

Analyzing the effect of TCP and server population on massively multiplayer games

Mirko Suznjevic¹, Jose Saldana², Maja Matijasevic¹, Julián Fernández-Navajas², José Ruiz-Mas²

¹University of Zagreb Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia

[e-mail: {mirko.suznjevic, maja.matijasevic}@fer.hr]

²Communication Technologies Group (GTC) – Aragon Institute of Engineering Research (I3A)
Ada Byron Building. EINA, University of Zaragoza. 50018 Zaragoza, Spain

[e-mail: {jsaldana, navajas, jruiz}@unizar.es]

*Corresponding author: Mirko Suznjevic

Abstract

Many Massively Multiplayer Online Role-Playing Games (MMORPGs) use TCP flows for communication between the server and the game clients. The utilization of TCP, which was not initially designed for (soft) real-time services, has many implications for the competing traffic flows. In this paper we present a series of studies which explore the competition between MMORPG and other traffic flows. For that aim, we first extend a source-based traffic model, based on player's activities during the day, to also incorporate the impact of the number of players sharing a server (server population) on network traffic. Based on real traffic traces, we statistically model the influence of the variation of the server's player population on the network traffic, depending on the action categories (i.e., types of in-game player behaviour). Using the developed traffic model we prove that, while server population only modifies specific action categories, this effect is significant enough to be observed on the overall traffic. We find that TCP *Vegas* is a good option for competing flows in order not to throttle the MMORPG flows, and that TCP *SACK* is more respectful with game flows than other TCP variants, namely *Tahoe*, *Reno* and *New Reno*. Other tests show that MMORPG flows do not significantly reduce their sending window size when competing against UDP flows. Additionally, we study the effect of RTT unfairness between MMORPG flows, showing that it is less important than in the case of network-limited TCP flows.

Keywords: MMORPG, traffic modelling, online gaming, player behaviour, TCP

This work has been partially financed by the Project "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; the European Community Seventh Framework Programme under grant agreement no. 285939 (ACROSS); CPUFLIPI Project (MICINN TIN2010-17298); Project TAMA, Government of Aragon; Project Catedra Telefonica, Univ. Zaragoza; European Social Fund in collaboration with the Government of Aragon. The authors would like to thank John Miller for his advice regarding some details of the game traffic. We also want to thank Tanja Kauric for her help in obtaining the traffic traces of the game.

1. Introduction

Massively Multiplayer Online Role-Playing Games (MMORPGs) have become one of the most profitable genres in the gaming industry. The leading MMORPG in the market, namely *World of Warcraft* (WoW) by *Activision Blizzard*, at its peak, had approximately 12 million players [1], and it reported around one billion US dollars of profit in 2010. MMORPG players demand interactive virtual worlds, so a good underlying network quality is needed. In other words, the traffic generated by virtual worlds of MMORPGs has very high Quality of Service demands in terms of delay and packet loss. While MMORPGs are real-time multiuser virtual worlds, many of them use TCP for communicating the actions of the player to the server, and vice-versa.

The use of TCP as a transport protocol is not very widespread in the area of networked games. Besides flash based web games, TCP is not that common. Most of the games which feature full real time 3D virtual worlds use UDP, including First Person Shooters (FPS), racing, Real Time Strategy (RTS), Multiplayer Online Battle Arena (MOBA), etc. MMORPGs are one of the few game genres in which the use of TCP is employed (although UDP is used as well, depending on the game [5]). Most MMORPGs use the same client-server architecture with client holding all the application logic and 3D information, while the only thing exchanged with the server are the updates of specific entities in the virtual world. This enables very low requirements of these games on network bandwidth which are common in the whole genre [13]. WoW is, according to [2], still the most popular subscription based MMORPG in terms of number of players with around 8 million active subscribers (even after losing 4 million). Therefore, WoW still holds a very large portion of the market, and its traffic has more impact on the network than the traffic of next ten MMORPGs combined which makes it a logical choice for a study focused on effects of using TCP in an online game.

Using TCP has many implications, taking into account that TCP was initially designed for bulk transfers [3], [4], with the main objective of transmitting an existing amount of data with the maximum throughput, always maintaining fairness between flows. Since the throughput limit of bulk transfers mainly depends on network bandwidth, we can talk about *network-limited traffic*. However, an MMORPG sends information which does not previously exist, but is continuously generated according to the player's actions. As a consequence, MMORPGs do not always exhaust their bandwidth share, as sometimes the application has nothing to send (*application-limited traffic*), and this makes some TCP mechanisms inefficient or even counterproductive. As an example, the authors of [4] reported that TCP back-off mechanism did not activate, and also that fast retransmissions are very exceptional because of the high inter-packet times, leaving most recoveries to be made by timeout; furthermore, a correctly received TCP packet following a lost packet would be blocked from delivery to the application until the lost packet was eventually retransmitted successfully. In addition, some TCP mechanisms (e.g., Delayed Acknowledgment mechanism [5]) innately cause Quality of Service degradation.

While the impact of using TCP on MMORPGs performance has been studied [3], as well as the inspection of different TCP versions [4], limited work has been done on investigating how other TCP flows in a network would impact the MMORPG flows, as well as the analysis of RTT unfairness. MMORPG flows share a large portion of the network path with other flows, both in local networks of the end users (e.g., wireless LAN, or DSL connection) as well as in the network operator core network. The rise of these games is modifying the traffic mix

present in operator’s networks, increasing the rate of small TCP packets which, due to the interactivity requirements of the games, cannot be considered as “best effort” flows. This fact makes it necessary to study the interaction of MMORPG application-limited TCP flows with “traditional” flows, such as TCP bulk transfers and UDP flows.

To perform realistic tests, accurate statistical models of the network traffic are commonly used. Such models are also useful for network capacity planning and optimization. However, the traffic of online games in general is very difficult to model [6] and strongly dependant on player behaviour at the application level [7]. What must be taken into account is that an MMORPG allows a wide range of activities, from picking flowers to fighting dragons with the help of some friends; and logically, the game interactivity and usage of network resources varies significantly depending on the deployed task.

To enable the characterization of the network traffic under different application conditions (i.e., picking flowers against fighting dragons), in previous work we have grouped activities within the game into different behaviour categories: *Questing*, *Trading*, *Player versus Player (PvP) combat*, *Dungeons*, *Raiding* and *Uncategorized* [9]. We will follow this classification in the present work. As shown in Fig. 1, which has been obtained from empirical traces, the traffic varies significantly depending on the player’s activity. A source-based network traffic model based on defined behavioural categories was presented in [10]. This model comprises two main components: 1) a teletraffic model for each of the proposed categories, 2) player’s behaviour (i.e., the probability of a player deploying an activity in a certain moment).

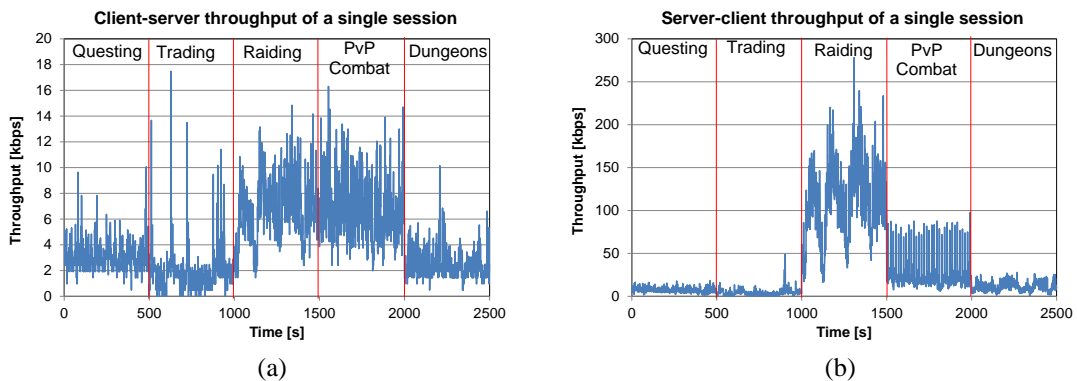


Fig. 1. Bandwidth usage of the original game for different categories: a) client-server; b) server-client (the trace presented in the figure is a set of characteristic fragments obtained from separate captures)

At the same time, one of the major problems for MMORPG’s providers is scalability. To calculate the virtual world state when a large number of entities are in a small area of that world is a difficult task in real time [11]. Therefore, game operators often replicate the virtual world into multiple independent “shards” or servers, and divide the player base across them. As a consequence, players cannot interact between shards. These techniques combined with Area of Interest (AOI) management (avatars out of the AOI of a player are considered not to interact with him), create significant load reductions. For different reasons, however, some of the shards become highly popular, whereas other are almost “deserted” (e.g., in WoW a highly popular server in Europe is *Outland*, while an almost deserted one is *Jaedenar*). The server population (i.e., the number of players on particular server or shard) has a significant effect on network traffic characteristics, mainly in the server-to-client direction. This is because the server has to send to the client application the updates of the state of all the entities in the AOI of the player, and this information increases in volume if the server is more crowded.

Having in mind the need for an accurate traffic model, we first deploy tests with the proposal of capturing the impact of the number of players in a shard onto the network traffic. In other words, the initial traffic model is extended to be able to modify Application Protocol Data Unit (APDU) sizes and Inter-APDU Arrival Times (IAT) according to the population of a particular server. This mainly affects the activities in which the number of players significantly varies, as *Trading* and *PvP combat*. This relationship between the server traffic and the number of players in the server has been measured for a concrete game, but this phenomenon is expected to happen in other similar games which use sharding.

Once the effect of server population has been statistically characterized, a full traffic model has been implemented in NS2 network simulator, and used to deploy a number of tests with the aim to explore the interactions between MMORPG and other traffic flows. The competition between different flows when sharing a bottleneck is studied, including:

- Interaction of MMORPG and network-limited traffic using different TCP variants,
- Competition of MMORPG and UDP traffic,
- Influence of RTT on the behaviour of MMORPG flows (RTT unfairness),
- Global effect of the amount of players in a shard.

All in all, the contribution of this paper is threefold: first, the study of how MMORPG traffic characteristics change depending on the number of players which, to the best of our knowledge, is reported and measured for the first time. Second, once a realistic model is available, several tests exploring interactions between TCP MMORPG flows and other traditional applications have been deployed in order to illustrate the characteristics of competition between MMORPG flows with other network flows. Finally, the proposed model has been fully implemented in NS2. This NS2 script, which allows a wide range of tests, is offered to the research community. As previously stated, the model takes into account the variation of the player behaviour during the hours of the day, modifying the probability of the player's activity, and the duration of each activity.

