] K.-T. Chen, P. Huang, and C.-L. Lei, "How sensitive are online gamers to network quality?" *Communications of the ACM*, vol. 49, no. 11, pp. 34–38, 2006.

[53] W.-C. Feng, D. Brandt, and D. Saha, "A long-term study of popular MMORPG," in *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, 2007, pp. 19–24.

[54] D. Pittman and C. GauthierDickey, "A Measurement Study of Virtual Populations in Massively Multiplayer Online Games," in *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, 2007, pp. 25–30.

[55] D. Pittman and C. G. Dickey, "Characterizing Virtual Populations in Massively Multiplayer Online Role-Playing Games," in *Advances in Multimedia Modeling*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 5916, pp. 87–97.

[56] X. Zhuang, A. Bharambe, J. Pang, and S. Seshan, "Player Dynamics in Massively Multiplayer Online Games," School of Computer Science, Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-CS-07-158, 2007.

[57] P.-Y. Tarng, K.-T. Chen, and P. Huang, "An analysis of WoW players' game hours," in *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, 2008, pp. 47–52.

[58] Y.-T. Lee and K.-T. Chen, "Is server consolidation beneficial to MMORPG? A case study of World of Warcraft," in *IEEE International Conference on Cloud Computing*, 2010, pp. 435–442.

[59] P. Svoboda, W. Karner, and M. Rupp, "Traffic analysis and modeling for World of Warcraft," in *Communications, 2007. ICC '07. IEEE International Conference on*, 2007, pp. 1612–1617.

[60] M. Suznjevic, O. Dobrijevic, and M. Matijasevic, "Hack, slash, and chat: A study of players' behavior and communication in MMORPGs," in *Proceedings of the 8th Workshop on Network and System Support for Games*, 2009, p. 6.

[61] J. Kawale, A. Pal, and J. Srivastava, "Churn prediction in MMORPGs: A social influence based approach," in *Proceedings of the International Conference on Computational Science and Engineering*, 2009, pp. 423–428.

[62] M. Kwok and G. Yeung, "Characterization of user behavior in a multi-player online game," in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2005, p. 6.

[63] H. Liang, R. N. Silva, W. T. Ooi, and M. Motani, "Avatar mobility in user-created networked virtual worlds: measurements, analysis, and implications," *Multimedia Tools and Applications*, vol. 45, no. 1-3, pp. 163–190, 2009.

[64] S. Fernandes, R. Antonello, J. Moreira, D. Sadok, and C. Kamienski, "Traffic Analysis Beyond This World: the Case of Second Life," in *Proceedings of the 17th International workshop on Network and Operating Systems Support for Digital Audio & Video (NOSSDAV)*, 2007.

[65] M. Kihl, A. Aurelius, and C. Lagerstedt, "Analysis of World of Warcraft Traffic Patterns and User behavior," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, 2010, pp. 218–223.

[66] J. L. Miller and J. Crowcroft, "Avatar movement in World of Warcraft battlegrounds," in *Proceedings of the 8th Annual Workshop on Network and Systems Support for Games*, 2009, pp. 1:1–1:6.

[67] J. Miller and J. Crowcroft, "Group movement in World of Warcraft Battlegrounds," *Int. J. Adv. Media Commun.*, vol. 4, pp. 387–404, 2010.

[68] H. Backhaus and S. Krause, "Voronoi-based adaptive scalable transfer revisited: gain and loss of a Voronoi-based peer-to-peer approach for MMOG," in *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, 2007, pp. 49–54.

[69] M. Varvello, E. Biersack, and C. Diot, "Dynamic clustering in Delaunay-based P2P networked virtual environments," in *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, 2007, pp. 105–110.

[70] A. Sinha, K. Mitchell, and D. Medhi, "Network Game Traffic: A Broadband Access Perspective," *Computer Networks*, vol. 49, no. 1, pp. 71–83, 2004.

[71] G. Box, G. M. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Controls*. New York: Prentice-Hall, 1994.

[72] T. Henderson and S. Bhatti, "Modelling user behaviour in networked games," in *Proceedings of ACM Multimedia'01*, 2001, pp. 212–220.

