Traffic modeling of player action categories in a MMORPG

Mirko Suznjevic
mirko.suznjevic@fer.hr

Ivana Stupar
ivana.stupar@fer.hr

Maja Matijasevic
maja.matijasevic@fer.hr

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

ABSTRACT

In this paper we present a user action specific modeling of network traffic in a Massively Multiplayer Online Role-Playing Game (MMORPG). We have performed measurements for each of the previously defined action categories for MMORPGs (*Trading, Questing, Dungeons, Raiding, and Player versus Player Combat*) and formed models based on the obtained traces. Models are implemented through modification of Distributed Internet Traffic Generator, and verified through comparison with real traffic. As a case study we use World of Warcraft.

Categories and Subject Descriptors

K.8.0 [**Personal Computing**]: General – Games; H.4.4.3 [**Information Systems Applications**]: Communications; I.6.5 [**Simulation and Modeling**]: Model development—*traffic measurements, statistical analyses, model implementation*

General Terms

Measurement, Performance, Experimentation, Verification

Keywords

MMORPG, World of Warcraft, Traffic, Modeling

1. INTRODUCTION

Massively Multiplayer Online Role-Playing Games (MMORPGs) are becoming increasingly popular in the recent years. The amount of network traffic created by these virtual worlds is becoming more and more significant in the overall gaming traffic which is estimated to increase by the rate of 37% in the period from 2009–2014 [1], a second largest growth after video related traffic categories. For game publishers, MMORPGs present a very interesting field of computer entertainment industry as they provide very good revenues. The number of MMORPG titles in the market is growing constantly and game providers are trying to keep the existing player base, and also to attract new players. In order to achieve this, a satisfying level of Quality of Service (QoS) must be provided. Networking is one of the important aspects in the overall performance of the virtual world. In order to be realistic and immersive, a virtual world needs to be responsive, which, on the network level, means that the latency values need to be low, and crucial data must be delivered in timely manner. In order to provide the best possible networking service, traffic must be studied, analyzed, and modeled. For example, recently a network provider, due to the change introduced by the new version of a game, had issues with correctly labeling that traffic as gaming class, which resulted in a much lower QoS level followed by a revolt of the customers [25].

In this paper we model the traffic of the currently most popular subscription based MMORPG – *World of Warcraft* (WoW) [15]. In the previous works [26, 34], which modeled the traffic of WoW, authors agree that it is hard to achieve a good fit of the model. Also, the traffic of MMORPGs can be very variable and can change based on the parameters of the situation in the virtual world [7, 18, 23]. Our approach is to model the traffic across a set of action categories which represent the situation in the virtual world. Through this methodology we aim to model the traffic more precisely and to describe the relationship between the application state and the network characteristics. In our previous works we have defined action categories and performed analysis of the traffic characteristics of each category [31, 33]. Data, on which the model has been based, has been gathered during 2008 in the 2.x version of WoW (i.e., expansion *The Burning Crusade*). The process of modeling was done based on the algorithm presented in [4]. We have implemented the models in Distributed Internet Traffic Generator (D-ITG)[2], and performed validation of the generated traffic.

We have also performed measurements of player behavior in terms of defined categories [30], and investigated relationship of player motivations and their in-game behavior [32]. Based on the measurements of player behavior in future work we aim to develop a model of player behavior based on action categories which combined with traffic models will allow us to simulate the overall game traffic better.

The remainder of this paper is structured as follows: in section 2 we present the related work, in section 3 we briefly describe WoW and our action categories, in section 4 we describe the process of measuring and preparing the data, in section 5 we describe the modeling methodology, in section 6 we present the results, in section 7 we explain the implementation, and we conclude the paper in section 8.

DISIO 2011, March 21, Barcelona, Spain

DOI 10.4108/icst.simutools.2011.245546

2. RELATED WORK

With the increase of popularity of online games, the interest of the academic community for the topics related to network traffic generated by games increased as well. One of the first works in the field of traffic modeling of network games was done by Borella [4] who analyzed the traffic from the First Person Shooter (FPS) game *Quake*, using methodology introduced in [27]. There are many more works which model the traffic of different games from the FPS genre: *Quake 3* [23], *Quake 4* [9], *Halo* [22], *Halo 2* [36], *Half-Life* [16], *Counter–Strike* [10, 14].

