Automatic Retrieval of Network Traffic Data for Analysis of Network-In-world Action Relations in MMOGs

Mário Ferreira, José Queirós, Ricardo Morla

Abstract— **Massive multiplayer online games (MMOGs) are increasingly popular because they provide entertainment, numerous opportunities for socialization, and the ability for users to make money. Cornerstone to MMOGs is the underlying network traffic between MMOG clients and servers; understanding this traffic is important for application developers trying to optimize game performance and for ISPs trying to provide a better quality of service for their customers. This paper goes a step forward in helping to understand MMOG network traffic as it describes an automated approach to collect traffic samples and map them to specific in-world actions in the MMOG game Second LifeTM. We show the validity of our approach by collecting 10 samples from 8 different actions, characterizing these samples, and grouping samples by action using a k-means clustering algorithm.**

Index Terms—**Massive Multiplayer Online Games, Computer networks, Modeling**

I. Introduction

THE latest wave of virtual world technology – including the Massive Multiplayer Online Games “World Of Warcraft” and “Second Life” – showed us that it is possible to have blooming social, technological, and economic worlds that have an impact on the real world. These games are radically different from the single player games of the 80’s and 90’s: the in-world content is created and deployed online by the users, rather than offline by game creators. This has a strong impact on the characteristics of the data that flows between client and server. Understanding these characteristics is important for application developers that try to improve the performance of the online games and for ISPs that try to provide a better quality of service for their clients. Previous studies have tried to understand the relation between in-world actions and content and the network traffic that these actions generate. However, their data collection is manual, which is a slowdown for the capture of large number of samples needed for statistically significant analysis of the captured data. This paper proposes an approach for automating the capture of such data.

Section II describes the architecture and implementation of our approach. For validating our approach, section III shows how we collected and characterized 10 samples for each of 8 different in-world actions. Section IV continues the validation of our approach by showing an example of grouping the collected samples in different actions using standard k-means clustering. Section V discusses related work and section VI presents concluding remarks.

II. Arquitecture and Implementation

There are three elements involved in our approach: the SL client, the SL server, and the network traffic monitor. The following steps are involved in collecting a network log sample: 1) install, setup, and run the SL client, SL server, and traffic monitor; 2) use the client to log into the SL server; 3) have the avatar do the desired set of in-world actions; 4) shutdown client, monitor, and server; 5) label resulting network log sample with in-world action.

We used Wireshark’s (http://www.wireshark.org) command-line interface and a command-line SL client available in the libopenmetaverse package (http://www.openmetaverse.org). This enabled the monitoring and the in-world actions to be fully automated. In-world actions were done in the open source OpenSim Second Life server (http://opensimulator.org) deployed specifically for this project.

We developed a graphical control interface for the person conducting the experiments to be able to define which actions should be done in-world, how many times to repeat those actions, and where to save the sample logs; fig. 1 shows the architecture of our approach, and fig. 2 shows a screenshot of the graphical interface.

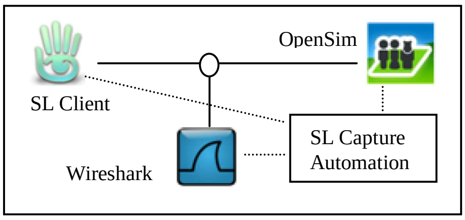

Fig. 1. Architecture of the proposed approach, including the application that automates the data collection process.

M. Ferreira, J.Queirós, and R.Morla are with the Faculty of Engineering of the University of Porto and INESC Porto, Porto, Portugal (phone: +351 222 094 200; fax: +351 222 094 250; emails:mlf|jqueirós|rmorla@inescporto.pt).

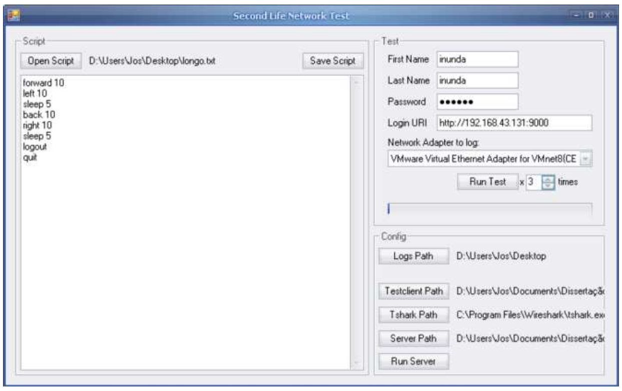

Fig. 2. Graphical interface of our data collection tool.