[73] I. V. Geel. (2011, June) Tracking of active MMOG subscriptions. [Online]. Available: www.mmodata.net

[74] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore, "'Alone together?' Exploring the social dynamics of massively multiplayer online games," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2006, pp. 407 – 416.

[75] B. Nardi and J. Harris, "Strangers and friends: collaborative play in World of Warcraft," in *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, 2006, pp. 149–158.

[76] S. Ratti, B. Hariri, and S. Shirmohammadi, "A Survey of First-Person Shooter Gaming Traffic on the Internet," *IEEE Internet Computing*, vol. 14, no. 5, pp. 60–69, 2010.

[77] H. E. Hurst, "Methods of using long-term storage in reservoirs," *Proceedings of the Institution of Civil Engineers, Part I*, pp. 519–577, 1955.

[78] P. Danzig and S. Jamin, "tcplib: A Library of TCP Internetwork Traffic Characteristics," Computer Science Department, University of Southern California, Tech. Rep. CS-SYS-91-01, 1991.

[79] V. Paxson, "Empirically-Derived Analytic Models of Wide-Area TCP Connections," *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 316–336, 1994.

[80] M. S. Borella, "Source models of network game traffic," *Computer Communications*, vol. 23, no. 4, pp. 403–410, 2000.

[81] J. Lakkakorpi, A. Heiner, and J. Ruutuc, "Measurement and characterization of Internet gaming traffic," Espoo, Tech. Rep., Feb. 2002, Research Seminar on Networking.

[82] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979.

[83] S. P. Pederson and M. E. Johnson, "Estimating model discrepancy," *Technometrics*, vol. 32, no. 3, pp. 305–314, 1990.

[84] 3rd generation partnership project 2 3GPP2. (2004, dec) cdma2000 evaluation methodology 3gpp2 c.r1002-0. [Online]. Available: http://www.3gpp2.org/public_html/specs/C.R1002-0_v1.0_041221.pdf

[85] T. Lang, P. Branch, and G. Armitage, "A synthetic traffic model for Quake3," in *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2004, pp. 233–238.

[86] T. Lang and G. Armitage, "An ns2 model for the Xbox system link game Halo," Swinburne University of Technology, Faculty of Information and Communication Technologies, Center for Advanced Internet Architectures, Tech. Rep. 030613A, 2003.

[87] T. Lang, G. Armitage, P. Branch, and H.-Y. Choo, "A synthetic traffic model for Half-Life," in *in Australian Telecommunications, Networks and Applications Conference (ATNAC*, 2003.

[88] J. Färber, "Network game traffic modelling," in *NetGames '02: Proceedings of the 1st workshop on Network and system support for games*, 2002, pp. 53–57.

[89] A. Dainotti, A. Pescapé, and G. Ventre, "A packet-level traffic model of Starcraft," in *HOT-P2P '05: Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems*, 2005, pp. 33–42.

[90] A. Cricenti and P. Branch, "ARMA(1,1) modeling of Quake4 server to client game traffic," in *NetGames '07: Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, 2007, pp. 70–74.

[91] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "An empirical evaluation of TCP performance in online games," in *ACE '06: Proceedings of the 2006 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2006, p. 5.

[92] Y. Wu, H. Huang, and D. Zhang, "Traffic Modeling for Massive Multiplayer On-line Role Playing Game (MMORPG) in GPRS Access Network," in *Proceedings of the International Conference on Communications, Circuits and Systems Proceedings*, 2006, pp. 1811–1815.

[93] Y.-T. Han and H.-S. Park, "Game Traffic Classification Using Statistical Characteristics and the Transport Layer," *Electronics and Telecommunications Research Institute*, vol. 32, no. 1, pp. 22–32, 2010.

[94] J. Kim, E. Hong, and J. Choi, "Measurement and Analysis of a Massively Multiplayer Online Role Playing Game Traffic," in *Proceedings of Advanced Network Conference*, 2003, pp. 1–8.

[95] J. Kim, J. Choi, D. Chang, T. Kwon, Y. Choi, and E. Yuk, "Traffic characteristics of a Massively Multiplayer Online Role Playing Game," in *Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support for Games*, Oct. 2005, pp. 1–8.