As MMORPGs became more and more significant in the online game market, especially by the contribution of their network traffic, researchers started to shift from FPS games and Real-Time Strategies (RTS) such as *Starcraft* [11], to studying MMORPGs. Chen, Huang, and Lei [7] have performed a detailed analysis of *ShenZhou Online*, on a network trace captured on the server side with the assistance of the game operators. They have determined the following properties of the MMORPG traffic: tiny packets, periodicity, significant signaling overhead, temporal dependence of packet arrivals within connections and aggregate traffic. Explanations of those characteristics have been offered through specific user behavior (e.g., practice of team play, user behavior diversity, etc.). They state that the problem of modeling user behavior and source traffic in MMORPGs is especially challenging due to the diversity of user behaviors. Furthermore, the authors have used their trace to analyze the relationship between network QoS and session times [5], to further examine implications of player interactions on generated network traffic [8], and to investigate performance of TCP for MMORPGs [6].

Svoboda, Karner, and Rupp [34] analyzed the traffic trace of WoW captured within the mobile core network. They found that WoW is amongst the top 10 TCP based services in the monitored network and consumed 1% of all TCP based traffic. Authors performed active measurements and captured a trace of two groups consisting of five WoW clients connected to ADSL lines. Their analysis at packet level shows that the packet size in the downlink direction is quite smooth and can be modeled with a Weibull distribution while the uplink packet sizes have large discrete steps. They note that the high number of transmitted packets are ACK packets carrying no payload. Packet Inter-Arrival Time (IAT) was modeled by a joint distribution of three variables.

Kim et al. [18], [19] analyzed a traffic trace gathered at the server side link of the MMORPG *Lineage II*. They captured and analyzed over 7.6 billion packets and analyzed the aggregate traffic and per session traffic. Authors note a significant asymmetry of the upstream and downstream traffic, high percentage of the packets in the upstream consisting of only signaling information (77.1%) while the server packets having only 2.4% of pure signaling packets. Size of the packet payload in the upstream direction shows that packets are relatively small, half of the packets have less than 20 bytes of payload and 99% less than 50 bytes. Daily and weekly patterns both in the number of users, as well in the bandwidth load on the server exist with a linear correlation between the number of users and bandwidth.

Griwodz and Halorsen [17] analyze an 1 hour long, server–side, network trace of an MMORPG *Anarchy Online* provided to them by the game provider *Funcom*. Their results show that while single TCP streams are thin, the server link can be carrying hundreds or thousands of concurrent streams which together may cause congestion without reducing the sending rate. Also, using TCP does not have to be slower than using UDP as the send buffer is usually empty and an event may be sent immediately.

Wi, Huang, and Zhang model the traffic of a MMORPG *World of Legend* based on trace obtained while accessing the game through mobile GPRS access network. For packet IATs they have used an Extreme Value distribution for the client side traffic and a sum of two Extreme Value distributions for the server side traffic. Packet size was modeled as sum of discrete steps (on 66 bytes and 72 bytes) and an Extreme Value distribution for the server side and as a deterministic distribution for the client side (77 bytes).

While many works in the area of traffic analysis and modeling acknowledge the influence of different situations in the virtual world on the traffic patterns [23, 18, 7], the following works explore this relationship further. Park, Kim and Kim [26] collect and analyze network traffic traces of FPS *Quake 3* (Q3) and a MMORPG WoW. They define user actions based on the number of players and player behavior. For Q3 actions are defined as: Shooting, Moving, Normal, No Play. For WoW actions are defined as: Hunting the NPCs (Non-Player Characters), Battle with players, Moving, and No play. Authors perform modeling of every type of behavior; for WoW size of the packets is modeled by the Exponential distribution on the server side and Normal on the client side, while packet IATs are modeled with the normal distribution.

Further going into the micro scale, and analyzing contents on the packet level is done by Szabó, Veres, and Molnár [35]. Authors claim that the nature of human behavior has a high impact on traffic characteristics and that it influences the traffic both at macroscopic level (e.g., traffic rate) and at microscopic (payload content) level. They measure and analyze the traffic of WoW and *Silk Road Online*. They define the states of the virtual world by using two axes, the movement of the player (moving, stalling), and the number of surrounding players as a mean to determine the location (in or outside of the a densely populated area – "the city"). This results in four possible states: Moving in the city, Moving outside the city, Stalling in the city, and Stalling outside the city. Identification of the separate states was done through active measurements and wavelet analysis. Validation of the model was done through controlled measurements and comparison with the defined states.

Traffic analyses of the Linden Labs 3D virtual world *Second Life* also confirmed the influence of avatar movement rate and popularity of specific parts of the virtual world on the network traffic [24, 20].

Our approach is to model the traffic based on user actions through categories which are sufficiently general, distinct, and measurable in terms of player behavior in MMORPGs. While these models are created for WoW, we think that it is very likely that they capture general trends of the MMORPG traffic.Further study would be needed to prove it in a scientific way. Namely, most MMORPGs have common characteristics which shape their network traffic: a client–server architecture, some area of interest technique, and transporting only updates about dynamic entities' positions and actions over the network.