The structure of the paper is as follows: in Section 2, we review the previous work in MMORPG traffic classification, characterization and interaction with other kinds of traffic. Section 3 addresses the question of obtaining the traffic model depending on the activity and population. Section 4 presents the MMORPG traffic competition tests deployed using the model, and the Conclusions section closes the paper.

2. Related Work

Network traffic of MMORPGs has been a target of a number of studies which are summarized in survey papers [12], [13]. Wu, Chen *et al.*[3], [5] evaluate performance of TCP for online games in general and discuss whether TCP is in fact a suitable protocol for MMORPGs. They identify some network performance degradation problems derived from the traffic characteristics of this game genre: tiny packets, low packet rate, application-limited traffic generation and bi-directional traffic. They remark that TCP is normally used by applications which deploy bulk transfers, so the bandwidth is limited by the network. In the case of MMORPGs, there are some moments in which the application has no data to send, and this may produce the effect of the TCP congestion window being reset unnecessarily. In other cases, the window may become arbitrarily large, producing bursts demanding a high amount of bandwidth from the network.

The authors of [4] investigate the performance of different TCP versions with respect to

retransmission delay when low-rate, real-time event streams are sent to clients. They tested the existing TCP variants, i.e., *New Reno* (plain, with SACK, DSACK, FACK, and with DSACK and FACK), *Westwood*, BIC and *Vegas* on Linux. They concluded that there are only small differences between TCP variants which are used for MMORPG flows, but also that multiplexing different flows into one TCP connection and a more aggressive timeout retransmission time promise a reduction of the delay perceived at the application level. In addition [14] explored the reservation on part of the path between a game server and a number of clients, discussing the implications of using it for network infrastructure. In our previous work [15] we presented a preliminary study of the coexistence of WoW and network-limited TCP using different variants. The present paper further explores this issue, including the effect of RTT, and it also extends the study to the coexistence with other flows.

In the area of network traffic modelling, WoW has been a use case in several studies, as it is the most popular subscription-based MMORPG. An initial traffic model for this game was presented by Svoboda [16], together with the analysis of a traffic trace captured within a 3G mobile core network, which showed that WoW was one of the ten most popular TCP services in the monitored network. Other MMORPGs have also been modelled: *Lineage* [17] and *World of Legend* [18]. All the previously listed models are based on a methodology proposed in [19]; however, a different traffic model for WoW [20] is based on a transformational scheme developed for the highly erratic network traffic of online games.

While previous approaches focus on modelling the traffic of a game as a whole, another approach for addressing the highly variable network traffic of MMORPGs is based on classification of application level states. In [21] Park *et al.* applied such approach for MMORPG – WoW and a First Person Shooter (FPS) – *Quake 3*. The actions defined for WoW are: *Hunting the NPCs*, *Battle with players*, *Moving*, and *No play*. Another classification has been proposed by Wang *et al.* [22], proposing *Hunting*, *Battlefield*, and *Downtown*. The authors also inspect different scenarios of movement for the player (in the real world): subway, bus, and campus. In the following work, the same research group proposed traffic models for each category, which were implemented in NS2 [23].

There are several studies which focus on the number of players in MMORPGs [24], [25], [26], [27], [26][28]. The most in-depth analyses have been performed by means of cooperation between game providers with the research community by providing internal datasets. Such analyses have been performed for *EvE Online* [24] and *EverQuest II* [25].

Another approach to the investigation of the number of players is based on the use of application specific capabilities of clients, in order to obtain more information regarding the state of the virtual world. Through the development of scripts which run within the client, it is possible to obtain a range of information regarding currently active players in the virtual world. Notable studies using this approach are Pittman's [26], [27] and Lee's [28], both been performed on WoW. Pitman *et al.* investigated a number of players on *Aerie Peak*, a North American WoW server labelled as *full*. They noted that the average daily number of users peaks around 3,500 players [27]. The authors also concluded that the spatial distribution of players in the virtual world is not uniform, but there are hot spots in which players gather, while a large portion of the virtual world is empty. On the other side, Lee *et al.* presented a dataset comprising information regarding players on an Asian WoW server *Light's Hope*, captured for 1,107 days [28]. Lee's dataset only comprises the data about one of the two factions of the game, and presents peaks below 500 players. Even if we double that value in an effort to estimate the number of both factions, it is still 3.5 times lower than the values reported by Pittman *et al.* Therefore, it is evident that different servers for the same game have

significantly different player populations. The causes of this may vary, depending on social dynamics on each particular server, technical difficulties, server migrations, etc. The discrepancies in player population cause differences in both computational and network load.

In previous work, we defined the following action categories: *Questing*, *Trading*, *Player vs Player (PvP) combat*, *Dungeons*, *Raiding* [9], we developed network traffic models for each of the categories, performed player behaviour measurements and modelling [8], and created a software architecture able to generate real traffic using the previously defined model [10].

Some traffic models of other game genres (i.e., FPS games) [29], [30] have considered the effect of the number of players sharing the virtual world. However, the number of concurrent players of these genres is limited to a few tens, whereas an MMORPG shard can host thousands of players, adding a high degree of complexity which requires different techniques like e.g., the definition of the Area of Interest. The specific contribution of the current paper, regarding traffic modelling, is the study of the modification of the MMORPG's patterns for specific action categories which are affected by differences in server population. The most affected category is *Trading*, due to the previously identified phenomena of players grouping in one or more "hubs", which are usually capital cities [26]. In addition, *PvP combat* traffic shows a strong dependence on the number of players, since this category is designed for a number of players ranging between 4 and 100. To the best of our knowledge, none of the existing traffic models for MMORPGs takes into account the server population (i.e., the number of players on the server).

3. Population and Player-dependant Traffic Model

As previously shown, the population of different servers in MMORPGs varies significantly. The causes of this phenomenon can be various, but they can mostly be attributed to sociological and game related influences. For example, in WoW, the English speaking server *Outland* in Europe (at the time of writing this paper) is one of the most popular servers and "it is well known" among the player population that this is the server where there are a lot of players focused on *PvP combat*. In addition, some servers are designed for different languages (e.g., German, French, etc.) and this fact can also impact their population. Game mechanics such as division of server by types, instability of certain server hardware, migrations between shards, etc., also have an impact on overall server population.

We consider that modelling traffic at application level is a more accurate approach than doing it at packet level, adopting the approach in [16]. In addition, this makes the model independent from the underlying network technology. Thus, we characterize APDU size and IAT, instead of working with packets. In addition, we avoid the simplification of considering that the player always generates traffic with the same statistical distribution: as seen in Fig. 1, traffic strongly varies depending on player's activities. As a consequence, the advanced traffic model uses two steps: we first model the player's behaviour, and then we devise the parameters which rule the traffic generation. A previous model, including the behavioural parameters as well as the teletraffic statistic distributions for each of the action categories, can be found in [10]. As we have already said, we want to construct an improved model which modifies its statistics according to the population of the server.

3.1 Identification of action categories in which traffic depends on population

We now describe the tests we have deployed in order to identify the activities in which the

traffic significantly varies with the number of players in the server. The client generates a traffic flow transmitting the player's actions to the server. At the same time, the server calculates the next state of the virtual world and transmits it to the client. Thus, the server has to communicate to the client the movements and characteristics of the rest of the avatars in its Area of Interest. So logically, the population of the server influences server to client traffic, whereas the information sent by the client is not population dependant.

First, the traffic of *Trading* category has shown its strong dependency on the server population. The reason is that *Trading* is mostly performed in capital cities in WoW, which have been proven to be "hot spots" for player gathering [27], so interaction between players is frequent. Also, points in the virtual world in which the players can perform offline trading through auctioning items (i.e., *Auction Houses*) are located in the big cities. On the other hand, *Questing* is performed in very large areas, so even in highly populated servers there are never more than a few players in the vicinity. The results in [31] also indicate that stay time of players in quest areas is dependent on the level for which the questing area has been designed, further lowering the density of the players in those areas.

Regarding the rest of activities, they are based on "instancing". Instances are areas of the virtual world which are replicated for each group of players who enter them, and the number of players is fixed during the activity. As a consequence, they do not depend on the server population, but on the number of players in the instance. *Raiding* and *Dungeons* are the categories which are defined with a fixed number of players, so we do not model these action categories as dependant of number of players, and we use the models described in [10].

On the other hand, we have modelled *PvP combat* to be aware of the number of players, as each *PvP* area is designed for a different number of players (from four to eighty). It should be noted that the population of the particular server does not have an effect on *PvP combat*; only the population of the specific area in which the player has been assigned has an influence. All in all, we will model the statistics of *Trading* and *PvP combat* as population dependant. In the next subsections we describe in detail the developed statistical models.

3.2 Measurement methodology

The method for measuring the number of players performing a particular activity is similar to the one proposed in [26]: the use of the `/who` call of the WoW command line user interface. This call is used to obtain the list of online players in a certain zone, of specific level, class, or name. As a parameter, we use a name of the zone in which the avatar was located in order to obtain the number of players in the surrounding area.

In order to characterize these activities, we performed measurements of the number of players in two WoW servers with different population: *Bladefist*, a low populated realm, and *Outland*, a server labelled as "full". Measurements were performed between 20:00 and 21:00 hours. This time frame is considered as "prime" time (i.e., when the number of players on the shard is among the highest). The results showed that there are up to six times more players on *Outland*. This difference in the number of players results in significantly different traffic characteristics of server-to-client traffic.

First, we have captured network traces with different numbers of active players in the vicinity. With that aim, we have placed a virtual character in *Stormwind*, a capital city of the *Alliance* faction and performed trading activities (e.g., browsing auction house, visiting bank, checking mail, etc.); we captured network traffic for 30 minutes, and also measured the number of avatars in the area. As a result, we obtained seven traffic traces with different numbers of active players around, from 30 to over 600. We also obtained a trace of a player in

a completely deserted area, with no other players around, in order to obtain the characteristics of traffic when only “keep alive” data is sent.