[96] C. Griwodz and P. Halorsen, "The fun of using TCP for a MMORPG," in *Proceedings of the 2006 International workshop on Network and operating systems support for digital audio and video*, 2006, pp. 1:1–1:7.

[97] K.-T. Chen, J.-W. Jiang, P. Huang, H.-H. Chu, C.-L. Lei, and W.-C. Chen, "Identifying MMORPG Bots: A Traffic Analysis Approach," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 3:1–3:22, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1155/2009/797159

[98] G. Szabó, A. Veres, and S. Molnár, "Effects of user behavior on MMORPG traffic," in *Proceedings of the 2009 IEEE International Conference on Communications*, 2009, pp. 1531–1536.

[99] G.Szabó, A. Veres, and S. Molnár, "On the impacts of human interactions in MMORPG traffic," *Multimedia Tools and Applications*, vol. 45, no. 1-3, pp. 133–161, 2009.

[100] M. Suznjevic, I. Stupar, and M. Matijasevic, “Traffic modeling of player action categories in a MMORPG,” in *Proceedings of the 2nd workshop on DIstributed SImulation and Online gaming (DISIO)*, 2011, p. 8.

[101] I. A. Oliver, A. H. Miller, and C. Allison, “Virtual worlds, real traffic: interaction and adaptation,” in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, 2010, pp. 305–316.

[102] J. Kinicki and M. Claypool, “Traffic analysis of avatars in Second Life,” in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2008, pp. 69–74.

[103] R. Antonello, S. Fernandes, J. Moreira, P. Cunha, C. Kamienski, and D. Sadok, “Traffic analysis and synthetic models of Second Life,” *Multimedia Tools and Applications*, vol. 15, no. 1, pp. 33–47, 2009.

[104] (2011, June) Shen Zhou Online. [Online]. Available: http://www.ewsoft.com.tw/

[105] C.-C. Wu, K.-T. Chen, C.-M. Chen, P. Huang, and C.-L. Lei, “On the challenge and design of transport protocols for MMORPGs,” *Multimedia Tools and Applications*, vol. 45, no. 1–3, pp. 7–32, 2009.

[106] (2011, June) World of Warcraft. [Online]. Available: http://www.battle.net/wow/

[107] A. Hilven and A. Woodward, “How safe is Azeroth, or, are MMORPGs a security risk?” in *Proceedings of the 5th Australian Information Security management Conference*, 2007, pp. 87–95.

[108] K. Shin, J. Kim, K. Sohn, C. J. Park, and S. Choi, “Transformation Approach to Model Online Gaming Traffic,” *ETRI Journal*, vol. 33, no. 2, pp. 219–229, 2011.

[109] (2011, June) Left 4 Dead. [Online]. Available: http://www.l4d.com/blog/

[110] K. Shin, J. Kim, K. Sohn, C. Park, and S. Choi, “Online gaming traffic generator for reproducing gamer behavior,” in *Proceedings of the 9th international conference on Entertainment computing*, 2010, pp. 160–170.

[111] (2011, June) Lineage II. [Online]. Available: http://www.lineage2.com/

[112] (2011, June) World of Legend. [Online]. Available: http://woool.sdo.com/home/homepage.htm

[113] (2011, Jun) Anarchy Online. [Online]. Available: http://www.anarchy-online.com/

[114] (2011, June) Ragnarok Online. [Online]. Available: http://www.ragnarokonline.com/

[115] J. Sommers, H. Kim, and P. Barford. (2006, Dec.) Harpoon: a flow-level traffic generator. [Online]. Available: http://pages.cs.wisc.edu/~jsommers/harpoon/

[116] J. Sommers, H. Kim, and P. Barford, "Harpoon: a flow-level traffic generator for router and network tests," *SIGMETRICS Performance Evaluation Review*, vol. 32, no. 1, pp. 392–392, 2004.