3. WOW AND ACTION CATEGORIES

WoW uses a client–server architecture, in which the virtual world is replicated on multiple "shards". A shard is a copy of a part or the whole virtual world, which resides on a specific server. Players assigned to one shard can interact with the world and other players in that shard, but they are not able to interact with players or objects in other shards. According to data from [15], WoW is currently the most popular subscription based MMORPG with 12 million active players worldwide [13]. In a MMORPG, the player typically controls a virtual character (avatar) which represents him/her in the virtual world. Players may perform a variety of actions, which typically differ depending on the game content. Nevertheless, several key fundamental elements, which are common for the most MMORPGs, may be identified: progression or advancement in player's level, social interaction, in-game culture, and character customization. Focusing on player progression and based on several key characteristics (e.g., number of actively participating players, dynamics of player input, number of active Non-Player Characters (NPCs) etc.), we defined specific action categories for MMORPGs [31, 33]. These action categories have been defined on a case study of WoW, but are considered to be applicable to MMORGPS in general:

Trading: Exchange and creation of virtual goods, between two players directly or through auction system.

Questing: Performing different tasks given by NPCs for specific rewards (e.g., experience and virtual goods). Mostly single player activity, but there are also quests which require a group effort.

Dungeons: Combat between a small player group and hostile NPCs, in specific instances (i.e., isolated portions of the virtual world which are replicated for each group of players). Instances do not allow interruption or help from players outside the specific group. This is a primary activity for small groups.

Raiding: Fighting among large group of players and more difficult and complex NPCs. This category is similar to *Dungeons*, but it is larger on all scales. As the complexity of the task increases, so does the value of the prizes, therefore *Raiding* yields the best rewards.

PvP combat: Combat between players with very low number or in complete absence of NPCs. Player count may vary significantly.

For more details on action categories specification and properties an interested reader is referred to [31, 33].

4. MEASUREMENTS

For the purpose of creating the traffic models for each action category we had to obtain the action specific network traces. Required network traces were gathered with the help of six real WoW players (volunteers). Players included in the measurement process gathered the data which was used for modeling and validation.

4.1 Packet capture

We have performed packet capturing on the clients side by using the Wireshark software network protocol analyzer (http://www.wireshark.org/). The following measurement procedure was used: as the player was about to do a specific action in the virtual world, he or she would start data capture with Wireshark. As *Dungeons*, *Raiding*, and *PvP combat* categories are performed in the specific areas of the virtual world called instances, the testing player would start the capture upon entering the instance (e.g., *Gruul's Lair* is a raiding instance, *Hellfire Citadel* is a dungeon, and *Arathi Basin* is a PvP battleground) and stop the capture when leaving the instance. For the *Questing* category, the players started the capture upon receiving the quest and stopped it once they finished all actions related to the quest (i.e., exiting the game, or setting out to do some other action). For *Trading* category, the players were instructed to capture the session in which they tried to sell or buy something in the auction house or from another player. Also, other trading actions were included, such as checking in-game mail and/or bank for retrieving items, and crafting virtual items. In the end, the players saved the trace into a file, and annotated it with a very detailed designation of what they did during that particular capture. This detailed designation was needed to solve the problem associated with the players immersion in the game. Namely, the players often forgot to stop (as well as to start) the capture at the right moment, resulting in more than one type of player actions being represented in a given trace. Through the filtering process, a large portion of taken measurement data was discarded and not taken into consideration. We captured 83 context specific network traces comprising of 1395940 packets. Interested parties may contact the first author to obtain this data (anonymized), free of charge for use in research and education, under certain agreed conditions.

4.2 Packet filtering

We note that there is a large TCP signaling overhead in our traffic trace, as the TCP ACK packets carrying no payload are quite common. We performed the filtering of the trace in which we excluded TCP ACK packets with empty payload, but we have noted the percentage of those for each category. The initial packet trace consisted of a high number of server side packets having the size corresponding to the Maximum Transmission Unit (MTU). We assume that this is the result of application trying to send datagrams larger than MTUs. In order to determine the Application Protocol Data Unit (APDU) size we have implemented an algorithm proposed in [34]. They noted that some packets had the TCP PSH flag set which is the case when application has data that it needs to have sent across the network immediately. According to the assumption, if APDU is larger than the MTU, the TCP service splits the APDU and assigns the PSH flag to set for the last packet in the sequence. We follow this approach by processing our dataset in order to calculate correct APDU values and inter-arrival times between subsequent APDU, and modeling those values (not the packet size or packet IAT).