PvP combat in WoW is a quite complex activity, but it always involves two teams who fight to ensure victory through either killing all the members of the opposing team (arenas), or gathering a certain number of points or objectives depending on the map (battlegrounds). The number of players is fixed to brackets which can be 4, 6, and 10 for arenas and 20, 30 and 80 for battlegrounds. For the purposes of player dependant *PvP combat* modelling, we used 20 traces of arena matches and battlegrounds [8].

3.3 Population dependant teletraffic model of *Trading*

It has been previously noted that Weibull distribution shows the best fit for both APDU size and IAT of the general WoW traffic [10], [16]. While we previously modelled *Trading* category with Lognormal distribution [10], we obtained better results with Weibull distribution when performing fitting for specific number of players. We determined the parameters of the Weibull distribution for each of the captured traces and defined a relationship between those parameters and the number of players. Using non-linear regression, we estimated the parameters of APDU size dependence of number of players in this way:

$$\begin{aligned}\alpha &= 55.067 * N^{0.357} \\ \gamma &= 1.02 + 0.000406 * N\end{aligned}\tag{1}$$

where α and γ are the *scale* and *shape* factors of the Weibull distribution, and N is the number of active players in the AOI.

Trading IAT is described with the following formulae:

$$\begin{aligned}\alpha &= 118,508 + 298,763 * e^{(-0,0119498 * N)} \\ \gamma &= 1.149 * N^{0.068}\end{aligned}\tag{2}$$

As shown in **Fig. 2** (where M stands for “Measured” and G stands for “simulation Generated”), the fit is quite good, especially for the APDU (**Fig. 2a**), while for IAT the trend is captured (**Fig. 2b**), but with more significant discrepancies. For the sake of clarity, we only plot the two measurement cases which we consider borderline (30 and 602 players), and one common case (121 player). The rest of the measurements have between 100 and 500 players average and behave similarly to the depicted curves.

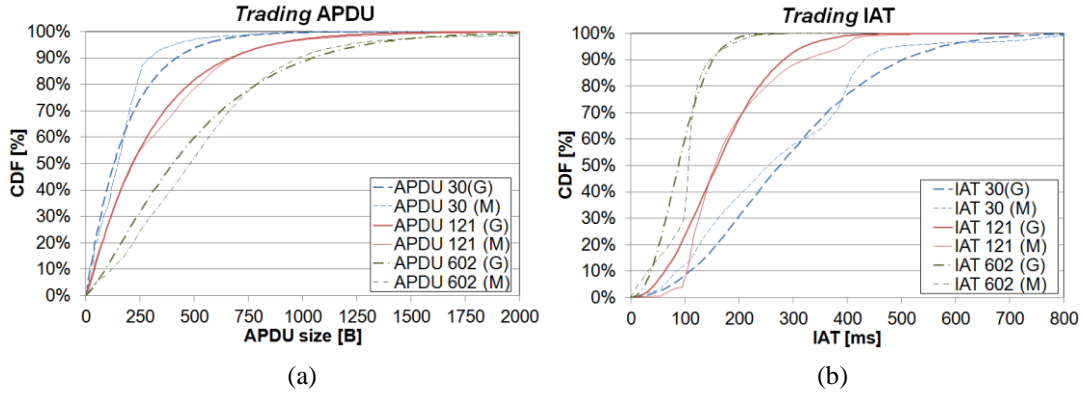


Fig. 2. CDFs of a) APDU and b) IAT for different number of players for measured (M) and simulation generated (G) traces of *Trading*

3.4 Statistical model of *PvP* combat

We show the characteristics of both APDU and IAT for every *PvP* activity in [Fig. 3](#), as well as the respective statistical models (common for IAT, and based on the number of players for APDU). According to the results shown in [Fig. 3b](#), we have not modelled IAT for *PvP* combat as depending on the number of players, since the IAT distribution has shown to be fairly constant for every *PvP* activity, regardless of the number of players. This is due to the fact that *PvP* is a very dynamic action category with constant movement and use of various abilities, which results in very frequent updates from the server, regardless of the number of players [\[8\]](#).

We model APDU size as dependant of the number of players (i.e., the arena bracket or specific battleground). For arenas the modelling procedure is simpler, as they are small areas in which players are constantly in the AOI of the others. Through comparison of arena traces of different brackets, we extracted how much additional data is transferred per additional player, obtaining a mean APDU increase of 38 bytes per player. Thus, we fit the APDU size to a Weibull distribution dependant on the number of players. The formula is as follows:

$$\begin{aligned} \text{Mean}(N) &= \text{Mean}(4) \cdot 38 \cdot (N-4) \\ \alpha &= \text{Mean}(N) / 1.178 \\ \gamma &= 0.76 \end{aligned} \tag{3}$$

where $\text{Mean}(i)$ is the mean of the distribution describing i players, and $\text{Mean}(4) = 258.33$ (estimated from the measurements).

For battlegrounds the case is more complex, as they are larger in terms of virtual space, and not all players are always in the AOI of all other players. That is why we first estimate the average number of players in the AOI, depending on the battleground. The estimation is based on the measurement traces of each particular battleground and the values obtained from measurements of arenas in which we know exactly how many players are in the AOI. The estimated values of the average number of users are listed in [Table 1](#).

Table 1. Estimated Average Number of Players per Battleground

Battleground	Average number of players
Alterac Valley	21.32
Arathi Basin	10.90
Warsong Gulch	10.11
Eye of the Storm	13.14
Strand of the Ancients	15.03

The formula for battlegrounds has also been slightly modified in comparison with arenas, as in arenas there is constant fight until someone “dies”, whereas in battlegrounds there are also time periods in which players wait to be “reborn” after they have been killed, and time periods in which the player is travelling from one end of the battleground to another, etc. These differences reflect on the characteristics of the Weibull distribution. The final battleground formula for server APDU sizes is:

$$Mean(N) = Mean(4) \cdot 38 \cdot (N-4) \quad (4)$$

$$\alpha = Mean(N) / 1.073$$

$$\gamma = 0.87$$

where the parameter N is representing an estimated average number of players in the AOI for the battleground and $Mean(4) = 258.33$. In Fig. 3 the results of the model are plotted versus the measured data for cases of 2v2 and 5v5 arenas and for *Alterac Valley* (AV) battleground. As it can be observed, the models tend to slightly overestimate the empirical distribution, but the general trend is captured well.

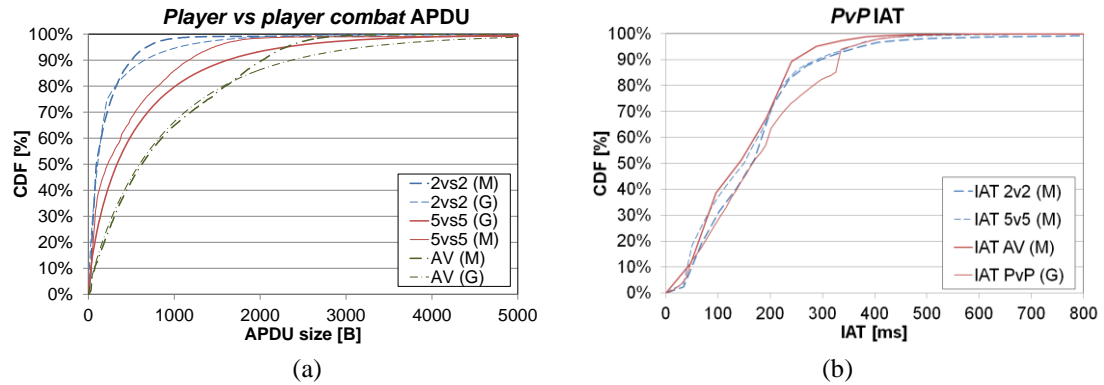


Fig. 3. CDFs of a) APDU b) IAT, for different number of players for measured (M) and simulation generated (G) traces of *PvP combat*

As *PvP combat* has been fractionated into arenas and battlegrounds, and some parameters of the model are dependent on the battleground type, we have to fully adapt the behavioural model. We make two assumptions here: first, we consider the duration of the *PvP* activity as independent of whether it is a battleground or an arena; second, we assume a battleground/arena ratio of 50%:50%, since there is no reliable data based on which to

estimate these parameters.

Our model makes the decision of a player entering a specific bracket of arena or battleground, based on data gathered by WoW add-on census (the dataset and the add-on are further described on <http://www.warcraftrealms.com>). The parameters of popularity of each particular arena bracket or battleground are displayed in **Table 2** and **Table 3**, respectively. The popularity parameter describes how often players join specific battleground map or specific arena bracket.

Table 2. Popularity of arena brackets

Arena	Popularity
2v2	45.05%
3v3	45.16%
5v5	9.79%

Table 3. Popularity of specific battlegrounds

Battleground	Popularity
Arathi Basin	24.56%
Warsong Gulch	32.50%
Eye of the Storm	7.96%
Strand of the Ancients	2.08%
Alterac Valley	32.89%

3.5 Implementation of the advanced model

Once we have obtained the statistics of the traffic for each activity, the model has been fully implemented in NS2. It includes the probability of a player deploying each activity, which varies with the hour of the day. This script, allowing a wide range of tests, is offered to the research community. As an example of the use of the model, **Fig. 4** shows the daily behavioural variation obtained when simulating 3,000 players during a whole day. It can be seen that it does capture the main trends in player behaviour such as the high rise of *Raiding* in the evening [10]. The slight discrepancies with respect to previous results [10] are mainly caused by the fact that we do not model the fluctuation of the players during the day.

