Monitoring and Analysis of Player Behavior in World of Warcraft

Mirko Sužnjević, Maja Matijašević, Borna Brozović
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, Zagreb, Croatia
{mirko.suznjevic, maja.matijasevic, borna.brozovic}@fer.hr

Abstract - **Massively Multiplayer Online Role-Playing Games (MMORPGs) comprise persistent virtual worlds in which thousands of players are interacting. Information regarding player behavior in MMORPGs is highly relevant for network planning, game design, server load balancing, etc. The main contribution of this paper is wJournal – a tool for detailed measuring of player behavior for the most popular MMORPG – *World of Warcraft* by *Activision Blizzard*. The behavioral model on which wJournal is based upon consists of six player action categories (*Questing, Trading, PvP combat, Dungeons, Raiding, and Uncategorized*). We demonstrate the use of wJournal by monitoring and analysis of 16 players over the course of four weeks' time.**

I. INTRODUCTION

Massively Multiplayer Online Role-Playing Games (MMORPGs), unlike other multiplayer games, support very large numbers of concurrent users, possibly tens of thousands, playing together in a common, persistent virtual world. Each user is represented by a virtual character through which the player interacts with the game. Through the course of the game, virtual characters become more powerful and progress in terms of levels (up to a maximum level), abilities, and possession of virtual items (e.g. swords and armor). As of February 2012, according to *mmodata.net* [1], a reference source for the (estimated) number of players for subscription based games, there are around 21 million MMORPG subscribers, out of which more than 10 million play the *World of Warcraft* (WoW), a MMORPG developed by *Activision Blizzard*.

A wide popularity of MMORPGs reflects onto network traffic as well. For example, according to measurements reported in 2007 in a live 3G mobile core network in Austria [2], the amount of TCP traffic generated by WoW constituted over 1% of total TCP traffic. While the volume of MMORPG traffic is, in general, still relatively low, it is growing fast. According to Cisco Visual Networking Index [3], the online gaming traffic volume (including casual online gaming, networked console gaming, and multiplayer virtual-world gaming) will grow with a compound annual growth rate of 43% in the period 2010–2015; topped only by Internet video (48%).

Network traffic monitoring and analysis have always played a key role in studying and improving the operation of telecommunication networks and the Internet. With respect to online gaming, a recent trend in traffic monitoring and analysis has been oriented towards understanding player behavior with respect to traffic modeling and user perceived quality. Information about player behavior can be utilized in many ways. From the perspective of network infrastructure equipment vendors and network operators, it is a key for development of traffic models used for network planning dimensioning, and testing [4][5]. From the game design perspective, information regarding player behavior and movement in the virtual world is crucial for virtual world partitioning, area of interest management algorithms, and setting minimum bandwidth and latency requirements. From the business perspective of a game provider, it helps estimate users (dis)interest in the game and predict player churn (i.e., the number of users who quit playing entirely). Player behavior also has an impact on computational resources and energy consumption [6]. Thus, tools for monitoring and analysis of player behavior are needed by all stakeholders in the online gaming business.

In this paper, we present our contribution, a tool we developed for monitoring and analysis of player behavior in WoW. The tool collects player behavior related data on selected game related characteristics, such as a number of sessions over a monitoring period, a number of different game characters within each session, types and timing of gaming activities, and the areas visited in the virtual world. For the categorization of gaming activities, we apply a previously developed model of player behavior [7]. As people keep journals to record their activities and happenings in real life, so does the tool for their virtual persona(s) in WoW – hence its name *wJournal* (as short for "*WoW journal*"). We illustrate the use of the *wJournal* by performing measurements to characterize user behavior in a group of 16 players, and compare it to previous results, and also to investigate the relationship between the player behavior, location, and movement patterns in the virtual world.

The remainder of the paper is structured as follows: Section II briefly summarizes related work on approaches and monitoring and data collection for studying player behavior, and Section III presents the player behavior model which serves as a basis for our tool. Section IV presents the design and implementation of *wJournal*, and Section V describes and discusses the measurement results. Finally, Section VI concludes the paper.