[117] NRL PROTocol Engineering Advanced Networking (PROTEAN) Research Group. (2011, June) Multi-generator (mgen). [Online]. Available: http://cs.itd.nrl.navy.mil/work/mgen/index.php

[118] J. Laine, S. Saaristo, and R. Prior. (2002, Sept.) Rude real-time udp data emitter and crude collector for rude. [Online]. Available: http://rude.sourceforge.net/

[119] W. Adamson. Udpgen. [Online]. Available: http://www.citi.umich.edu/projects/qbone/generator.html

[120] R. Sandilands. (2009, Dec.) Network traffic generator. [Online]. Available: http://robert.rsa3.com/traffic.html

[121] C. Krasic. (2002, Feb.) Network traffic generator. [Online]. Available: http://mxtraf.sourceforge.net/

[122] Y. Bachar, O. Ovadia, , G. Netapov, and D. I. David. (2002) Ntgen. [Online]. Available: http://softlab-pro-web.technion.ac.il/projects/NTGen/html/Description.htm

[123] R. Jones. (2009, June) Netpref. [Online]. Available: http://www.netperf.org/netperf/

[124] SRI International and University of Southern California. (2008, Sept.) Traffig generator. [Online]. Available: http://www.postel.org/tg/tg.htm

[125] Anonymous. Traffig generator. [Online]. Available: http://www.st.rim.or.jp/~yumo/pub/tfgen.html

[126] G. Kramer. (2000, Sept.) Generator of self-similar network traffic. [Online]. Available: http://wwwcsif.cs.ucdavis.edu/~kramer/code/trf_gen1.html

[127] P. Heegaard. (2011, Apr.) Gensyn - generator of synthetic internet traffic. [Online]. Available: http://www.item.ntnu.no/people/personalpages/fac/poulh/gensyn

[128] P. E. Heegaard, "Gensyn - a java based generator of synthetic internet traffic linking user behaviour models to real network protocols," in *ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, 2000.

[129] V. Ribeiro, R. King, and N. Hoven. (2003, June) Poisson traffic generator. [Online]. Available: http://www.spin.rice.edu/Software/poisson_gen/

[130] R. Jonkman. Netspec. [Online]. Available: http://www.ittc.ku.edu/netspec/

[131] J. Manner. (2006, Jan.) Jugi's traffic generator. [Online]. Available: http://www.cs.helsinki.fi/u/jmanner/software/jtg/

[132] P. Vincet. Trafgen. [Online]. Available: http://www-lor.int-evry.fr/~vincent/java/trafGen/trafGenEn.htm

[133] S. Zander, D. Kennedy, and G. Armitage. (2007, Mar.) Kute – kernel-based traffic engine. [Online]. Available: http://caia.swin.edu.au/genius/tools/kute/

[134] P. Srivats. (2011, May) Ostinato packet/traffic generator and analyzer. [Online]. Available: http://code.google.com/p/ostinato/#Introduction

[135] M. Matijasevic, D. Gracanin, K. P. Valavanis, and I. Lovrek, "A framework for multiuser distributed virtual environments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, pp. 416–429, 2002.

[136] Wikipedia, the free encyclopedia. (2011, Jun) Virtual economy. [Online]. Available: http://en.wikipedia.org/wiki/Virtual_economy

[137] WoWWiki, the free Warcraft universe info source. (2011, June) Lady vashj (tactics). [Online]. Available: http://www.wowwiki.com/Lady_Vashj_%28tactics%29

[138] M. Busse, B. Lamparter, M. Mauve, and W. Effelsberg, "Lightweight qos-support for networked mobile gaming," in *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*, ser. NetGames '04, 2004, pp. 85–92.

[139] M. Maier and M. Herzog, "Online gaming and P2P file sharing in next-generation EPONs," *IEEE Communications Magazine*, vol. 48, no. 2, pp. 48 – 55, 2002.

[140] M. Suznjevic. (2011, Mar.) WoW add-on for measuring player behavior. [Online]. Available: http://www.fer.hr/tel/en/research/research_groups/netmedia/software

[141] (2011, Feb.) World of Warcraft ace3 addon development framework. [Online]. Available: http://www.wowace.com/addons/ace3/

[142] WoWWiki, the free Warcraft universe info source. (2011, Feb.) World of Warcraft API. World of Warcraft wiki pages. Activision Blizzard. [Online]. Available: http://www.wowwiki.com/World_of_Warcraft_API

[143] N. Yee. (2011, Jun) Motivation assessment. [Online]. Available: http://nickyee.com/daedalus/assess/motsurvey1.php

[144] WoWWiki, the free Warcraft universe info source. (2011, June) WoW addon library for higher time precision. [Online]. Available: http://wowwiki.com/GameTime:Get

[145] J. Simic, "Visualisation of information regarding player behavior in networked virtual envirnment (in Croatian)," Diploma thesis, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia, Jan. 2011.