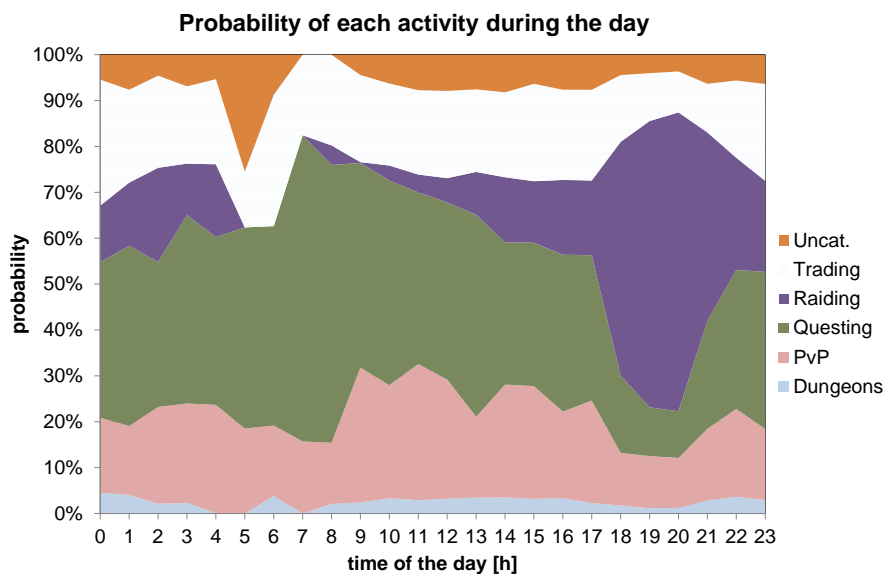


Fig. 4. Daily pattern of players' activities

All in all, as an example of the results of the developed model, **Fig. 5a** shows the client-server traces generated by the NS2 script, for each activity. To illustrate the bandwidth difference depending on the number of players and activity, **Fig. 5b** shows the effect of the server population for server-client traffic of *Trading* and the same effect depending on the different scenarios for *PvP combat*. As it can be seen, the script captures the differences in traffic characteristics for particular player behaviour (note that **Fig. 5b** is not comparable with **Fig. 1b**, since it only shows the activities that modify their behaviour with the number of players).

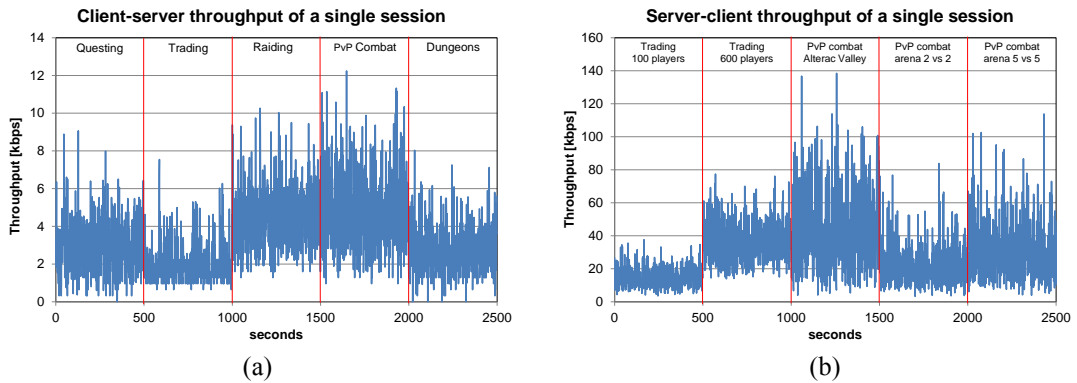


Fig. 5. Throughput generated by the model for: a) client to server for each category; b) server to client for *Trading* and *PvP combat* with different options

4. Tests and Results

In this section we present the results of different experiments using the complete NS2 traffic model described previously, with the aim of illustrating the behaviour of these flows when they share the network with other traffic. Thus, we will mainly focus on the competition between application-limited TCP game traffic and other flows, and how this competition modifies the communication parameters. The considered competing flows are FTP, which use TCP so as to get as much throughput as possible (network-limited), and UDP Constant Bit Rate (CBR), which generate the same throughput despite the status of the network.

The obtained results will be presented terms of the most interesting parameter for each kind of traffic:

- For the MMORPG TCP traffic we will mainly focus on Round Trip Time (RTT), taking into account that latency is the most important parameter, because the interactivity of the game mainly depends on it. In order to estimate the RTT in NS2, we will use the parameters that govern TCP dynamics (e.g., retransmissions), namely “*smoothed RTT*” and “*RTT variation*”, which are calculated and updated frequently, according to the network conditions. They are subsequently used to obtain the value of Retransmission Timeout (RTO) [32]. If this timeout expires, the packet is retransmitted.
- For the FTP background traffic, we will present the achieved throughput, since these flows try to get as much as possible of the bandwidth share. The size of the sending window of TCP will also be presented if required.
- For the CBR traffic, the results can be presented in terms of bandwidth or packet

loss rate which, are directly related in this case, since the traffic is sent in an open loop.

In order to create a scenario where the different flows share a bottleneck, we have configured a dumbbell topology (Fig. 6): two pairs of client-server connections (*A* and *B*) correspond to game nodes and the other pair is used for generating background traffic.

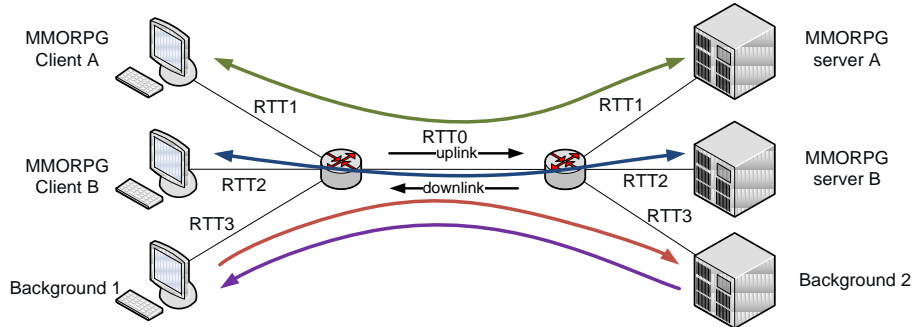


Fig. 6. Network topology used in the tests

Regarding the parameters of the simulations, when activity exchange is not explicitly required by the test, we will use by default *Questing* traffic, one of the most popular activities, which also presents a relatively stable traffic profile. The advantage of using a single activity in those tests is that we avoid the influence of player behaviour, which may obscure the observations. In the last subsection we include activity exchange so as to better observe the characteristics of the realistic traffic generated through the model. Each *Questing* flow generates about 2.2 kbps in the client-server connection, and roughly 18 kbps in the server-client one. By default, the bandwidth in the bottleneck is 10 Mbps in both directions. Queue sizes are 100 packets. The Round Trip Time (RTT) delay of the bottleneck (RTT_0) is 40 ms. By default, the rest of RTT_x delays are set to 0. Each simulation lasts 200 seconds, and a “tick” of 1 second is used to calculate the average RTT or throughput during each interval. The tick is 0.1 seconds for the TCP window size.

4.1 Competition of MMORPG and FTP flows: the effect of TCP variants

In this subsection we will study the competition of application-limited (MMORPG) and network-limited TCP flows (we will use FTP as a typical application). To this aim, 100 MMORPG sessions are established between *client A* and *server A*, and 10 FTP upload and 10 FTP download sessions are set between *background 1* and *2*, with a packet size of 1,500 bytes.

In order to correctly imitate the behaviour of WoW, which uses a single session, piggybacking the ACKs in packets in the opposite direction [16], the *Full-TCP* NS2 TCP implementation (using *Reno*) has been used for game sessions. For the FTP sessions, five TCP variants are tested: *Tahoe*, *Reno*, *New Reno*, *SACK* and *Vegas*, using the standard NS2 implementations for these protocols. Each TCP session is started using a different random delay, in order to avoid the effect of the synchronization of TCP window sizes [30]. Thus, the first seconds of the presented graphs may present a transient behaviour different from the stationary.

The effect of the different TCP variants used for the background flows can be observed in Fig. 7 (on behalf of clarity, only *New Reno*, *SACK* and *Vegas* are shown), where the *smoothed RTT* is presented. Table 4 presents the average RTT on each case, between all the game flows.

This RTT increase is caused by queuing delay at the bottleneck. These results are complemented with those in [Fig. 8, 9](#) and [10](#), which present the aggregate throughput obtained by the FTP flows, and the TCP window size for *New Reno*, *SACK* and *Vegas*. Finally, [Table 5](#) summarizes the FTP throughput results for all the testes TCP variants.

It can first be observed that when no background traffic is present, the estimated latency is roughly 40 ms. The behaviour of *New Reno* and *SACK* is very similar, but it can be seen that game traffic obtains better results when *SACK* is used for the background traffic: 7 ms are reduced from the RTT, at the cost of a slighter share of the bandwidth of the link for the FTP flows (an average of 849 kbps instead of 874 kbps per flow, i.e. a 3 % throughput reduction). The reason for this can be that *SACK* reduces the retransmissions, since its ACK mechanism is able to acknowledge packet ranges, so the behaviour results a bit less aggressive. The results for *Tahoe* and *Reno* are very similar to those of *New Reno*.

TCP *Vegas* behaves in a very different way, as shown in [Fig. 10](#): the window size does not grow aggressively, but it remains constant. Because of this *timid* behaviour [\[34\]](#), when TCP *Vegas* notices the RTT increase, it maintains its sending rate, even in the absence of packet loss. In the uplink, *Vegas* only gets 814 kbps (which means a reduction of 9 % with respect to *New Reno*), and in the downlink the reduction is roughly 10 %. However, as a counterpart, this is translated into a significant reduction of the RTT of the MMORPG flows: only 14 ms are added to the 40 ms of default RTT, whereas in *New Reno* this figure is 44 ms.

It should be noted that in all these cases the game is still playable, taking into account that the literature has reported that MMORPG players can tolerate up to some hundreds of milliseconds of RTT [\[35\]](#).

As a result, we can conclude that the most popular TCP variants (i.e., *New Reno* and *SACK*) can be respectful with competing MMORPG traffic, although they add some latency due to their behaviour trying to get the maximum bandwidth share. On the other hand TCP *Vegas* could be considered as a good option for competing flows in order not the throttle the MMORPG flows and by that to better preserve the quality of game experience for the MMORPG players.