III. EXPERIMENT DATA CHARACTERIZATION

We used the software that we developed to collect 10 network log samples for each of 8 different actions, in a total of 80 network log samples. Table I shows the different actions that were chosen for these experiments. All components of the experiment (SL server and client, Wireshark, and the control interface) ran in the same host. With this setup we reduced interferences from external components on the network. The server was empty of any virtual objects or additional avatars.

TABLE I
8 ACTIONS CHOSEN FOR THIS EXPERIMENT

Action Index	Action Name	Action Description
1	Back	Walk backwards for 10 seconds.
2	Fly	Start flying; stop flying.
3	Fly To	Fly forward for 5 seconds.
4	Forward	Walk forward for 10 seconds.
5	In/Out	Login; logout.
6	In/Wait/Out	Login, wait 60 seconds, logout:
7	Left	Walk left for 10 seconds.
8	Right	Walk right for 10 seconds.

Fig. 3 shows the number of packets captured for each sample. Although the samples in each column 1 to 8 correspond to the same action, it is clear that the tool does not generate samples with the same number of packets. Table II shows the variation in the number of packets for each action; actions 1, 5, and 7 have relative standard deviation below 10%, whereas for samples 2 and 6 this value is above 25%. The latter correspond to actions that have fewer packets.

In order to characterize this data we computed the empirical probability distribution function (pdf, histogram divided by total number of packets) of the packet lengths for each sample. The histogram bins range from 50 to 200 bytes, and the bin size is 1 byte. Fig. 4 shows an example of such pdfs, where some variation can be seen in the data. To better characterize such variation, we computed the Euclidian norm of each histogram and the standard variation of such norms; this can be seen in fig. 5 and table III, respectively.

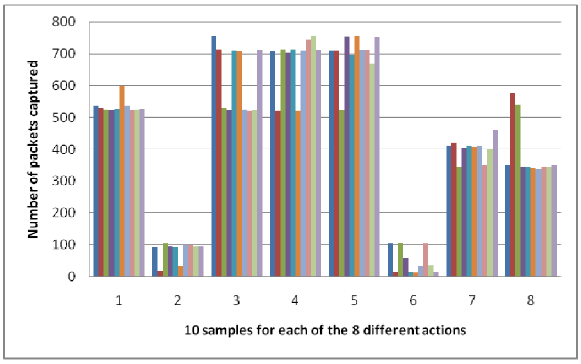

Fig. 3. Number of packets captured in each sample, grouped by action (1 to 8).

TABLE II
AVERAGE, STANDARD DEVIATION, AND RELATIVE STANDARD DEVIATION OF THE NUMBER OF PACKETS PER SAMPLE IN EACH ACTION

Action Index	Average	Std. Dev.	Std. Dev. / Average
1	668.25	23.42	3.51%
2	103.88	30.26	29.13%
3	776.75	104.28	13.43%
4	894.50	85.80	10.10%
5	873,50	67.91	7.77%
6	63,63	39.91	62.72%
7	503,50	33.29	6.61%
8	484,75	90.14	18.60%

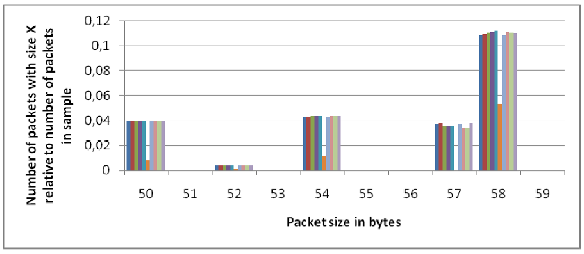

Fig. 4. Example empirical probability distribution function of the 10 samples of action 1, limited to 10 bins (50-59 bytes) for easier visualization.

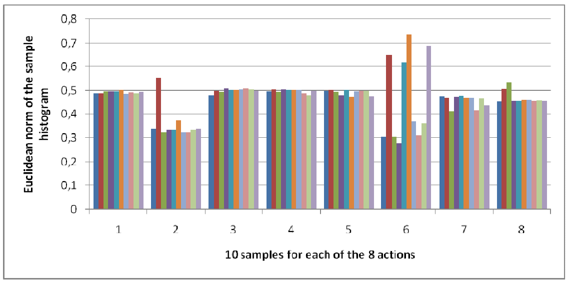

Fig. 5. Euclidean norm of the pdf of each sample, grouped by action.