II. RELATED WORK

There are three complementary approaches for studying player behavior in MMORPGs, based on where and how the player behavior data is tracked and recorded:

- At the server: this approach is based on the game provider's internal server data logs, which, obviously, can only be obtained or accessed only in close cooperation with the game provider;
- At the client: this approach is based on either extending the default game client by developing an add-on (using an API provided by the game developer), or, creating a new or modified game client with desired functionality (possibly subject to technical, legal, and/or licensing restrictions). Using an add-on (also spelled as "addon"), or a modified client for behavior monitoring requires cooperation with players;
- In the network: this approach is based on capturing game traffic in the network and analyzing it, usually offline, to extract the behavioral information from those traces. Access to network infrastructure is needed in this approach, and depending on the measurement point, cooperation with the network operator may be needed (or not).

Each approach has its advantages and disadvantages with respect to the amount and descriptiveness of data, as well as its suitability for a particular purpose. For the purposes of this work, we briefly summarize the user behavior characteristics which have been studied in literature.

Server data approach. Feng, Brandt and Saha [8] have performed an analysis on several years of server log data provided by developers of popular MMO *Eve online*. They investigate the number of players logged into the game at any given time. They have also presented a long term analysis and demonstrated how adding new content to the game influences player behavior.

Client data approach. In addition to collecting data about the player using the client itself, several works use polling, meaning they use a script, running on a client, which uses regular game client mechanisms to "poll", i.e., obtain information on, other virtual characters (i.e., players' avatars) present in the virtual world. This approach is used by Pittman and GauthierDickey [9] who record and model the behavior of two MMORPGs: WoW and *Warhammer Online* by *Mythic Entertainment*. Authors also mention that another option for measuring player behavior would be to modify the game client, which could provide better results, but with possible legal consequences. Lee, Chen, Cheng and Lei [10] gather the data of player behavior in WoW over the course of three years, constantly refreshing the avatar presence. Data is used to predict avatar play time, since there was no way to correlate the avatar to the actual player. Zhuang et al. [11] measure session length, player availability, player downtime, and player location in the virtual world. Authors classify the virtual world zones into areas named *Questing*, *Traveling*, and *Cities*, and calculate the time a player stays in each area. In our previous work [7], we introduced a WoW add-on named *World of Warcraft Session Activity Logger* (WSA-Logger), a predecessor to *wJournal*. (Compared to WSA-Logger, *wJournal* has several improvements in terms of functionality and performance, which will be described later in this paper.)

Network approach. This approach is applied by Kihl, Aurelius, and Lagerstedt [12] who inspect network traffic in an access network in Sweden and extract the traffic from a number of households in which people played WoW. They present information regarding hourly and daily patterns of the number of active players, traffic load, and session length. A correlation between time of the day and traffic generated by WoW is used to predict server load.

III. PLAYER BEHAVIOR MODEL

In this section we briefly explain the behavioral model on which *wJournal* is based upon. The behavioral model, specified in our previous work [7], and examined in respect to psychological player motivation [13], specifies six action categories of user behavior, which translate to the states in the finite state machine used by the software. We base the internal logic in *wJournal* on states and transitions corresponding to their equivalents in the system of WoW: the states correspond to the action categories, and the transitions correspond to events (messages sent) which denote the start of a certain game activity.

In our player behavior model [7], we have defined the following action categories for MMORPGs: 1) *Questing*, 2) *Trading*, 3) *Dungeons*, 4) *Raiding*, 5) *Player versus Player (PvP) combat*, and 6) *Uncategorized*. We briefly describe the action categories next.

Questing is an action category involving solving task given by Non-Player Characters (NPCs) for a reward in terms of experience, virtual items, or in-game currency. It is usually performed by a player alone, as questing tasks are usually fairly simple (e.g., kill 10 wolfs).