5. METHODOLOGY

Traffic modeling for games is done by defining analytic traffic models (i.e., mathematical description). These models are easier both to convey and to analyze compared to empirical models of traffic (e.g. tcplib [12]). In this work we follow the approach for application traffic modeling by Paxson [27], firstly used in the area of network games by Borella [4]. The algorithm is also described in detail in [21], and it is presented as follows:

1. The probability distribution of the data set is examined and an appropriate analytical distribution is cho-

Table 1: Model parameters for *Trading*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	15612	Deterministic (8 distinct values)	$6 : 5.25\%, 10 : 4.21\%, 14 : 34.05\%, 15 : 5.72\%, 18 : 3.19\%, 35 : 32.50\%, 39 : 9.14\%, 51 : 5.94\%$	0.0911	32(0.20)/0	0.24
Client IAT	15602	Weibull <500 + Weibull >500 + Deterministic (2 distinct values)	$(\gamma = 0.99, \alpha = 176.74) : 50.53\%, (\gamma = 0.66, \alpha = 1220.33, \mu = 500.95) : 28.53\%, 0 : 17.60\%, 500 : 3.34\%$	0.0817	10(0.06%)/-	0.29
Server APDU size	27082	Lognormal	$(\mu = 4.16, \alpha = 1.15) : 100\%,$	0.0888	145(0.54%)/-	0.05
Server IAT	27081	Lognormal + Deterministic (2 distinct values)	$(\mu = 5.62, \alpha = 0.95) : 82.68\%, 200 : 9.62\%, 218 : 7.7\%$	0.1063	23(0.08%)/-	0.17

sen. This is usually done through the visual examination of the Probability Density Function (PDF) or Cumulative Distribution Function (CDF) of the data. An example of analytical distribution is Weibull distribution with the following PDF:

$$f(x) = \frac{\gamma}{\alpha}(\frac{x-\mu}{\alpha})^{\gamma-1}e^{-(\frac{x-\mu}{\alpha})^{\gamma}} \qquad (1)$$

where γ is the shape parameter, α is the scale parameter, and μ is the location parameter.

A very valuable tool in this process is a Quantile-Quantile plot (Q-Q plot), which is a graphical method for comparing two distributions by plotting their quantiles against each other. First, the set of intervals for the quantiles are chosen. A point (x, y) on the plot corresponds to one of the quantiles of the second distribution (y-coordinate) plotted against the same quantile of the first distribution (x-coordinate). In this way, by plotting an empirical distribution, $F(x)$, against the chosen distribution $G(x)$ we can observe the goodness of fit. If the resulting points are in a straight line, it means that the $F(x) = G(x)$, but in practice there are often deviations in the fit. Through Q-Q plot it is easy to observe where deviations occur (e.g., lower tail, the main body, upper tail).

In order to manage large data sets values are aggregated into “bins”. The final results may become skewed depending on the bin choice. The algorithm for choosing the optimal bin size is taken from [29]:

$$w = 3.49\sigma n^{(-1/3)} \qquad (2)$$

where σ is the estimate of the standard deviation and n is the number of observations.

2. The data set is fitted onto an analytical distribution using method of least squares to determine the parameters of the distribution.

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

4. Calculating the λ^2 discrepancy measure. As the standard goodness of fit tests are biased for large and messy datasets, discrepancy measure is used [28]. In short we will explain the discrepancy measure. If we have observed n instances of a random variable Y which we want to model using another distribution Z, N is the number of bins in which we partition the distribution Z. Each bin has a probability p_i associated with it, which is the proportion of the distribution Z falling into the ith bin. Let Y_i be the number of observations of Y that actually fell into the ith bin. Chi-Square goodness of test X^2 is defined as:

$$X^2 = \sum_{i=0}^{N} \frac{(Y_i - np_i)^2}{np_i} \qquad (3)$$

K parameter is defined as:

$$K = \sum_{i=0}^{N} \frac{(Y_i - np_i)}{np_i} \qquad (4)$$

Estimator for discrepancy for the grouped data is defined as:

$$\hat{\lambda}^2 = \frac{X^2 - K - df}{n-1} \qquad (5)$$

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

$$\hat{X}^2 = \sum_{i=0}^{N} \frac{(np_i - Y_i)^2}{Y_i} \qquad (6)$$

$$\hat{K} = \sum_{i=0}^{N} \frac{(np_i - Y_i)}{Y_i} \qquad (7)$$

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

$$\xi = log_2 \frac{a}{b} \qquad (8)$$

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

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

6. RESULTS

In this section we present our fits for the APDU size, and IAT across all action categories. Additional detail about traffic characteristics of each category can be found in [31]. Client packet sizes are comprised of several discrete steps. The most frequent values of payload sizes vary constantly across different actions, but packets of size 35 B are the most frequent which is in contrast with the [34] who modeled WoW client traffic with packets of size 6 B, 19 B and 43 B. We are assuming that these packets are responsible for carrying information about character's movement as suggested in [35]. Client side IATs can be divided in two sectors, below 500ms and above. For both areas Weibull distribution showed as a good fit. Also, significant "spikes" exist on values 0 and 500, which we assume to be due to dynamics of player activity. Subsequent packets (0 ms IAT) are sent while player is performing a highly dynamic action (e.g., highest percentage of 0ms IATs is in *PvP combat*), while 500 ms IAT is probably due to some sort of keep alive mechanism. Server side APDU size have a good fit in Weibull distribution with some discrete steps (usually at 37 B). Also, we have noted that spikes occur around 7000 B, probably related to loading instances where significant data must be transported as these spikes occur only in action categories which are related to instances (i.e., *Dungeons, Raiding, PvP combat*). Discrete steps for server side IATs have been observed at 44 ms, 200 ms, 218 ms and 328 ms. While the step at 200 ms can be explained with the TCP delayed ACK mechanism, the rest of the steps are probably inherent to the WoW application protocol. Server IATs have, in general, been the most complex to model. Parameters of the models for action categories are given in respective Tables 1-5.

Trading shows a very high number of pure signaling packets from the client side, the smallest mean of server side APDU sizes (164 B) and the lowest packet rate.

Questing as the most "general" category comprises various types of actions. This category is dominant in the behavior of the players during the process of obtaining the maximum character level. At the later stages of the game, in the game's "stationary" state, questing is not so prevailing as the players turn to more group based activities. More information about the player behavior patterns can be found in [31, 30].

PvP combat category is the most dynamic one in terms of user input, which results in the highest packet rate for the client traffic (10pps). Also, this is a category with the largest number of distinct APDU sizes in the client traffic. The highest percentage of the packets with the value of 35 B, the ones carrying information on player movement, is in *PvP combat*, which requires constant moving from the players.

Dungeons is the "average" category, as it's traffic characteristics are in the middle between the single player activities

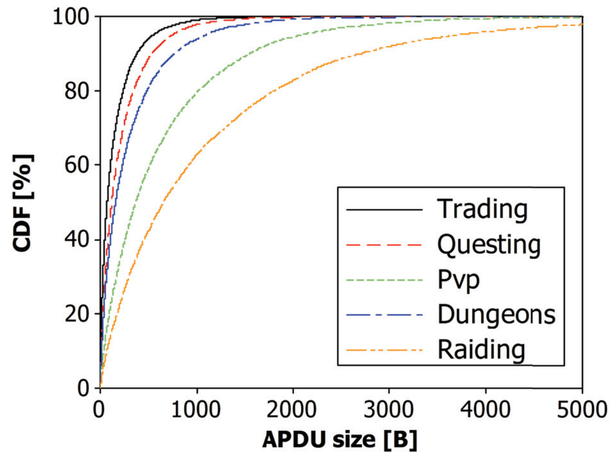

Figure 1: Weibull distributions of the server APDU sizes

on one side and large group activities on the other side.

Raiding creates the largest amount of information from the server side. As we can see from the Figure 1, *Raiding* traffic from the server side carries the largest payloads. This is understandable, as this category is usually involving many players and NPCs, so all the updates for each of the entities have to be sent. Server side APDU size is directly connected to the number of active entities in the area of interest of particular avatar. As the size of the payload is high, so is the rate of packets (9.5 pps). This causes a large number of "pure" TCP ACK packets to be sent from the client side.

7. IMPLEMENTATION AND VALIDATION

To create the testbed for our model validation, we implemented our models of application protocols through modification of Distributed Internet Traffic Generator (D-ITG) [2]. D-ITG is an open-source traffic generator which allows defining distributions of packet sizes and IATs of generated traffic. It also offers a choice of several transport and application layer protocols. Detailed architecture overview of the Distributed Internet Traffic Generator can be found in [3].

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

In order to validate that the generated traffic has the required characteristics, we have performed comparison between the parameters of the analytical model, generated traffic, and the traffic from the validation traces. In this way we can see how good is the model fit vs validation traces, and how closely generated traffic follows the parameters set by the model. Due to the space constraints of the paper we