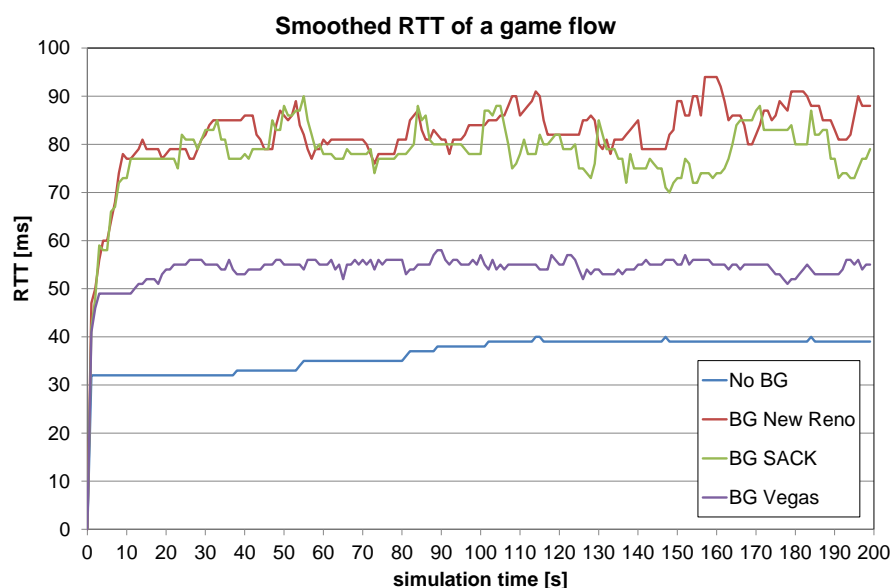


Fig. 7. Competition MMORPG vs FTP traffic for different TCP variants: smoothed RTT of a game flow

Table 4. Average RTT between all the game flows when different TCP variants are used for the background traffic

BG TCP variant	avg smoothed RTT	avg RTT variation
No BG	38.76 ms	3.05 ms
<i>Tahoe</i>	82.61 ms	7.24 ms
<i>Reno</i>	81.23 ms	7.36 ms
<i>New Reno</i>	84.54 ms	6.76 ms
<i>SACK</i>	77.64 ms	6.85 ms
<i>Vegas</i>	54.74 ms	3.65 ms

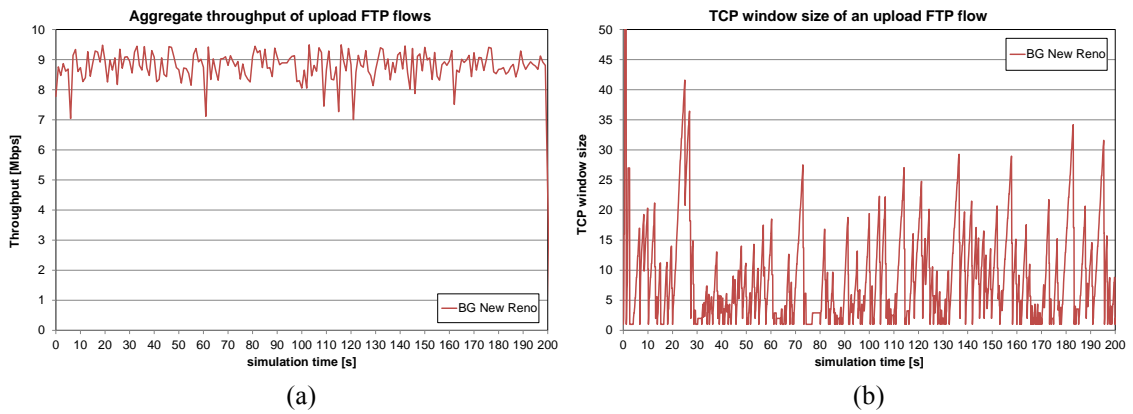


Fig. 8. Competition between MMORPG and FTP background traffic using TCP *New Reno*: a) aggregate throughput of the 10 FTP flows; b) evolution of the TCP window of one of the flows

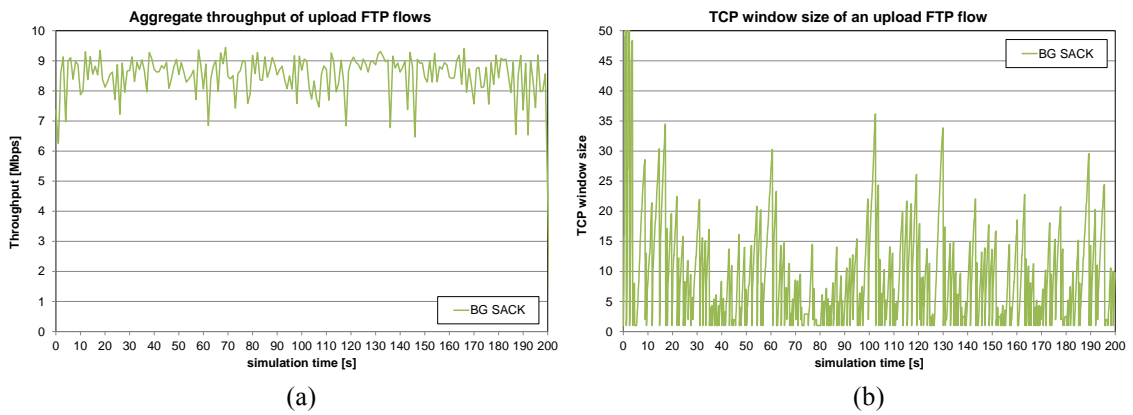


Fig. 9. Competition of MMORPG and FTP background traffic using TCP *SACK*: a) aggregate throughput of the 10 FTP flows; b) evolution of the TCP window of one of the flows

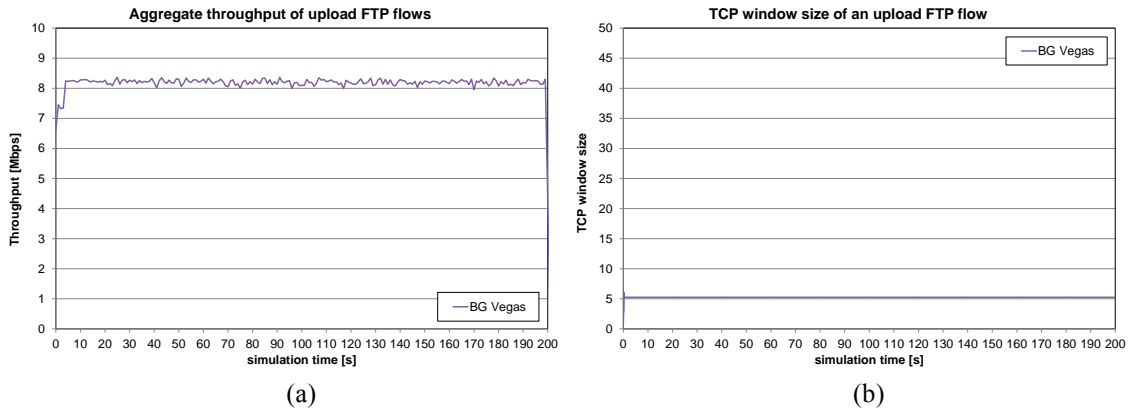


Fig. 10. Competition of MMORPG and FTP background traffic using TCP *Vegas*: a) aggregate throughput of the 10 FTP flows; b) evolution of the TCP window of one of the flows

Table 5. Average throughput obtained by the FTP background flows using different TCP variants

BG TCP variant	avg throughput uplink	avg throughput downlink
<i>Tahoe</i>	871.15 kbps	736.08 kbps
<i>Reno</i>	866.58 kbps	746.53 kbps
<i>New Reno</i>	874.58 kbps	749.85 kbps
<i>SACK</i>	849.84 kbps	717.88 kbps
<i>Vegas</i>	814.76 kbps	672.24 kbps

4.2 Competition of MMORPG and UDP background traffic

It is normally assumed [33] that, when TCP and UDP flows compete for a shared link, UDP gets all the required bandwidth while TCP uses the remainder of the bandwidth, because of TCP flow control mechanism. However, in the case of MMORPGs, in which the traffic is not limited by the network, but is application-limited, a different behaviour can be observed.

We have set this scenario: 100 MMORPG *Questing* sessions (roughly 2Mbps in the server-client direction) are run from *client A* to *server A*. At the same time, three UDP flows are sent between *background 1* and *2*. The total amount of UDP traffic is the aggregation of these three flows: small packets (40 bytes, 50% of the packets), medium packets (576 bytes, 10% of the packets) and large packets (1,500 bytes, 40% of the packets) [36].

In **Fig. 11**, 7Mbps of UDP traffic share the link with 100 MMORPG sessions. Since there is enough bandwidth, each flow obtains the required amount. The first row of **Table 6** summarizes the results: first, it can be seen that the RTT of the MMORPG connections is not affected, since the obtained value is similar to the result obtained when no background traffic was present (**Table 4**). In addition, it can be seen that the packet loss rate of the CBR traffic is residual. When 8Mbps of UDP CBR traffic are sent (**Fig. 12**), the average RTT rises up to 48 ms. At the same time, the packet loss rate of the UDP flows is slightly increased.

Table 6. Summary of the tests: average RTT of all the MMORPG flows; packet loss rate in the downlink

CBR throughput (downlink)	Average RTT MMORPG	Average packet loss rate CBR
7 Mbps	39.35 ms	0.05 %
8 Mbps	48.16 ms	0.34 %
8.5 Mbps	38.58 ms	4.61 %
9 Mbps	76.78 ms	11.67 %

However, when UDP traffic is 8.5Mbps (Fig. 13) and the total offered traffic is above the bandwidth limit, the tendency is that MMORPG traffic maintains its bandwidth use of roughly 2Mbps, whereas UDP traffic only obtains the remaining throughput.

This is also observed in Fig. 14, where the offered UDP traffic is 9Mbps. The bandwidth obtained by the UDP flows is roughly the same as in the previous case, whereas MMORPG maintains its bandwidth, with a slight increase of 3.22 %, caused by duplicate game packets. All in all, it can be seen that MMORPG traffic flows roughly obtain the same throughput, despite the amount of competing UDP traffic.