Trading category includes actions which serve for exchange of virtual goods or money between two players, or, an NPC and a player. Such actions include sending in-game mail, trading virtual items, visiting banks, etc.

Dungeons is a primary small group activity in which a group of players, typically 4-5, fights against a large number of hostile NPCs inside confined areas of the virtual world. These areas are replicated for each group of players and are called instances.

Raiding is an action category which is very similar to *Dungeons*, but larger on all scales. It involves larger groups of players (i.e., 10 and more), NPC opponents are more challenging, and fights are more complex. Therefore, *Raiding* provides better rewards then *Questing* and *Dungeons*.

PvP combat involves actions of battling between players, both in instanced and in non-instanced areas. Its distinctive characteristics are high action pace, high player mobility, and low number of NPCs.

Uncategorized comprises the time within the session which cannot be uniquely assigned to one of the five categories mentioned previously.

IV. DESIGN AND IMPLEMENTATION OF WJOURNAL

wJournal has been designed as a WoW add-on. This design choice has been based on the fundamental requirement which stems from the player behavior model

– the ability to detect what the user is doing in the WoW virtual world (i.e., to which action category this activity refers to), when and for how long, and where (in which are of the virtual world). *wJournal* enables tracking of the number of sessions, the number of different characters within each tracked session, start time and end time of session segments which represent a single behavior category within a session, and the list of visited areas with associated time duration of the visit. Compared to our previous add-on (WSA-Logger), *wJournal* has improved performance in terms of reduced memory usage (70kB total) and CPU load (less than 1% on the default CPU performance meter). It also has a new functionality for tracking user movement patterns. Other minor improvements include bug fixes and some GUI details.

A. *Event tracking in wJournal*

Whenever a certain activity in the game is performed, the WoW API of the game client fires an event related to that activity. For example, when the user opens a mailbox inside the game, the MAIL_SHOW event is fired. The number of events has been growing with the game evolution, and as of March 2012, there are 736 distinct events within WoW. In order to track player behavior with respect to the player behavior model, certain in-game events are helpful for determining action categories, and some are not. For example, the event MAIL_CLOSE may be uniquely assigned to the *Trading* category, and the event BATTLEFIELDS_SHOW may be uniquely assigned to *PvP combat*. However, a large number of events happen constantly during gameplay; making it impossible to distinguish the action category based solely on that event. Event tracking by using WoW API is explained in more detail in our previous work [7].

For the purposes of this work, it suffices to say that *wJournal* tracks *Questing*, *Trading*, and *non-instanced PvP combat*, as a chain of events starting from the first event assigned to a given category, to the first event which does not belong in the same category. *Dungeons*, *Raiding*, and *instanced PvP combat* are all variations of instanced world space and are tracked differently. Namely, these categories are fully defined by two events labeling the entrance in and exit out of the instanced area.

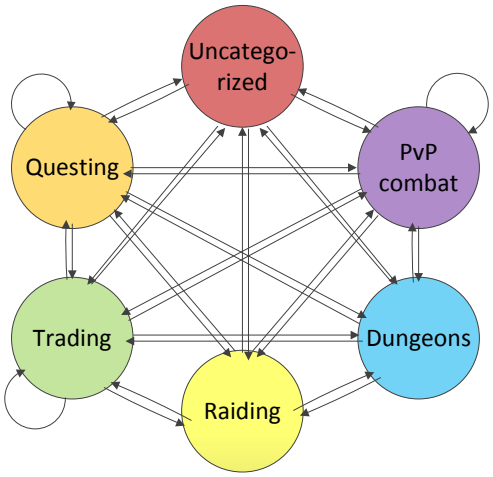

Figure 1. User behavior tracking finite state machine in wJournal

So, each time a player enters and spends time in an instanced area, the player is considered to be performing events belonging to the respective behavior category.