Table 2: Model parameters for *Questing*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	63541	Deterministic (8 distinct values)	6:4.96%, 10:7.34%, 14:20.75%, 18:2.82%, 21:2.36%, 35:50.18%, 39:9.20%, 51:2.39%	0.0415	11(0.017%)/0	0.0.46
Client IAT	63531	Weibull <520 + Weibull >520 + Deterministic (2 distinct values)	$(\gamma = 1.19, \alpha = 236.22)$: 55.7%, $(\gamma = 0.84, \alpha = 1073.63, \mu = 525.95)$: 12.6%, 0 : 16.46%, 500 : 15.24%	0.1608	101/(0.17%)/-	0
Server APDU size	99163	Lognormal	$(\alpha = 1.22, \mu = 4.55)$: 100%	0.2304	163(0.16%)/-	0.05
Server IAT	99177	Normal<420 + Weibull>420 + Deterministic (3 distinct values)	$(\mu = 212.87, \sigma = 96.59)$: 71.51%, $(\gamma = 0.91, \alpha = 451.55, \mu = 419.96)$: 7.49%, 44 : 2.15%, 218 : 12.27%, 328 : 6.58%	0.1364	47(0.48%)/-	0.21

Table 3: Model parameters for *PvP combat*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	66635	Deterministic (7 distinct values)	6:7.63%, 10:5.60%, 14:13.12%, 19:3.11%, 35:59.50%, 51:6.66%, 58:4.38%	0.1307	0/0	0.39
Client IAT	66631	Weibull + Deterministic (2 distinct values)	$(\gamma = 0.79, \alpha = 208.50)$: 78.4%, 0 : 20.18%, 500 : 1.42%,	0.0681	155/(0.23%)/-	0.24
Server APDU size	71594	Weibull<7200 + Largest Extreme Value>7200 + Deterministic(1 distinct value)	$(\gamma = 0.92, \alpha = 538.59)$: 93.08%, $(\mu = 7754.99, \alpha = 394.83)$: 0.73%, 37 : 6.19%	0.0183	14(0.02%)/-	0
Server IAT	71594	Weibull + Deterministic (3 distinct values)	$(\gamma = 1.71, \alpha = 193.26)$: 83.32%, 44 : 4.13%, 200 : 4.11, 328 : 8.44%	0.1799	97(0.14%)/-	0.22

Table 4: Model parameters for *Dungeons*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	50460	Deterministic (7 distinct values)	6:4.57%, 10:8.00%, 14:16.28%, 19:4.04%, 22:8.28%, 35:55.70%, 51:3.13%	0.1048	55(0.04%)/0	0.27
Client IAT	50460	Weibull + Deterministic (1 distinct value)	$(\gamma = 0.58, \alpha = 268.37)$: 95.86%, 500 : 4.14%	0.2038	221(0.44%)/-	0.33
Server APDU size	96035	Weibull<7450 + Largest Extreme Value>7450	$(\gamma = 0.89, \alpha = 221.83)$: 99.15%, $(\mu = 7698.83, \alpha = 198.842)$: 0.85%	0.0331	63(0.065%)/-	0.01
Server IAT	96056	Weibull<405 + Weibull>405 + Deterministic (3 distinct values)	$(\gamma = 2.28, \alpha = 231.3)$: 78.35%, $\gamma = 0.79, \alpha = 344.14, \mu = 405.96)$: 2.58%, 44 : 3.06%, 200 : 9.55%, 328 : 6.46%	0.1443	32(0.03%)/-	0.16

Table 5: Model parameters for *Raiding*

Data	Count	Model	Parameters	$\hat{\lambda}^2$	Tail	ACF(1)
Client APDU size	19136	Deterministic (8 distinct values)	6:3.81% 10:4.35%, 14:12.15%, 19:20.18%, 20:3.63%, 29:6.81%, 35:45.53%, 51:3.54%	0.1022	19(0.1%)/0	0.27
Client IAT	19135	Weibull + Deterministic (1 distinct value)	$(\gamma = 0.76, \alpha = 299.52)$: 85.73%, 0 : 14.27%	0.0898	65(0.34%)/-	0.27
Server APDU size	37801	Weibull<7200 + Weibull>7200	$(\gamma = 0.86, \alpha = 941.79)$: 98.97%, $(\gamma = 0.91, \alpha = 1183.28, \mu = 7298.20)$: 1.03% :	0.0342	16(0.04%)/+	0.16
Server IAT	37801	Weibull + Deterministic (2 distinct values)	$(\gamma = 1.99, \alpha = 188.92)$: 84.39%, 44 : 9.55%, 200 : 6.06%	0.0660	6(0.02%)/-	0.03

