

# Towards Understanding the Evolution of Wars in Virtual and Real Worlds

Géza Szabó\*, András Veres\*, Sándor Molnár†  
\*TrafficLab, Ericsson Research, Budapest, Hungary  
Email: {geza.szabo, andras.veres}@ericsson.com  
†High Speed Networks Laboratory,

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics  
E-mail: molnar@tmit.bme.hu

## Abstract

*Emerging Massively Multiplayer Online Real Time Strategy games require complex game server architecture to make the transmission of the state information of a huge number of units generated by a lot of players feasible. This architecture design is supported by network traffic simulations based on the accurate characterization of player behavior. However, these characteristics of player behavior in RTS games have not been investigated yet, thus, this is the main motivation of this paper. In particular, we introduce a method that can identify the war periods in real-time strategy game sessions based on non-intrusive measurements thus, it is possible to analyze a vast number of game plays. The paper also presents some early results comparing the real and gaming worlds.*

**Keywords.** *real-time strategy games, traffic modeling, architecture design*

## 1. Introduction

Current wide-spread Massively Multiplayer Online Games (MMOGs) are implemented mainly as heavily centralized client-server architecture. However, this architecture has several limitations like poor scalability. To overcome such drawbacks an intensive research has been initiated to focus on more decentralized or even peer-to-peer support systems. In case of MMOGs a single game server cannot handle all game events efficiently, requiring the world to be divided into several smaller parts which are served by a cluster of game servers [1]. In case of Massively Multiplayer Online Role Playing Games (MMORPGs) it is a straightforward approach as the player can only be at one place at a time. Player movement across territory boundaries is handled by the handover of the player instance between the different gaming servers.

Providing Real Time Strategy (RTS) game service to a high number of simultaneous players (MMORTS) causes the problem of server load balancing becoming even more complex. The player can manage hundreds of units spread across the virtual world. It is inefficient to transmit all the information around every single unit of the player. Authors of [2] propose to obtain up-to-date data from the server only

in those cases when the player is actually concentrating on the specific territory. The ingame events on the surrounding area of the left-alone units are simulated. The authors argued that the economy, military build-up and combat related tasks are following each other therefore, information about the duration of these tasks is vital to make efficient network traffic simulations possible. In [2] authors argued that there had not been such published work yet thus, they applied guessed parameters based on their own gaming experiences. The motivation of our work is to find efficient methods to extract this necessary information for MMORTS player and network simulation. Therefore, we believe that our results contribute to establishing a solid base for the architecture design of upcoming MMORTS games.

This paper focuses on the issue to establish the possibility to answer the above questions. We need a method to collect the necessary statistics from data sources containing bulk information. In some papers authors use game server logs ([3], [4]) as an input for further analysis. In RTS games any player can be the candidate to host the game server thus logs should be collected from plenty of end users which makes this approach less feasible. Another possible approach is to monitor the network traffic. As there is a number of different games e.g., a deep packet inspection method is not generally applicable as it would be valid for only one specific game. A statistical method would be desirable as it is general and works for even new upcoming games as well. It would make it feasible to analyze vast number of gaming sessions in non-intrusive measurements obtained at high network aggregation level. In current literature there is no such method which can collect the necessary statistics thus we propose a method which builds on heuristics to deduce the battle events from passive measurements.

We observed the internal structure of the network traffic when the forces are collected and sent into battle and also when the battle is over. This observation has been examined and validated with several methods and found that even if a heuristic method based on statistical properties of the traffic has several fallbacks it can be applied on the recognition of battles with high hit ratio.