TABLE III
AVERAGE AND STANDARD DEVIATION OF THE EUCLIDEAN NORM OF THE PDF OF EACH ACTION'S SAMPLES

Action Index	Average	Std. Dev.
1	49.29%	0.48%
2	35.89%	7.03%
3	50.12%	0.77%
4	49.83%	0.78%
5	49.29%	1.12%
6	46.26%	18.55%
7	45.76%	2.47%
8	46.99%	2.78%

This characterization based on the Euclidean norm of the empirical pdf points to the same conclusion as the characterization based on total number of packets captured: samples captured for actions 2 and 6 show higher variations and, consequently, may prove more difficult to use in the analysis of actions based on network data.

IV. IN-WORLD ACTION CLUSTERING

After characterizing the data, we tried to cluster actions in the virtual world based on the empirical probability distributions of the samples that we collected using our tool.

We used the k-means standard algorithm from MATLAB and all 201 bins from the empirical probability distribution as parameters to the clustering algorithm. We grouped samples in 8 classes, one class for each action. Table IV presents the results of this clustering.

TABLE IV
CLUSTERING RESULTS

Action Index	Sample Index 1	2	3	4	5	6	7	8	9	10	Dominant Class	Success Rate
1	1	1	1	1	1	3	1	1	1	1	1	90%
2	7	7	7	7	7	7	7	7	7	7	7	100%
3	4	4	5	5	4	4	5	5	5	4	4 and 5	50%
4	3	1	3	3	3	1	3	3	3	3	3	80%
5	4	4	5	4	4	4	4	4	4	4	4	90%
6	7	7	7	7	7	7	7	7	7	7	7	100%
7	2	2	2	2	2	2	2	2	2	2	2	100%
8	6	8	8	6	7	6	6	6	6	6	6	70%

Each action has a dominant class, which is the clustering class to which more samples of the action were assigned to. There's a match for action 3 with 50% on each class.

Clearly the clustering algorithm can group together network samples from action 1 and 7 with 90% success and above. However, it cannot distinguish from samples of actions 2 and 6, and samples of action 3 seem to be closely related to those of action 5.

Although actions 2 and 6 are very different (fly vs. in, wait, and out), we notice that in the previous section, actions 2 and 6 have been shown to have higher variance in the number of packets per sample and in the Euclidean norm of the pdf of their samples. We also notice that the total number of packets for samples of actions 2 and 6 is significantly lower than those of the other actions. All of this may explain their grouping in the same class by the clustering algorithm.

V. RELATED WORK

The characterization of the network behavior of computer games started with first person shooters in local area networks and focused mostly on the delay introduced by the local network [1]. With the advent of online games such as Second Life, [2] tried to understand the relation between bit-rate, packet size, packet inter-arrival time, and amount of traffic with popular vs. unpopular SL sites, different hours of the day, and 3 different avatar actions – standing, walking, and flying. [3] defines additional features for sites including an approximation on the number of objects and avatars in the site, and the additional action of teleporting. [4] expands the work of [2] to model SL client traffic with a number of known distributions including beta, gamma, and lognormal.

The key issue with existing work that tries to characterize SL traffic is that the process of collecting data is manual; there is no automated approach to collect multiple samples of the same action, which is crucial for statistically significant results. Our approach provides such a mechanism.

VI. CONCLUSIONS

This paper presented and automatic approach to collect network log samples triggered by specific actions in a virtual 3D world. Using this approach, we collected samples from different actions, characterized these samples regarding number of captured packets and the Euclidean norm of their empirical probability distributions, and finally analyzed the relation between samples and in-world actions using k-means clustering. These examples show that it is possible to use our approach to automatically collect network log samples and analyze the relation between in-world actions in Second Life and these samples.

REFERENCES

[1] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu. "The effect of latency on user performance in Warcraft III". 2nd Workshop on Network and System Support for Games, pp 3-14, 2003.

[2] S. Fernandes, C. Kamienski, D. Sadok, J. Moreira, and R. Antonello. "Traffic Analysis Beyond This World: the Case of Second Life". NOSSDAV, Illinois, USA, June 2007

[3] J. Kinicki, M. Claypool. "Traffic analysis of avatars in Second Life". International Workshop on Network and Operating System Support for Digital Audio and Video, 2008, pages 69-74.

[4] R. Antonello, S. Fernandes, J. Moreira, P. Cunha, C. Kamienski, and D. Sadok. "Traffic analysis and synthetic models of Second Life". Springer Multimedia Systems Journal, Volume 15, Number 1 / February, 2009