B. *Player behavior tracking in wJournal*

A player behavior within the game client, as tracked by the add-on, can be modeled as a finite state machine, as shown in Fig. 1. The set of states corresponds to "user state", as the user "being active" in one of the six action categories. The set of input events corresponds to user actions in the game world, and a set of output events corresponds to events fired by the API. The transition function takes the current state and an input event, and returns the output event and the next state. Since there are no such restrictions within WoW, any state can change into any other state, thus transitions are possible from each state to all other states. Also, for non-instanced states (*Questing*, *Trading*, and *non-instanced PvP combat*), there is a loop denoting that there can be a series of events which do not change the current state. For instanced states (*Raiding*, *Dungeons*, and *instanced PvP combat*), as mentioned before, the state is defined through the instance entry and instance exit events.

By implementing the model described above, we are able to track the players' behavior in all of their characters separately, show them the gathered statistics in form of bar-charts, and export all the data needed in a log file.

C. *wJournal implementation*

As mentioned earlier, *wJournal* has been implemented as a WoW add-on, by using Lua scripting language and Ace3 development framework (http://www.wowace.com/add-ons/ace3/). A WoW add-on, in general, is a collection of files which reside within the respective subdirectory in the WoW directory tree on the player's computer. An add-on directory contains a (mandatory) table of contents file, and a collection of individual components which implement the desired functionality, such as Lua scripts, XML files, fonts, graphics, sounds, etc. These files are loaded into the game scripting system when the client is started, and executed within the client to make desired modifications to the user interface. Due to security reasons, the scripting interface has limitations, in particular in terms of access to the file system and the operating system resources. It should also be noted that the scripting interface does not have access to the game world part of the client, so as to prevent misuse such as, e.g., automating behaviors or modifying gameplay. For more information on creating and programming WoW add-ons, an interested reader is referred to [14].

1) *wJournal add-on structure*

The wJournal directory contains the following files: 1) a table of contents (metadata) file, 2) an XML file used to determine the location of, and order in which Lua scripts (files with .lua extension) are loaded, and 3) Lua script files (the actual code which runs the add-on). The amount of data that needs to be tracked at any given time is substantial, so our add-on is broken down into multiple files for better code readability. In addition to the files

mentioned above, two additional files include *Zones* and *Events*. Their function is explained in the text below.

2) Zone tracking

Virtual world of WoW contains more than 100 distinct zones. A *zone* is an area in the virtual world defined by its size and name. A character's location at a given time can be determined by using a script. Keeping track of the zone in which the player is in drastically reduces the number of events that we need to keep track of. For example, if a player's character is transferred to an instanced zone like battlegrounds, there is no need to keep track of any other events while in it. Simply noting the zone entrance and exit is enough to determine which instanced behavior category a player is currently in.

3) Event tracking

The *Events.lua* file is used for storing a list of events used for determining the behavior category. This list is extendable, in the sense that new events (meaningful for our categories) added into the game (e.g., by game updates) may be added into this file to make *wJournal*'s decisions more precise.

4) Core logic

wJournal's program logic is stored in the *Core.lua* file. This file implements the finite state machine described above and records the data about the player behavior, with the goal of calculating and presenting the player behavior statistics. Obviously, calculating these in real time would have a negative impact on game client performance so that has to be avoided. The next problem that has to be solved is how to store collected data, and remain in line with security restrictions for add-on's Lua scripts. To avoid performance degradation and to work around file system access limitations, *wJournal* separates the behavior data presentation to the player from data storage for offline processing.

5) Player statistics shown within extended GUI

The display showing the behavior statistics to the player has been designed so as to be rather simple and non-intrusive and it can be easily put aside if needed.

Fig. 2 shows the display, using a bar chart for graphical representation of the basic player behavior statistics. In addition to bar length indicating the time spent in each action category, the chart also displays the percentage of the time spent in a certain category with respect to the total playing time.

6) Player statistics stored in a text file

For security reasons, Lua scripts in WoW add-ons are permitted write access only to an isolated folder, where settings for all add-ons are stored.

Figure 2. wJournal display of playing statistics

The problem is that using an add-on over an extended period of time, results in storing large amounts of data in files originally designed for settings.