The rest of the paper is organized as follows. Section 2 presents the related work. In Section 3 the impact of gaming

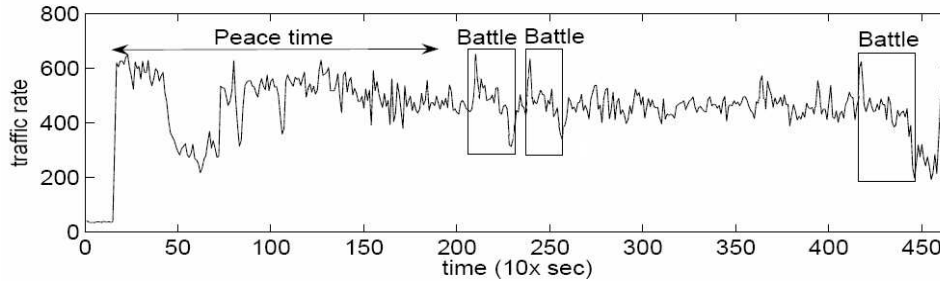


Figure 1. Cossacks measurement traffic intensity (packets/10 sec)

events on the traffic characteristics are discussed. Section 4 describes the proposed algorithm for battle detection. In Section 5 the proposed method is validated in several steps. In Section 6 we present the results of the method applied on the traffic of an operational broadband network. Finally Section 7 concludes the paper.

## 2. Related work

The predecessor in time, look and feel and server status update mechanism of recent RTS games and MMORPGs are the multiplayer First-Person-Shooter (FPS) games. There are some papers in literature which covers the topic of the impact of player behavior on the network traffic of the games. Authors of [5] discovered the dependence of server traffic bandwidth on ingame events of FPS games. In their later work [6] they managed to refine their measurements and divided UT2003 [7] into the fundamental user interaction components of movement and shooting, sub-dividing movement up into simple and complex movement, and sub-dividing shooting based on the precision of the weapons being shot.

Regarding MMORPG games, authors of [8] showed what happens when an avatar performs different actions in the virtual world, at different places and under different network conditions. The game client makes intensive use of network resources. They identified the reasons of high bandwidth consumption e.g., ingame music, numerous unique character clothes, etc. deriving from several game servers.

In [5] and [9], authors showed that RTS games traffic rate is influenced by the number of players. They argued that there are exact level shifts in the bandwidth in case of various numbers of players. The analysis of the effects of player behavior on the traffic characteristics of RTS games are currently missing. In this paper we show that there are other effects than player number which can be discovered in the network traffic of RTS games.

## 3. Impact of war on traffic characteristics

Current client-server gaming environments works by clients sending information about the actions of its own

player. The server assembles and distributes this information to make each client determine the state of the gaming environment. Each machine with identical set of information could process the data on its own and the environment would appear exactly the same for every player. In case of First Person Shooter (FPS) games or MMORPGs the number of commands a player can issue is similar during the whole game (apart from that in case of improving the capabilities of the character the player has a wider variety of choices).

In this section we will show that in case of RTS games, the number of commands the network has to transmit is in parallel with the number of units the user can manage. On the other hand, the number of commandable units is in connection with the state of the player's economy. A good example for how these factors influence the network traffic rate can be examined in the case of battles. Cossacks [10] was chosen for traffic analysis because it operates with grand armies containing several hundred units. We created a packet capture and the ingame video was recorded in parallel. Later the actions during game play and the packet rate were correlated. The battles can take several minutes long and during this time, the user manages the unit formations, attacks or withdraws and during the battle the player will lose most of the units. It was expected to see these effects in game traffic.

Figure 1 shows that on 10 sec timescales, mainly the gamer actions and the game world status (the players built-up economy, army) defines the structure of the traffic rate function. During peace time the players can not attack each other, thus they only build economy. However, the initial phase of the game has several falls in the traffic, which is the result of the lack of resources of the players during game play and can not do anything but wait for the finish of the construction of a building. During the game play progression the players establish a stable economy and the network traffic becomes smooth.

The battles result in typical changes in the gaming traffic. In the initial phase of the battle, the armies are collected and the units are marched to the front line which generates a local peak in the traffic rate. During the battle, as the units die, the traffic falls parallel. The biggest traffic rate

fall occurs in the case of the final battle when the army of one of the players is fully destroyed. After the battle the winner player occupies the loser's city with its surviving army. This implies low-rate traffic. The game soon ends and the collected statistics of the game play are sent, resulting in a higher traffic rate again. Comparing the reconstruction phase of the game when no battles but rebuilding of the economy takes place, there are no falls or peaks in game traffic.