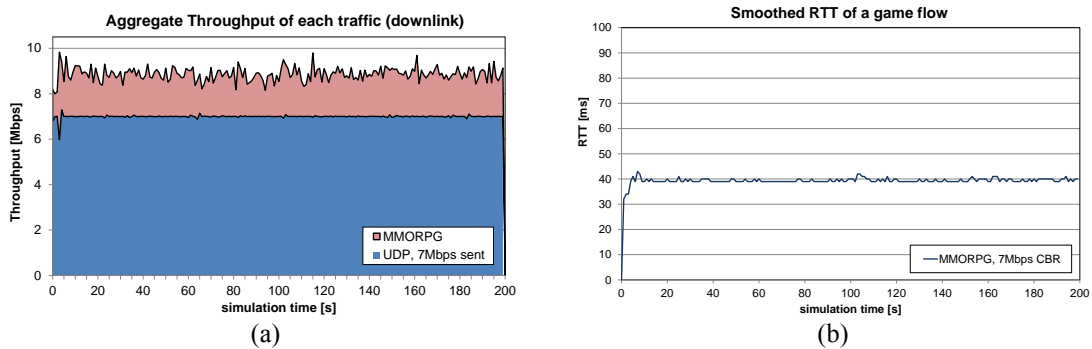


Fig. 11. Coexistence of 100 MMORPG flows and 7 Mbps of UDP CBR traffic: a) received throughput in the downlink by each traffic; b) smoothed RTT of one of the game flows

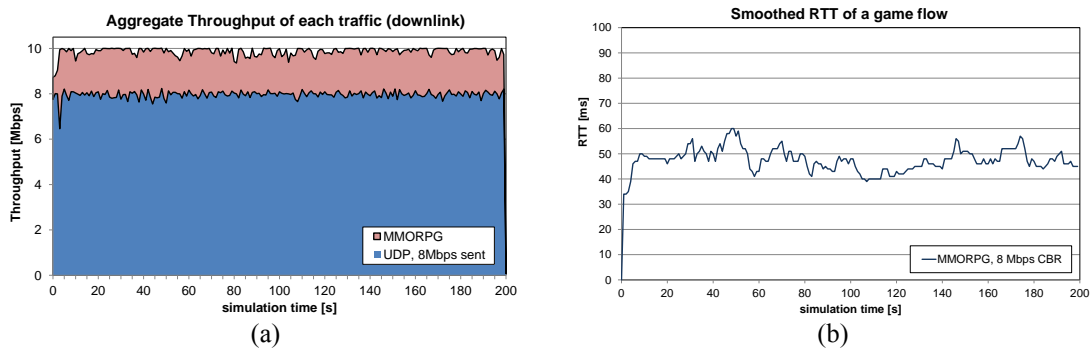


Fig. 12. Coexistence of 100 MMORPG flows and 8 Mbps of UDP CBR traffic: a) received throughput in the downlink by each traffic; b) smoothed RTT of one of the game flows

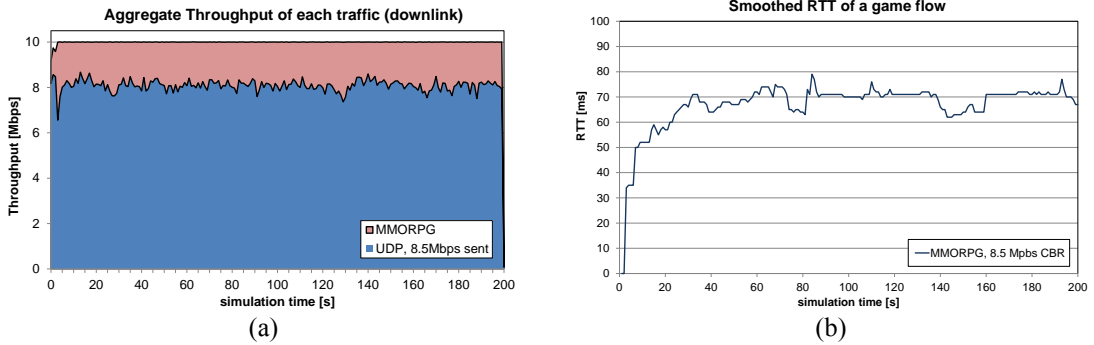


Fig. 13. Coexistence of 100 MMORPG flows and 8.5 Mbps of UDP CBR traffic: a) received throughput in the downlink by each traffic; b) smoothed RTT of one of the game flows

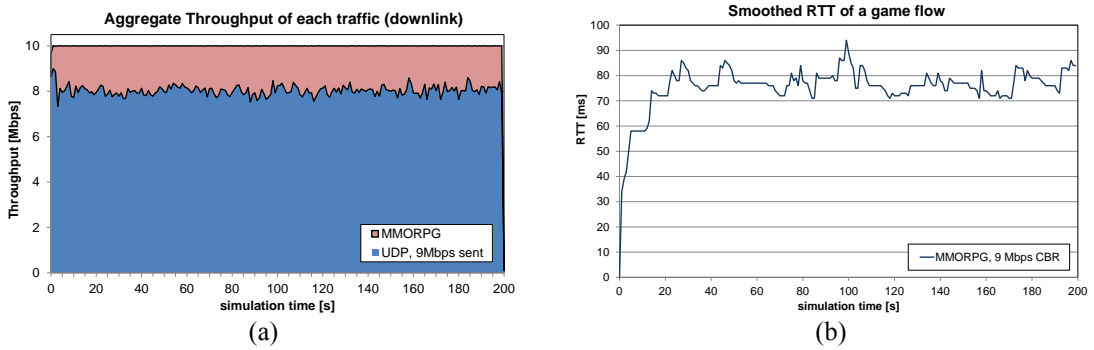


Fig. 14. Coexistence of 100 MMORPG flows and 9 Mbps of UDP CBR traffic: a) received throughput in the downlink by each traffic; b) smoothed RTT of one of the game flows

All in all, this behaviour is the opposite of that normally expected in the TCP-UDP competition [34], where TCP only obtains the traffic not used by UDP. The reason for this is that the bandwidth demand of the MMORPG flows is not governed by the congestion window, since they do not have an amount of traffic ready to be transferred, but keep on generating new information while the game evolves. In order to illustrate this, Fig. 15a shows the TCP window of an MMORPG session, in the case where no background traffic is present (negligible queuing delay), and (Fig. 15b) in the case of having 8.5 Mbps of UDP background traffic.

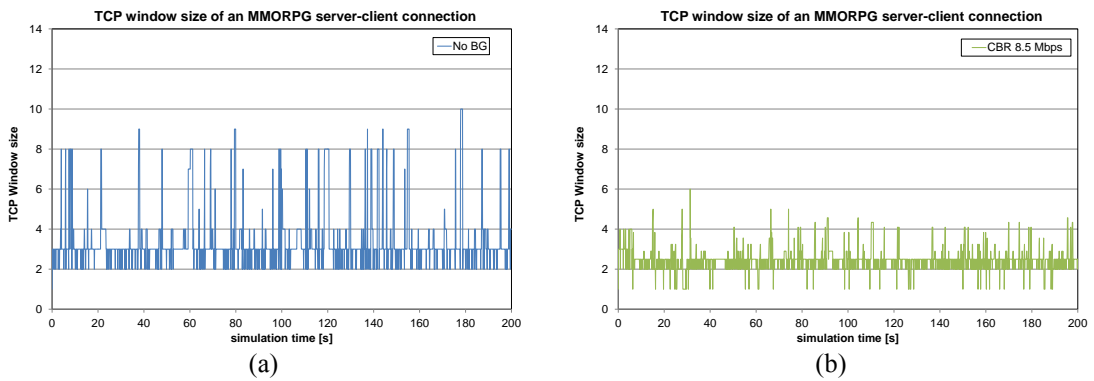


Fig. 15. TCP window size, server-client MMORPG traffic: a) no background; b) 8.5Mbps background

Although the size of the TCP window is reduced when background traffic is present, the flows keep on sending their traffic, since they do not need a high value of the TCP window in order to send it. This fits with the two opposed effects reported in [4]: on the one hand, the thinness of each stream makes that few packets are sent by RTT, making the window hardly grow. On the other hand, the flows do not correctly react to congestion, so they may keep on sending the same amount of traffic.

4.3 RTT unfairness tests

The aim of this subsection is the study of RTT unfairness between different MMORPG TCP flows. As reported in the literature [35], when two TCP flows with different RTTs compete for the same bottleneck link, the one with a smaller RTT has an advantage with respect to the other. The cause of this is that the most popular TCP versions (*New Reno*, *SACK*) increase their sending windows according to the received ACKs, so different delays will make them increase their rates differently.

However, in the case of MMORPG flows, we cannot expect a modification in the throughput, since the traffic is application-limited instead of network-limited. As we have already said, the main figure of merit for MMORPGs is RTT. Hence, an increase of the network RTT should be translated into a direct increase of the *smoothed RTT* seen by TCP. So the aim of this subsection is to check if there is any additional impairment caused by RTT unfairness, leaving apart the RTT increase itself.

In order to answer this question, we have set the parameters of the scenario this way: 100 MMORPG flows are established between *client A* and *server A*, and at the same time, 100 MMORPG flows are active between *client B* and *server B*. Background traffic consists of 10 FTP upload and 10 FTP download sessions using TCP *SACK*, between *background 1* and *background 2* nodes. The value of *RTT1* (Fig. 6) is set to different values, higher than 0, so the flows having a higher RTT, and thus the flows between *client A* and *server A* are at a disadvantage. The RTT of the bottleneck (*RTT0*) is always 40ms.

Fig. 16 presents the results obtained when 100 game flows from *client A* to *server A*, experiencing different amounts of RTT (from 50 to 90), share the bottleneck with 100 game flows from *client B* to *server B*, always with an RTT of 40 ms. Regarding the value of *smoothed RTT*, two things can be appreciated in Fig. 16a: first, the difference between the columns is roughly the difference in terms of network RTT: for example, when 100 flows with an RTT of 60 ms share the bottleneck with 100 flows with an RTT of 40 ms, the difference in terms of *smoothed RTT* is 21.85 ms (i.e., roughly 20 ms). This happens (with very small variations) for all the values. Second: the RTT value for the *client B* to *server B* connections always remains the same.

Regarding the results of *RTT variation* (Fig. 16b), it can be appreciated that the variation of the RTT is significantly higher for the flows experiencing a higher RTT, which will be translated in high jitter values.

The results of the throughput are not shown here, since the throughput variations between *A-A* and *B-B* flows are negligible (up to 3%), as it could be expected. We can conclude saying that *smoothed RTT* is not affected by this unfairness, but only *RTT variation* is worse for the flows with a higher RTT.