This, obviously, has a cumulative negative impact on performance. Our past experience with *WSALogger*, which used this mechanism, taught us that a better solution was needed. For *wJournal*, another solution has been found by using the WoW's option for creating custom chat channels. Custom chat channels have several properties that enable this solution: 1) they can be used by any number of users, 2) add-ons are allowed to write in such channels without the users having to type themselves, and 3) add-ons can selectively toggle chat logging on and off, allowing to write only the text generated by the add-on into the game client's chat history, virtually making it our export file.

A chat history line is structured as follows:

```
MM/DD hh:mm:ss [channel name] <avatar
        name>: <text>
```

Where the MM/DD field represents month and day, and it is followed by the time of the day expressed in hours, minutes and seconds (hh:mm:ss). The "channel name" is used to identify the originator of the chat message, as the avatar's name is used to identify the player who wrote the message. The "text" field represents the message received. Since the game client provides timestamps, channel names, and avatar names automatically, we use the text field for writing the data of interest. An algorithm for writing into a chat log is as follows:

1. Wait for an event;
2. If an event is registered as belonging to a certain behavior category, forward it to the state machine as a new input event;
3. If the state machine transitions to a new state, send the new behavior category name and zone in which this event has occurred to a custom chat channel.

There is also a basic algorithm for data filtering before writing into the chat log, defined as follows:

1. Wait for a change in the chat window;
2. If the channel on which the message is received and the current player's avatar name have been previously registered in *wJournal*, enable chat logging. Otherwise, disable chat logging.

Based on the chat history, it is possible to chronologically reconstruct the sequence of player behaviors, i.e., the behavior start time, the behavior end time, and the zone in which the behavior has occurred. To make handling of data easier, a C# parser in has been developed, which creates output in Microsoft Excel (XLS) format. This file contains the following: Player ID, character name, behavior category name, starting time of a category, ending time of a category, and a session ID. This allows us to easily sort, enumerate, find peak values and create graphs from collected data. We also study the data to analyze the time duration spent by the player per action category, and the spatial positioning of characters.

V. Measurement Results

We demonstrate the use of *wJournal* by the following experiment, run by student volunteers at the Faculty of

Electrical Engineering and Computing of the University of Zagreb, over a four weeks' time. The students "enlisted" a total of 16 WoW players to participate in the research. The participating players were requested to set up the *wJournal* add-on on their computers, and to continue their usual behavior inside the virtual world during the course of the experiment.

Data gathering started on May 5th, 2011, and it ended on June 10th, 2011. During this period, WoW patch 4.1.0 of the Cataclysm expansion was active. As the previous patch was introduced on October 12th, 2010 (meaning that no new content has been introduced which would "skew" the user behavior), players have behaved in their already settled ways, i.e., in usual patterns. During the experiment no personal data was collected, except player's age and gender. A total of 16 players participated in our research, with an average of 1.937 characters per player (as one player may use several characters in a same session).

The log analysis has been performed to establish the age and gender characteristics of the test group, their session characteristics, and the player movement.

A. *Age and gender analysis of the test group*

The average player age in our dataset is 24 years, with a 16-year-old being the youngest, and a 32-year-old the oldest subject. In general, younger subjects showed the tendency to play mainly with one character and focusing on single behavior categories, while older players showed a variety of interests. Distribution by gender is roughly equal (i.e., 9 male and 7 female players). With respect to activity categories, the collected data shows that "traditional" gender roles seem to extend into the virtual world as well, as female participants showed more interest in cooperative and healing roles, while male participants focused on playing competitive and more aggressive roles.

B. *Session length analysis*

We define a *session length* as the time spent in the game between logging in and logging out of a single character. One player can have multiple characters, but only one can be active at a time.