#### 4. Detection of war

Summarizing our findings in the previous section, the battles in the traffic rate of Cossacks induce an increase in the beginning of the battle due to the collection of available forces. The end of the battle induces a fall in traffic rate due to the mass losses in commandable units. This raises the question how general the problem is. The battle in an RTS game means a local peak in the traffic rate, then a decreasing trend ending with local minima. To grab these specific characteristics we calculated a linear regression in a sliding window for the traffic rate of the server (see Alg. 1 for the Matlab [11] skeleton code of the algorithm in details). The indicators of the battle are the following properties:

- Local minima in the gradient ( $p_1$ ) – *Is there a slope?*: We searched the coefficients of a polynomial  $p(x)$  of first degree that fits the data,  $p(x(i))$  to  $y(i)$ , in the least squares sense.  $x(i), i = n..n + k$  stands for the examined time interval where  $n$  is the start of the examined time interval, and  $k$  is the length of the sliding window;  $y(i)$  is the traffic rate in the  $i^{th}$  time. The result  $\vec{p}$  of  $p(x) = p_1x + p_2$  is a row vector of two-unit-long containing the polynomial coefficients in descending powers.
- Local maxima in the mean rate of the traffic in the sliding window – *Is there an increase in the rate?*
- Local minima in the variance of the difference between the fitted polynomial and original time series – *Does the linear fit well?*:  $\min\{|y(i) - p(x)|\}, i = n..n + k$
- The standardized variance of the gradient is above a limit – *Is there a unusual jump in the rate?*:  $\tilde{\sigma}\{p_1^i x\} > c, i = n..n + k$

In Figure 2 the analysis of the traffic of Figure 1 can be seen. It can be noted that at about 2300 sec there are two local minimas of the gradient function: local minima in the variance of the difference between the fitted polynomial and original time series and local maxima of the mean packet rate. The gradient variance has local maxima at about 2100 s. The effects do not occur in the same time and the sliding window shifts them away from each other and this has to be tolerated. We checked the constellation of all the positive occurrences of the properties in the length of the sliding window range, which was chosen to 30 time-unit (30x10sec).

---

#### Algorithm 1: RTS battle recognition algorithm

---

```

Input: rate, time-window: N
Output: out
1 nx = length(rate); m = N/2;
2 xv= (1: 1: length(rate));
3 allmean=mean(rate); for i=1:nx do
4   if ((i - m) >= 1) & ((i + m) <= nx) then
5     p = polyfit(xv(i-m:i+m),rate(i-m:i+m),1);
6     f = polyval(p,xv(i-m:i+m));
7     varianceV(i)=var(rate(i-m:i+m)-f);
8     meanV(i)=mean(rate(i-m:i+m));
9     out(i) = p(1);
10 varianceAvg = slidingavg(varianceV, N) /sliding average calculation/;
11 meanAvg = slidingavg(meanV-allmean, N);
12 outAvg = slidingavg(out, N);
13 outvar = slidingvar(out+min(out), N) /sliding variance calculation/;
14 varmin = maxima(-varianceAvg) /select local minimas/;
15 meanmax = maxima(meanAvg) /select local maximas/;
16 outmin = maxima(-outAvg) /select local minimas/;
17 for i=1:length(outmin) do
18   finds1 = find(abs((outmin(i) - varmin) < N));
19   finds1size = size(finds1);
20   if finds1size(1) > 0 & outvar(i) > 0.02 then
21     finds2 = find((abs(outmin(i)-meanmax) < N));
22     finds2size = size(finds2);
23     if finds2size(1) > 0 then
24       newout(outmin(i)) = 100 /mark battle/;

```

---

#### 5. Validation of war detection

During the validation we focused on the validation of the accuracy of the algorithm. With the validation such special cases can be revealed when the proposed algorithm makes mistakes. With the validation it is also possible to reveal how generally applicable is the proposed algorithm. The validation is difficult due to several reasons: the games can not be modified and extended to be capable of different logging activities and a mass number and wide variety of logged gaming activity from different players is difficult to obtain. In order to overcome these difficulties the validation of the proposed method has been performed in several steps.