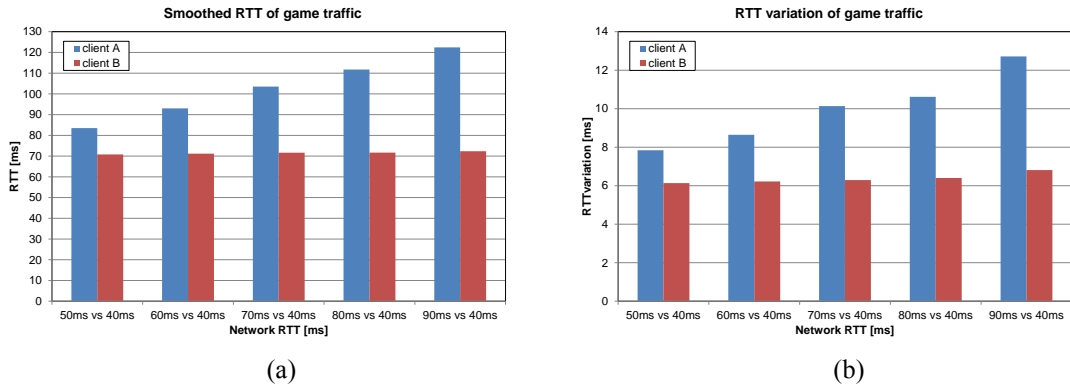


Fig. 16. Latency of MMORPG flows from *client A* and *client B* a) *Smoothed RTT* b) *RTT variation*

4.4 The effect of server population on the aggregate traffic

We have previously mentioned the importance of the number of players present on a server, since this can have an effect on the global traffic. This subsection uses the proposed model to confirm that the traffic can significantly vary according to the server population, and also depending on the hour of the day. The developed model takes into account both variables.

In the tests of this subsection we have established 100 MMORPG traffic flows between *client A* and *server A*, competing with 10 FTP upload flows, 10 FTP download flows (using TCP *SACK*) and UDP background traffic of 6 Mbps in both uplink and downlink. We have considered that the players are using low-populated servers (30 players) in one case, and high-populated ones (600 players) in the other. Taking into account that the two activities in which traffic significantly varies with the server population are *PvP combat* and *Trading*, we have selected three different hours of the day (see [Fig. 4](#)):

- 8:00 as an example of low *PvP combat* and high *Trading*;
- 11:00 where both activities have a high probability of being performed; and
- 20:00 where both activities have low probability.

In this case, the aggregate traffic (100 MMORPG flows) is composed of flows corresponding to the six action categories, according to their probabilities, which also depend on the hour of the day. In this way, we investigate if the variation of the APDU and IAT of *Trading* and *PvP combat* has a global effect, which can be observed on the aggregate server-to-client traffic.

The results are shown in [Fig. 17](#), where 200 simulation seconds are run for each case. The left figure displays the case of players in low populated servers, while the right displays the case of highly populated servers. During the initial seconds, there is a transient status in which FTP connections need some time in order to increase their sending windows. As a first observation, we see that the throughput of these 100 MMORPG sessions strongly varies with the hour of the day: at 8:00 it can vary from 2 to 2.4 Mbps, depending on the global population of the servers where the players are. However, at 20:00 it varies from 3.5 to 4.1 Mbps. It should be remarked that these variations are not caused by a different number of flows, but only by the global population of the servers to which these players are connected.

At the same time, it can be observed that at 11:00 ([Fig. 17c](#) and [Fig. 17d](#)), the difference between the traffic is significant: 2.18 Mbps with 30 players, and 2.96 Mbps with 600 players (35%). At 20:00 ([Fig. 17e](#) and [Fig. 17f](#)), the difference is only from 3.5 to 4.1 (17%). At 8.00 ([Fig. 17a](#) and [Fig. 17b](#)) the traffic ranges from 2 to 2.44 Mbps (22%).

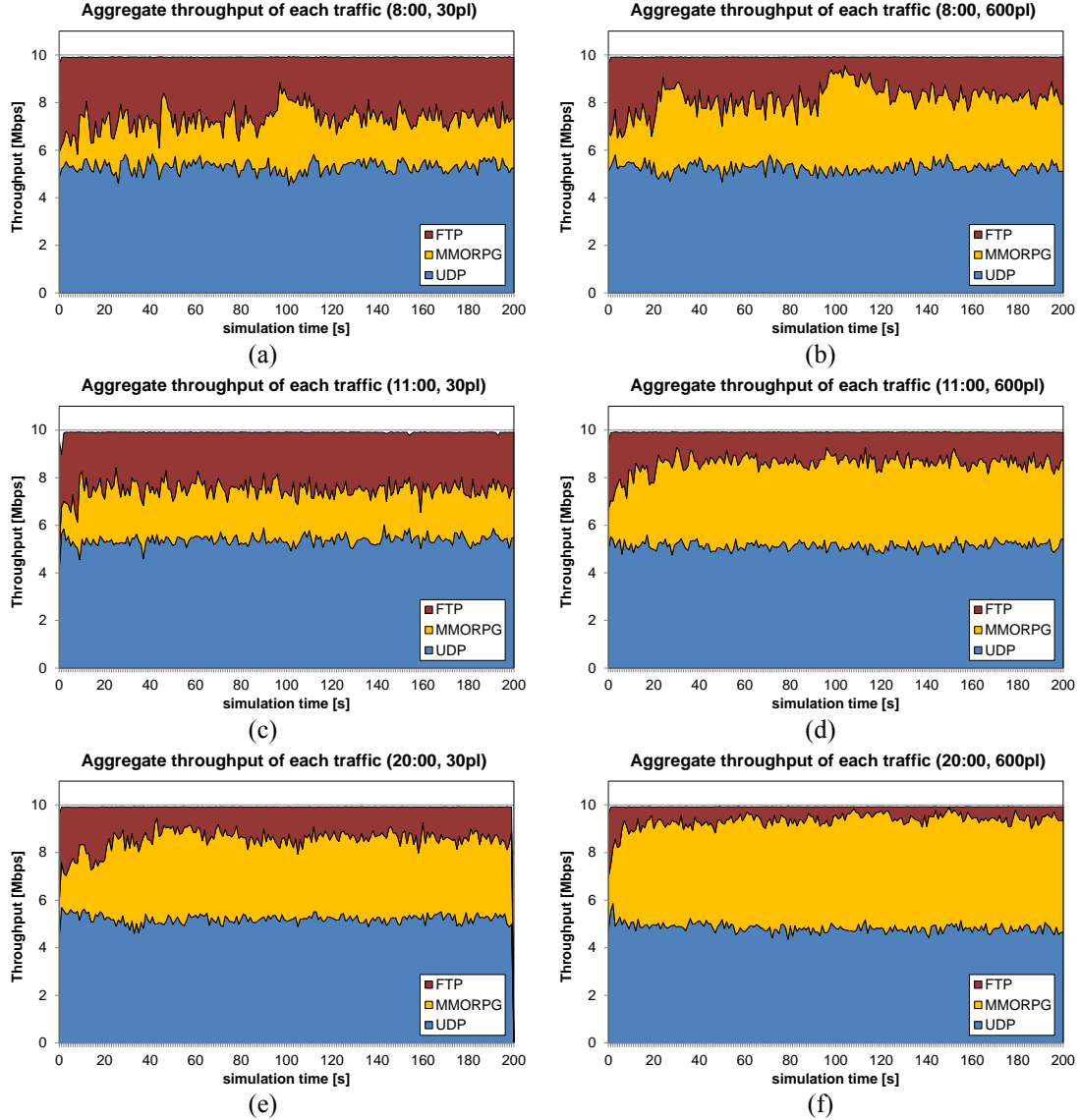


Fig. 17. Received throughput in the downlink by 100 MMORPG flows, 10 FTP and 6 Mbps of UDP traffic a) 8:00, 30 players in the servers; b) 8:00, 600 players in the servers; c) 11:00, 30 players; d) 11:00, 600 players; e) 20:00, 30 players; f) 20:00, 600 players

Finally, in order to explore if the variation of the bandwidth with the hour of the day has any influence on the RTT (measured as *smoothed RTT*), **Table 7** shows its average value for the 100 MMORPG flows. It can be seen that the variations are really small (between 1 % and 4 %), and this fits with another phenomenon that can be observed in **Fig. 17**, i.e. we see that in all the cases the UDP flows get reduced their 6Mbps of bandwidth share; whereas MMORPG flows maintain their throughput. FTP flows are only able to get the bandwidth that the two other kinds of flows leave free. This is in concordance with the results shown in Subsections 4.1 and 4.2, and with the effect reported in [4], i.e. the flows do not react to congestion due to their thinness, and they keep on sending the same amount of traffic.

Table 7. Average *smoothed RTT* of all the MMORPG flows

Hour of the day	<i>Smoothed RTT</i> 30 players	<i>Smoothed RTT</i> 600 players
8:00	91.39 ms	87.69 ms
11:00	90.46 ms	89.23 ms
20:00	93.61 ms	96.32 ms

The high variability of MMORPG traffic depending on the hour of the day and on the population of a server makes it difficult to calculate the number of servers required for provisioning an online game. In addition, many other factors have an influence, as e.g., the release of a new game or of new content of an existing one can cause a traffic rush. Although game developers may experience difficulties when predicting the success of a game, they may be interested on using statistical models in order to predict the demand variations according to the hour of the day.

4.5 Discussion of the results

This subsection summarizes and discusses the results presented in this section. In the first tests, we measured the competition of MMORPG and FTP flows using different variants of TCP. We first observed that the MMORPG flows are able to work properly even with network-limited TCP connections in the background. The most important parameter for players, namely RTT can be kept low for all the TCP variants. TCP *SACK* shows a more respectful behaviour with game flows: it increases less the game RTT, at the cost of achieving a slightly lower amount of bandwidth share than *Tahoe*, *Reno* and *New Reno*. This tradeoff is more accentuated with TCP *Vegas*, which reduces even more its throughput, but as a counterpart adds less delay (up to 30 ms) to the RTT of the game flows.

The second subsection has studied the mutual influence of MMORPG TCP flows and UDP CBR ones. In contrast with what normally happens with TCP flows, which only get the bandwidth share not used by UDP ones, the tests have shown that TCP game flows are able to maintain their throughput despite the amount of UDP traffic. The reason is that the bandwidth demand of the game flows is not governed by the congestion window, since they do not have an amount of predefined data to transfer, but they keep on generating new information according to the player actions and the game evolution.

The third battery of tests has explored the RTT unfairness, normally observed when several TCP flows with different RTTs share a bottleneck. This difference is not as important as for network-limited TCP flows. In this case, the bandwidth obtained does not vary with the RTT, taking into account that the traffic is mostly signalling. In addition, the RTT experienced by the game (measured in terms of TCP *smoothed RTT*) varies according to the real network RTT. Only the *RTT variation* shows worse results for the flows with a higher RTT, which will be translated into a higher jitter in the game traffic.