We also define a *session segment* as a part of the session in which only the activities in one action category take place. In our test group, an average session time is 62 minutes, with the longest session just under 14 hours. The shortest sessions lasted only a couple of seconds. Such sessions can represent an accidental character logons, as well as logons to simply check some equipment status, or chat. Therefore, information about shortest sessions is not reported separately, but it is calculated into average session lengths.

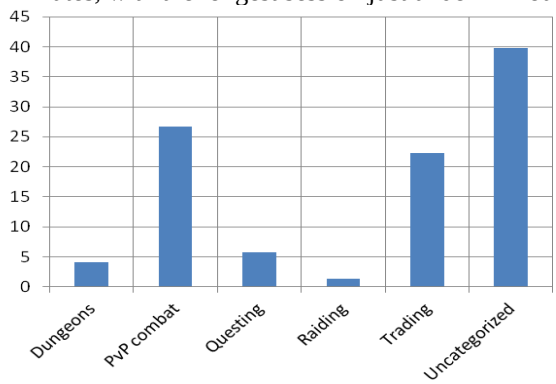

Figure 3. Occurrence percentages of behavior categories

C. *Behavior category duration analysis*

Fig. 3 shows the average occurrence (percentage) of the six behavior categories in our dataset, and Fig. 4 shows (the percentage of) the time spent in each category. Due to a relatively small number of participants in the experiment, who happened to be mostly interested in *PvP combat*, the behavior category spread is imbalanced. Actions recorded mainly belong to *PvP combat*, *Trading* and *Uncategorized* categories, while *Questing*, *Dungeons* and *Raiding* (in descending order) are less represented. A typical session with a maximum level character contains chatting, discussing tactics and trading while waiting to be matched against other players in *PvP combat*, which is played the most. When not playing *PvP*, players are mostly questing with their alternative characters, or leveling them through dungeons.

Only three (out of sixteen) test subjects have shown interest in raiding, which is played mainly in evening hours. By comparing Fig. 4, and Fig. 3, it may be noted that, even though *Uncategorized* session segments happen very often, their total time is not as high in percentage as expected. It is also noticeable that dungeons and raids have only a few occurrences, but their percentage in total time played is not insignificant because the corresponding session segments tend to be long (as shown in [7][13]). We can order the behavior categories from the most to the least frequent, as follows: *Uncategorized*, *Trading*, *PvP combat Questing*, *Dungeons*, *and Raiding*. The average length of each category, also in descending order: *Raiding*, *Questing*, *Dungeons, PvP combat, Uncategorized , Trading*.

D. *Player movement analysis*

Players are able to move through virtual zones in several ways: by land, by air, and through teleporting.

Travelling by land means a player can follow designated routes on foot. Thus, a player can only visit zones adjacent to his current zone.

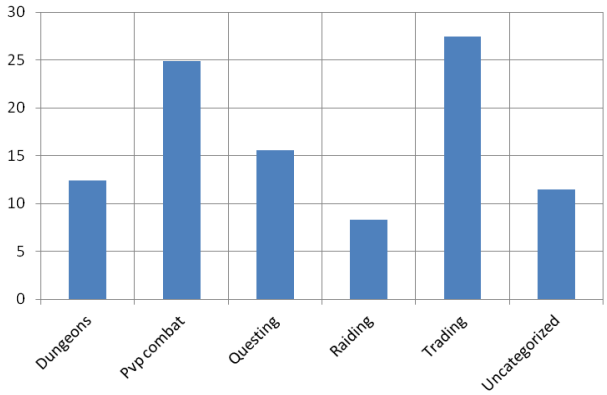

Figure 4. Percentages of time spent on each behavior category

This is characteristic for low level characters, since they have no faster options available until higher levels.

Travelling by air is acquired at mid levels. A player is able to purchase a flying mount which still limits him to travelling only to adjacent areas, but not as strict as roads do. Once the flying ability is acquired, there is no recognizable pattern in player movement when travelling by air.

Teleporting is a way of instantly transporting a player to a zone not necessarily adjacent to his current position.

Since there are virtually endless combinations of traveling patterns, no automated process is used for recognizing player movement. Instead, the parsed data is manually analyzed.