[146] M. Suznjevic and M. Matijasevic, "Why MMORPG players do what they do: relating motivations to action categories," *International Journal of Advanced Media and Communication*, vol. 4, no. 4, pp. 405–424, 2009.

[147] L. J. Gleser and D. S. Moore, " The Effect of Dependence on Chi-Squared and Empiric Distribution Tests of Fit," *The Annals of Statistics*, vol. 11, no. 4, pp. 1100–1108, 1983.

[148] J.-E. Tyvand, K. Begnum, and H. Hammer, "Deja vu - Predicting the number of players in online games through normalization of historical data," in *Proceedings of 10th workshop on Network and system support for games*, ser. NetGames '11, 2011, p. 2.

[149] (2010, Nov.) Distributed Internet Traffic Generator. [Online]. Available: http://www.grid.unina.it/software/ITG/

[150] S. Avallone, S. Guadagno, D. Emma, A. Pescapé, and G. Venturi, "D-ITG Distributed Internet Traffic Generator," in *Proceeding of International Conference on Quantitative Evaluation of Systems*, 2004, pp. 316–317.

[151] A. Botta, A. Dainotti, and A. Pescapé, "Do you trust your software-based traffic generator?" *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 158 – 165, 2004.

[152] (2012, Jan.) Linux containers. [Online]. Available: http://lxc.sourceforge.net/

[153] (2012, Jan.) Linux bridge. [Online]. Available: http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge/

Biography

Mirko Sužnjević was born on the 31st of January, 1983, in Karlovac, Croatia. He completed high school in Glina, and enrolled at the University of Zagreb, Faculty of Electrical Engineering and Computing (FER), in 2001. He graduated with a Diploma in Electrical Engineering (Dipl. Ing.) degree on the 19th of September, 2006, majoring in Telecommunication and Informatics. He graduated in the top 5% students in course of study Telecommunication and Informatics. His diploma thesis topic was entitled "Creating dynamically adaptable networked virtual environment" (supervisor Maja Matijasevic). As an undergraduate student, he participated in the workshop "Summer Camp" 2005 (Ericsson Nikola Tesla and FER) on the topic of "Exploring ICT Frontiers: Agents, IP Multimedia Subsystem and Distributed Computing". He also participated in the workshop "eNTERFACE'06" in 2006, which was organized in the collaboration between FER and the SIMILAR Network of Excellence, and funded by the Sixth Framework Programme of the European Union, where he worked on the project entitled "Introducing Network-Awareness for Networked Multimedia and Multi-modal Applications".

After graduation he was employed at FER, as a research assistant at the Department of Telecommunications and Informatics. Since 2006, he has been participating in the scientific project of the Ministry of Science, Education and Sports of Republic of Croatia: "Content Delivery and Mobility of Users and Services in New Generation Networks" (036-0362027-1639). He has also been working on five collaborative research projects in cooperation of FER and Ericsson Nikola Tesla R&D Centre: "Session-level Signalling for Advanced Multimedia", "Future Advanced Multimedia Service Enablers", "Session-level Signaling for Advanced Multimedia", "e-Health Research", and "Policy Based Resource Control for Complex IP Networks". Since October 2011, he has been employed as a senior researcher at FER, and funded by the EU FP7 project "ACROSS - Center of Research Excellence for Advanced Cooperative Systems" (Grant no. 285939). In the scope of his work at the university, he was also involved as a teaching assistant in preparation, organization, and execution of laboratory exercises in the following courses: “Multimedia communications”, “Telematic services”, and Seminar C++ (within course “Digital automata” and “Information, logic, languages”).