In the *first step* we created several active measurements in a controlled environment by capturing the generated network traffic of the game and manually log the time of the battles. After the measurement the proposed method was applied on the network traces. The battles which were indicated by the proposed method and the logged battle events were compared. Overlapping periods of battle periods in the validation and the results of the algorithm are regarded as true positives. In the validation phase 15 hours of RTS gaming traffic were collected and examined. In this phase of the validation 33 out of 36 battles (92%) were detected and 3 (8%) of the indicated battles were false positive. As this phase of the validation showed promising results we extended the validation process to be able to validate with more samples.

In the *second step* of the validation process we introduced

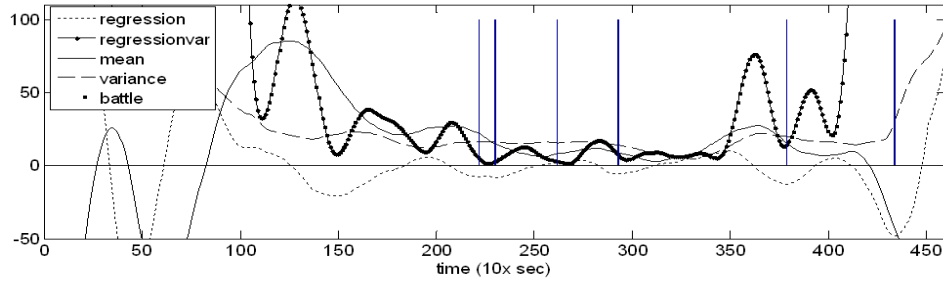


Figure 2. Analysis of Cossacks traffic and the effect of battles during gameplay /battles are indicated with the vertical poles/

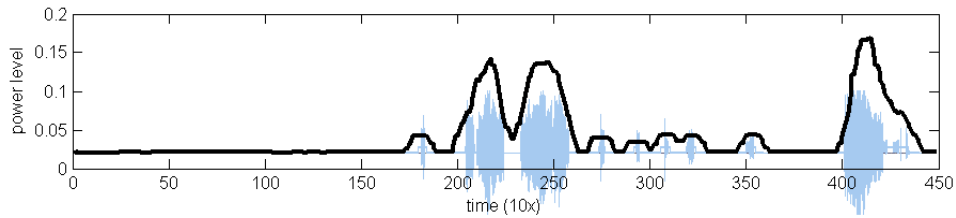


Figure 3. The recorded audio and the calculated power levels of the same Cossacks gameplay of Figure 1

an automatic method to quantify the size of the battles and to recognize battles automatically not only from the network traffic but from the game itself. It became clear that during gameplay it is not straightforward how a battle is characterized, in other words which events should be considered as battles: there are several cases when the units fire at each other, when players send scouts and they are caught, but these events are not huge battles just small conflicts. Our goal was to quantify the size of battles and this was done in the following way: the program structure of the Cossacks was examined and we managed to locate the list of sound effects of the game. All but unit dying effects were removed from the game to get rid of the e.g., unit control and management sounds. During play, beside the network traffic capture, the ingame sounds were recorded, which only contained the unit loss sounds due to our modification. The unit loss sounds are characteristic feature of a battle, furthermore, the power of the mixed sounds – as there can be more unit loss parallel – and how frequently these sounds occurred shows the size of the battle. The considerable size of the battle can be tracked by a simple limit on the power of the audio. As an example in Figure 3 the recorded audio and the calculated power levels of the same Cossacks gameplay of Figure 1 can be seen.