In our dataset, only 16% of characters are lower levels and are traveling by land or air. Most players are maximum level, where we find little to no movement at all. We have also confirmed that characters tend to gather in major cities. Idling and trading are the two behaviors found there. From capital cities they tend to enter queues for instanced actions. Approximately every 26 minutes players are teleported to a dungeon, or every 19 minutes to a battleground. Afterwards, they continue to idle or trade until the next queue is up. Most of the maximum level players follow this behavior pattern.

E. Relationship between actions performed and location in the virtual world

We have also analyzed which are the most frequent areas related to each action category. In general, players tend to visit only a few zones. Trading almost exclusively happens in capital cities or adjacent areas. Time spent in *Uncategorized* is mostly (over 75%) spent in capital cities as well. *PvP combat* is almost completely focused in instanced areas, and occurrences of *PvP combat* outside these areas are very rare. All instanced areas for *PvP combat* are frequently used. This differs from the *Dungeons* category, where the *Dungeons* intended for maximum level players are used 91% of the time. This means that a large amount of content is not used. This is even more pronounced in *Raiding* where only two instances have ever been visited.

VI. CONCLUSION

By using *wJournal*, we have gained an interesting insight into the player behavior in the test group. The following conclusions have been made:

- Players tend to gather at certain hotspots (hubs), making spatial distribution of server load uneven.
- Behaviors that players show most interest in, (e.g., PvP combat), mostly take place in instanced zones, causing constant "server hopping".
- Some of the instanced areas for dungeons and for raiding are very rarely used.

This information may be significant for designing virtual world partitioning algorithms and server load balancing, as well as on network capacity planning and dynamic resource reservation. Other uses, as explained at the beginning of the paper can also be envisioned.

For future work, we plan to create a tool for predicting server and CPU load based on user behavior patterns we have identified.

ACKNOWLEDGMENTS

This work was supported by the research project 036-0362027-1639 "Content Delivery and Mobility of Users and Services in New Generation Networks", by the Ministry of Science, Education, and Sports of the Republic of Croatia, and Ericsson Nikola Tesla, Zagreb, Croatia. Also, the research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement no. 285939 (ACROSS).

REFERENCES

[1] I. Van Geel, "MMOData Keeping track of the MMORPG scene", www.mmodata.net

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

[3] Cisco Visual Networking Index: Forecast and Methodology, 2010-2015, On-line: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html, February 2012.

[4] K.-T. Chen, P. Huang, C.-L. Lei, "Game Traffic Analysis: An MMORPG Perspective", Computer Networks 50(16), pp. 3002-3023 (2006)

[5] W. Cai, X. Wang, M. Chen, Y. Zhang, "MMOPRG Traffic Measurement, Modeling and Generator over WiFi and WiMax", In: Proceedings of the Global Telecommunications Conference (GLOBECOM 2010), IEEE, pp. 1-5 (2010)

[6] Y.T. Lee, K.T. Chen, "Is server consolidation beneficial to MMORPG? a case study of World of Warcraft", In: IEEE International Conference on Cloud Computing, pp. 435-442 (2010)

[7] M. Suznjevic, O. Dobrijevic, M. Matijasevic, "MMORPG player actions: Network performance, session patterns and latency requirements analysis", Multimedia Tools and Applications, 45(1-3), pp. 191-241 (2009)

[8] 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.

[9] D. Pittman and C. GauthierDickey, "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.

[10] 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.

[11] X. Zhuang, A. Bharambe, J. Pang, and S. Seshan, "Player dynamics in massively multiplayer online games", http://reportsarchive.adm.cs.cmu.edu/anon/2007/CMU-CS-07-158.pdf, October 2007.

[12] 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.

[13] M. Suznjevic, 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)

[14] J. Whitehead, II, R. Roe, "World of Warcraft Programming: A Guide and Reference for Creating WoW Addons", 2nd Edition., John Wiley & Sons, 2010.