Finally, we have used the developed model of the game traffic so as to obtain results illustrating the effect of the number of players in a server (server population). Different hours of the day have been selected, drawing some conclusions that mainly affect server-to-client traffic: the bandwidth may vary up to 35 % depending on the population of the server, and this difference may also vary according to the hour of the day. The cause of this phenomenon is that in some moments, the activities preferred by the players are more server-population

dependant. In addition, it has again be observed that game flows are able to maintain their throughput and a reasonable RTT, even in the presence of high amounts of combined FTP and UDP traffic flows.

5. Conclusion

This paper has studied the interactions between application-limited TCP traffic, typical of MMORPGs, and other flows. In order to do this, an advanced traffic model of a popular MMORPG game has been first developed. By using measurements deployed in real servers of the game, we have analyzed how the APDU and IAT of some activities vary with the number of players on the server, and this effect has been included in the statistics that rule the traffic generation in the model. In addition, the model is able to simulate user behaviour, generating traffic according to different activities, which have different probabilities depending on the hour of the day.

The traffic model has also been implemented in a network simulator, and a set of tests of the coexistence between traffic flows have been deployed: the coexistence of the game traffic with FTP using different TCP variants; the competition with UDP flows; the effect of different values of RTT between game flows; and the global effect of the server population on the aggregate traffic. It has been shown that TCP *SACK* is more respectful with game flows than *Tahoe*, *Reno* and *New Reno*, i.e. it achieves a slightly lower throughput but adds less delay to MMORPG flows. Furthermore, the behaviour of TCP *Vegas* is even better, not throttling the MMORPG flows and causing no growth on their RTT, since it does not increase aggressively its window size. Interestingly, we found that MMORPG flows are resilient to UDP traffic: since they do not need significant sizes of the TCP window, they are able to maintain their bandwidth share while maintaining their RTT in reasonable limits. The RTT unfairness for application-limited TCP traffic has also been studied, showing that only the RTT variation is affected. Finally, we have confirmed that the player population on the server has significant impact on the overall traffic (up to 30 %). Although game developers may experience difficulties when predicting the success of a new game, the use of statistical models able to predict the demand variations according to the hour of the day is seen as very convenient. As future work, we plan to improve the model so as to include players' arrivals and departures. In addition, the influence of different buffer sizes and policies on the coexistence of MMORPG traffic with other flows will be studied.

Acknowledgments

This work has been partially financed by the Project "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; the European Community Seventh Framework Programme under grant agreement no. 285939 (ACROSS); CPUFLIPI Project (MICINN TIN2010-17298); Project TAMA, Government of Aragon; Project Catedra Telefonica, Univ. Zaragoza; European Social Fund in collaboration with the Government of Aragon. The authors would like to thank John Miller for his advice regarding some details of the game traffic. We also want to thank Tanja Kauric for her help in obtaining the traffic traces of the game.

References

- [1] Activision Blizzard, "World of Warcraft subscriber base reaches 12 million worldwide," Oct. 2010, Available: <http://us.blizzard.com/en-us/company/press/pressreleases.html?id=2847881>
- [2] I. V. Geel, "MMOData: Keeping track of the MMORPG scene," Aug. 2013. Available: <http://mmodata.net/>
- [3] 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, pp. 7-32, 2009.
- [4] C. Griwodz, and P. Halvorsen, "The fun of using TCP for an MMORPG," *Proceedings of international workshop on Network and operating systems support for digital audio and video (NOSSDAV '06)*, pp. 1 - 7, New York, NY, USA, 2006.
- [5] K-T. Chen, C-Y. Huang, P. Huang, and C-L. Lei, "An Empirical Evaluation of TCP Performance in Online Games," *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology (ACE 06)*, p. 8, Hollywood, CA, USA, 2006.
- [6] 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.
- [7] 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.
- [8] M. Suznjevic, I. Stupar, and M. Matijasevic, "MMORPG player behavior model based on player action categories," *Proceedings of the 10th Workshop on Network and System Support for Games (NetGames '11)*, p. 6, Ottawa, Canada, 2011.
- [9] 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.
- [10] M. Suznjevic, I. Stupar, and M. Matijasevic, "A model and software architecture for MMORPG traffic generation based on player behavior," *Multimedia Systems*, Vol. 19, No. 3, pp. 93-101, 2012.
- [11] B. De Vleeschauwer, B. Van Den Bossche, T. Verdickt, F. De Turck, B. Dhoedt, and P. Demeester, "Dynamic microcell assignment for massively multiplayer online gaming," *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games (NetGames '05)*, pp. 1–7, Hawthorne, NY, USA, 2005.
- [12] X. Che, and B. Ip, "Review: Packet-level traffic analysis of online games from the genre characteristics perspective," *Journal of Network and Computer Applications*, Vol. 35, No. 1, pp. 240–252, 2012.
- [13] M. Suznjevic, and M. Matijasevic, "Player behavior and traffic characterization for MMORPGs: A survey," *Multimedia Systems*, Vol. 19, No. 3, pp. 199-220, 2012.
- [14] C. Majewski, C. Griwodz, and P. Halvorsen, "Translating latency requirements into resource requirements for game traffic," *Proceedings of the International Network Conference (INC)*, pp. 113 – 120, Plymouth, UK, 2006.
- [15] J. Saldana, M. Suznjevic, L. Sequeira, J. Fernandez-Navajas, M. Matijasevic, and J. Ruiz-Mas, "The Effect of TCP Variants on the Coexistence of MMORPG and Best-Effort Traffic," *Proceedings of IEEE ICCCN 2012, 8th International Workshop on Networking Issues in Multimedia Entertainment (NIME'12)*, pp. 1 - 5, Munich, Germany, 2012.
- [16] P. Svoboda, W. Karner, and M. Rupp, "Traffic analysis and modeling for World of Warcraft," *Proceedings of IEEE International Conference on Communications (ICC '07)*, pp. 1612–1617, Glasgow, Scotland, 2007.
- [17] J. Kim, E. Hong, and J. Choi, "Measurement and Analysis of a Massively Multiplayer Online Role Playing Game Traffic," *Proceedings of Advanced Network Conference*, pp. 1–8, 2003.
- [18] Y. Wu, H. Huang, and D. Zhang, "Traffic Modeling for Massive Multiplayer On-line Role Playing Game (MMORPG) in GPRS Access Network," *Proceedings of the International Conference on Communications, Circuits and Systems*, pp. 1811–1815, Guilin, China, 2006.

- [19] V. Paxson, "Empirically-Derived Analytic Models of Wide-Area TCP Connections," *IEEE/ACM Transactions on Networking*, Vol. 2, No. 2, pp. 316–336, 1994.
- [20] 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.
- [21] H. Park, T. Kim, and S. Kim, "Network traffic analysis and modeling for games," *Lecture Notes in Computer Science Internet and Network Economics*, pp. 1056–1065, 2005.
- [22] 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," *Proceedings of the 8th Workshop on Network and System Support for Games (NetGames '09)*, pp. 1-6, Paris, France, 2009.
- [23] X. Wang, T.T. Kwon, Y. Choi, M. Chen, Y. Zhang, "Characterizing the gaming traffic of World of Warcraft: From game scenarios to network access technologies," *IEEE Network*, Vol. 26, No. 1, pp. 27–34, 2012.
- [24] W.-C. Feng, D. Brandt, and D. Saha, "A long-term study of popular MMORPG," *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames '07)*, pp. 19–24, Melbourne, Australia, 2007.
- [25] J. Kawale, A. Pal, and J. Srivastava, "Churn prediction in MMORPGs: A social influence based approach," *Proceedings of the International Conference on Computational Science and Engineering*, pp. 423–428, Vancouver, Canada, 2009.
- [26] D. Pittman, and C. G. Dickey, "A Measurement Study of Virtual Populations in Massively Multiplayer Online Games," *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games (NetGames '07)*, pp. 25–30, Melbourne, Australia, 2007.
- [27] D. Pittman, and C. G. Dickey, "Characterizing Virtual Populations in Massively Multiplayer Online Role-Playing Games," *Lecture Notes in Computer Science Advances in Multimedia Modeling*, Vol. 5916, pp. 87–97, 2010.
- [28] Y.-T Lee, K-T Chen, Y.-M Cheng and C-L Lei, "World of Warcraft avatar history dataset," *Proceedings of the second annual ACM conference on Multimedia systems*, pp. 123–128, San Jose, CA, USA, 2011.
- [29] P. A. Branch, A. L. Cricenti, and G. J. Armitage, "An ARMA (1, 1) Prediction Model of First Person Shooter Game Traffic," *Proceedings of the IEEE 10th Workshop on Multimedia Signal Processing*, pp. 736-741, Cairns, Australia, 2008.
- [30] T. Lang, P. A. Branch, and G. J. Armitage, "A Synthetic Traffic Model for Quake3," *Proceedings of the Advances in Computer Entertainment Technology*, pp. 233-238, Singapore, 2004.
- [31] X. Zhuang, A. Bharambe, J. Pang, and S. Seshan, "Player Dynamics in Massively Multiplayer Online Games," School of Computer Science Carnegie Mellon University, *CMU-CS-07-158*, Pittsburgh, PA, USA, 2007.
- [32] T. Issariyakul, and E. Hossain, "Introduction to Network Simulator NS2," *Springer*, 2011.
- [33] H. Sawashima, and Y. H. H. Sunahara, "Characteristics of UDP packet loss: Effect of TCP traffic," *Proceeding of the Seventh Annual Conference of the Internet Society (INET '97)*, p.6, Kuala Lumpur, Malaysia, 1997.
- [34] G. Marfia, C. E. Palazzi, G. Pau, M. Gerla, and M. Roccetti, "TCP Libra: Derivation, analysis, and comparison with other RTT-fair TCPs," *Computer Networks*, Vol. 54, No. 14, pp. 2327-2344, 2010.
- [35] M. Ries, P. Svoboda, and M. Rupp, "Empirical study of subjective quality for massive multiplayer games," *Proceedings of the 15th International Conference on Systems, Signals and Image Processing (IWSSIP 2008)*, pp. 181-184, Bratislava, Slovakia, 2008.
- [36] Cooperative Association for Internet Data Analysis (CAIDA), "NASA Ames Internet Exchange Packet Length Distributions," Mar. 2008, Available: http://www.caida.org/research/traffic-analysis/AIX/plen_hist/