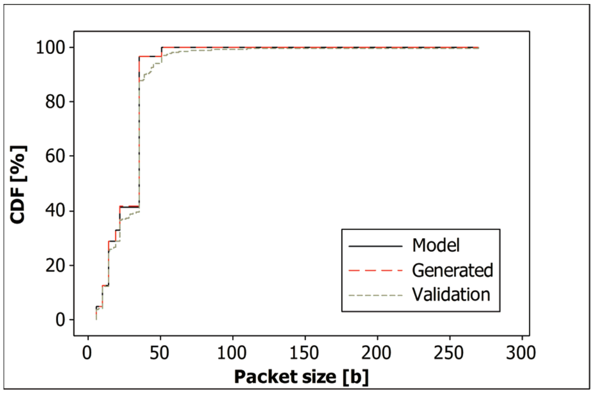

Figure 2: CDF of Dungeons client side PS

do not present figures for all parameters through all categories, but select the figures for one of the each parameters for four different action categories as representatives. Figure 2 shows the client generated traffic of *Dungeons* which is closely following the model, and rising slightly faster than the validation data, due to the discrete steps. The fit of the client IAT of the *Trading* category, shown on the Figure 3, is very reasonable, and the generated traffic follows the parameters of the model closely.

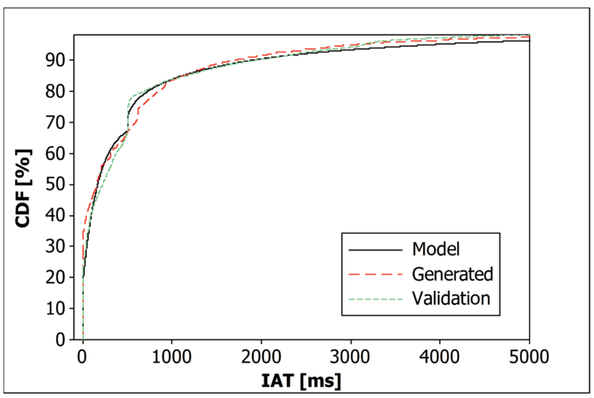

Figure 3: CDF of Trading client IAT

On the other hand the characteristics of the server payload for *PvP combat* show several discrepancies. We can see that the validation traces are limited on 1460 due to the MTU, while the generated traffic does not show such properties. By inspecting the generated traces we found out some issues with fragmentation which we assume stem from D-ITG, but determining the exact reason is still a work in progress.

Server IAT of the *Raiding* category is shown in the Figure 5. While the model underestimates the lower values, the distortion of the generated traffic is significant over the 200 ms mark. Clearly, this issue also requires further study.

8. CONCLUSION

We proposed several action-category based models of network traffic and their implementation based on D-ITG traffic generator. The comparison of between the model and the actual generated traffic shows a fairly good fit for the client traffic for action categories *Dungeons* and *Trading*, while the results for the server side traffic for the action categories *PvP combat* and *Raiding* need improvement.

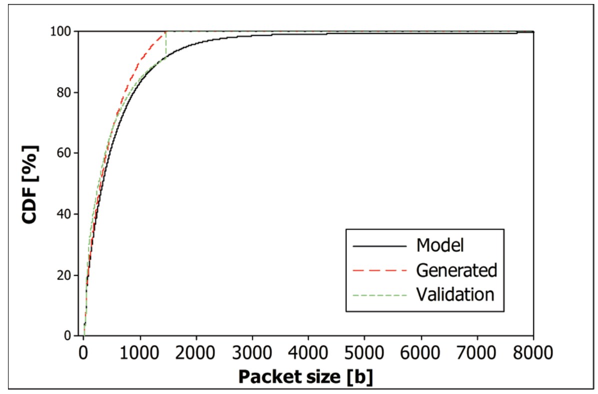

Figure 4: CDF of PvP combat server PS

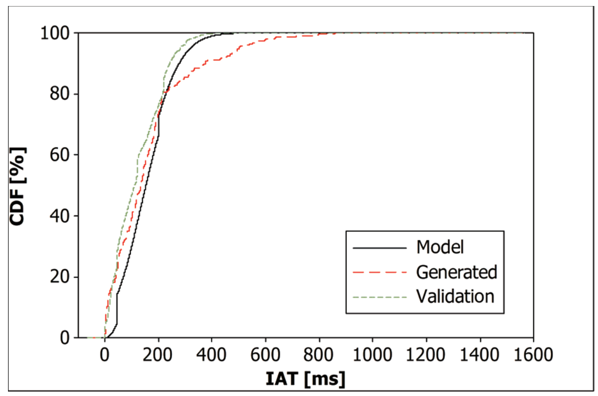

Figure 5: CDF of Raiding server IAT

9. ACKNOWLEDGMENTS

This work was supported by MZOS research project 036-0362027-1639, and Ericsson Nikola Tesla.

10. REFERENCES

[1] Cisco Visual Networking Index: Forecast and Methodology, 2009–2014, 2010. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[2] Distributed Internet Traffic Generator, Nov. 2010. http://www.grid.unina.it/software/ITG/

[3] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Venturi. D-ITG Distributed Internet Traffic Generator. pages 316–317, 2004.