During his PhD studies, he attended three summer/winter doctoral schools: “IP Traffic Characterization and Anomaly Detection”, in February 2010 in Torino, Italy, organized by the COST Action IC0703 Data Traffic Monitoring and Analysis (TMA) and Euro-NF European network of excellence; “Main Trends in Teletraffic Modelling”, in March 2011 in Valencia, Spain, organized by Euro-NF European network of excellence; and, “Bridging the gap between theory and practice in network monitoring and analysis”, in June 2011, in Naples, Italy, organized by the COST TMA.

As author and co-author, he published ten papers; four of which in international journals and six in conference proceedings. He is a member of the IEEE. He is fluent in English, and has a basic knowledge of German.

Životopis

Mirko Sužnjević rođen je 31.01.1983. u Karlovcu. Gimnaziju je završio u Glini, a visokoškolsko obrazovanje stekao je na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu (FER). Diplomirao je 19. rujna 2006. na smjeru Telekomunikacije i informatika u režimu studija s naglaskom na znanstvenoistraživačkom radu s diplomskim radom pod naslovom "Oblikovanje dinamički prilagodljive usluge mrežnog virtualnog okruženja" (mentor prof.dr.sc. Maja Matijašević). Diplomirao je unutar 5% najboljih studenata na smjeru.

Zaposlio se na FER-u na radnom mjestu zavodskog suradnika u Zavodu za telekomunikacije, gdje i danas radi. Na istom fakultetu upisuje i doktorski studij. Sudjeluje na znanstvenom projektu Ministarstva znanosti, obrazovanja i športa Republike Hrvatske, "Isporuka sadržaja i pokretljivost korisnika i usluga u mrežama nove generacije" (036-0362027-1639). Od 2006. godine do danas, sudjeluje i u nizu istraživačkih projekata koji se odvijaju u suradnji Fakulteta s istraživačko-razvojnim odjelom tvrtke Ericsson Nikola Tesla: "Session-level Signaling for Advanced Multimedia", "Future Advanced Multimedia Service Enablers", "Session-level Signalling for Advanced Multimedia", "e-Health Research" i "Policy Based Resource Control for Complex IP Networks". Od listopada 2011. godine zaposlen je kao istraživač na projektu "ACROSS - Center of Research Excellence for Advanced Cooperative Systems" (285939), u okviru sedmog okvirnog programa Europske Unije (FP7).

U nastavnom radu, Mirko Sužnjević sudjeluje u pripremi, organizaciji i provedbi laboratorijskih vježbi na predmetima "Višemedijske komunikacije", "Telematičke usluge", te Seminar C++ (u okviru predmeta "Digitalni automati" i "Informacija, logika, jezici").

Tijekom diplomskog studija sudjelovao je u radionici "Summer Camp 2005" (Ericsson Nikola Tesla i FER), s temom "Exploring ICT Frontiers: Agents, IP Multimedia Subsystem and Distributed Computing". U 2006. godini sudjeluje u radionici "eNTERFACE'06", organiziranoj u suradnji Fakulteta s mrežom istraživačkih centara SIMILAR, a potpomognutoj od strane šestog okvirnog programa (FP6) Europske Unije, gdje je radio na projektu "Uvođenje mrežne svjesnosti za umrežene višemedijske i višemodalne usluge".

Tijekom doktorskog studija pohađao je nekoliko doktorskih škola: "IP Traffic Characterization and Anomaly Detection", održanoj u veljači 2010. u Torinu (Italija), u organizaciji COST Action IC0703 Data Traffic Monitoring and Analysis (TMA) i Euro-NF evropske mreže izvrsnosti, "Main Trends in Teletraffic Modelling", održanoj u ožujku 2011. u Valenciji (Španjolska), u organizaciji Euro-NF europske mreže izvrsnosti, te "Bridging the gap between theory and practice in network monitoring and analysis" održanoj u svibnju 2011. u Napulju (Italija), u organizaciji COST TMA.

Kao autor i koautor objavio je deset znanstvenih radova, od kojih četiri u međunarodnim časopisima, a šest u zbornicima međunarodnih konferencija. Član je međunarodnog udruženja IEEE. Izvrsno govori i piše engleski jezik, a služi se i njemačkim jezikom.