[4] M. S. Borella. Source models of network game traffic. *Computer Communications*, 23(4):403–410, 2000.

[5] K. Chen, P. Huang, G. Wang, C. Huang, and C. Lei. On the sensitivity of online game playing time to network QoS. In *Proceedings of the IEEE INFOCOM'06*, pages 1–12, 2006.

[6] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. An empirical evaluation of tcp performance in online games. In *ACE '06: Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, page 5, 2006.

[7] K.-T. Chen, P. Huang, and C.-L. Lei. Game traffic analysis: An MMORPG perspective. *Computer Networks*, 51(3):19–24, 2007.

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

[9] A. Cricenti and P. Branch. ARMA(1,1) modeling of Quake4 server to client game traffic. In *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, pages 70–74, 2007.

[10] A. Dainotti, A. Botta, A. Pescapé, and G. Ventre. Searching for invariants in network games traffic. In *Proceedings of the 2006 ACM CoNEXT conference*, pages 43:1–43:2, 2006.

[11] A. Dainotti, A. Pescape, and G. Ventre. A packet-level traffic model of Starcraft. In *Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems*, pages 33–42, 2005.

[12] P. Danzig and S. Jamin. tcplib: A library of TCP internetwork traffic characteristics. Technical Report CS-SYS-91-01, Computer Science Department, University of Southern California, 1991.

[13] Blizzard Entertainment. World of Warcraft subscriber base reaches 12 million worldwide, Nov. 2010. http://us.blizzard.com/en-us/company/press/pressreleases.html?101007

[14] J. Färber. Network game traffic modelling. In *Proceedings of the 1st workshop on Network and system support for games*, pages 53–57, 2002.

[15] I. V. Geel. Mmodata charts v3.1, Nov. 2010. http://mmodata.blogspot.com/

[16] T. L. Grenville, T. Lang, G. Armitage, P. Branch, and H. yi Choo. A synthetic traffic model for half-life. In *in Australian Telecommunications, Networks and Applications Conference (ATNAC*, 2003.

[17] C. Griwodz and P. Halorsen. The fun of using TCP for a MMORPG. In *Proceedings of the 2006 international workshop on Network and operating systems support for digital audio and video*, pages 1:1–1:7, 2006.

[18] J. Kim, J. Choi, D. Chang, T. Kwon, Y. Choi, and E. Yuk. Traffic characteristics of a massively multi-player online role playing game. In *Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–8, October 2005.

[19] J. Kim, E. Hong, and J. Choi. Measurement and analysis of a massively multiplayer online role playing game traffic. In *Proceedings of Advanced Network Conference*, pages 1–8, 2003.

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

[21] J. Lakkakorpi, A. Heiner, and J. Ruutuc. Measurement and characterization of Internet gaming traffic, Feb. 2002. Research Seminar on Networking.

[22] T. Lang and G. Armitage. An Ns2 model for the Xbox System Link game Halo. Technical Report 030613A, Swinburne University of Technology, Faculty of Information and Communication Technologies, Center for Advanced Internet Architectures, 2003.

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

[24] H. Liang, R. N. Silva, W. T. Ooi, and M. Motani. Avatar mobility in user-created networked virtual worlds: measurements, analysis, and implications. *Multimedia Tools Applications*, 45(1-3):163–190, 2009.

[25] Virgin Media. World of warcraft latency issues, 2010. http://community.virginmedia.com/t5/Broadband-down-your-phone-line/World-of-Warcraft-Latency-Issues/m-p/161857

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

[27] V. Paxson. Empirically-derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(2):316–336, 1994.

[28] S. P. Pederson and M. E. Johnson. Estimating model discrepancy. *Technometrics*, 32(3):305–314, 1990.

[29] D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

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

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

[32] M. Suznjevic and M. Matijasevic. Why MMORPG players do what they do: relating motivations to action categories. *International Journal of Advanced Media and Communication*, 4(4):405–424, 2009.

[33] M. Suznjevic, M. Matijasevic, and O. Dobrijevic. Action specific massive multiplayer online role playing games traffic analysis: Case study of World of Warcraft. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, pages 106–107, 2008.

[34] P. Svoboda and W. K. M. Rupp. Traffic analysis and modeling for World of Warcraft. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 1612–1617, 2007.

[35] G. Szabó, A. Veres, and S. Molnár. On the impacts of human interactions in MMORPG traffic. *Multimedia Tools & Applications*, 45(1-3):133–161, 2009.

[36] S. Zander and G. Armitage. A traffic model for the Xbox game Halo 2. In *Proceedings of the international workshop on Network and operating systems support for digital audio and video*, pages 13–18, 2005.