We created several measurements with Cossacks, but to examine how generally the method is applicable we extended the validation for different games. We used Spring [12] to validate thoroughly the proposed method in bulk measurements in which case the traffic is generated by different players. The extraction of the sound effect files was also possible in Spring thus the record of the sound effects was

possible. To obtain bulk measurements we used [13] which is a collection of replay files of the game. Replays are automatically recorded for every game in Spring and the players are able to upload the interesting games to [13]. During the watching of the replay the original game is practically replayed with its sounds and as it is possible to host the replay for others to join to see, thus the network traffic is also replayed. In this phase of the validation 33 out of 38 battles (87%) were detected and 4 (11%) indicated battles were false positives.

**What we learned.** It is clear from the validation that our proposed algorithm makes mistakes sometimes, thus such events as e.g., the loss of the scouts is hardly ever detected. It was experienced that the reason for the mistakes is that a significant number of units has to be commanded and lost during a battle to make the method capable of detecting the battle event. In Figure 4 the cdf of the power level of the audio of all the battles in the RTS measurements comparing to the power level of the audio of recognized battles can be seen. As the cdf plot of the power level of the recognized battles are under the cdf plot of the power level of all the battles occurred in the game it is clear that the proposed algorithm can mainly identify the big battles.

The other aspect which would cause the algorithm to make mistakes is such a play style which neglects the management of the army. Our method builds upon the significant effects of the micromanagement in the game. The lack of the micromanagement activity would result in smooth game traffic. The high detection ratio showed us that the existence of big battles and intensive micromanagement is a realistic precondition in the RTS games.



In the *third step* of the validation process it would be important to examine how the proposed method is applicable in general. We collected some traces from [7] and also created active measurements from several other games from the RTS genre. Cossacks [10], TA Spring [12], Warcraft 3 [7], Command & Conquer Generals [7], C&C 3 [14], Age of Mythology [7], Starcraft [7] were also examined. The bigger the armies in a given game, the more confident the method is in finding the battles. The main battles were found in all of the enumerated games by the method, but certainly not all smaller scaled confrontations. One exception is Starcraft, where the proposed method does not work as it has unique game traffic characteristics among the RTS games in the sense that its traffic is similar to an FPS game, thus the battles and environment do not affect the rate. We are aware that in spite of the fact that the method works in most cases, this examination could be made more complete which can be the subject of another paper.

## 6. Comparison with real wars

A straightforward idea is to use parameters for player behavior simulations obtained from real world. We found that a possible candidate of an important simulation parameter is the battle length of the gaming world. We used this parameter and compared it to real world data.

We filtered out the gaming traffic from the total traffic of an operational broadband network with the method introduced in [15] and analyzed the filtered traffic with the methods presented in the previous sections. The examined traffic is a three-day-long passive measurement, which contained about 20 Warcraft 3 flows and 30 other RTS flows (e.g., [10], [14], etc.) containing 296 battles total. In this section we discuss one of the examinable parameters as an example. The questions which can be raised and answered from the revealed data is not restricted to these parameters of course.

In Figure 5 the distribution of battle lengths in RTS games can be seen. The short duration battles dominate the datasets for the Warcraft 3 games. In other RTS games the distribution is more scattered along the different time lengths. We compared the characteristics of the observed RTS games to real world data. We obtained the data about the war activities of the real world from the Correlates of War project [16]. [16] seeks to facilitate the collection, dissemination, and use of accurate and reliable quantitative data in international relations. We used the inter-state and intra-state dataset, which identifies interstate and intrastate wars and their participants between 1816 and 1997. In Figure 5 it can be seen that the short duration battles dominate the datasets both in the real and gaming world.

The cdfs are depicted in lin-log scale and it can be seen that the tail of the distributions are exponential both in the gaming and real world. To compare the battle durations of

the gaming and real world environment, the qq-plot of the two data series was depicted in Figure 6. There is a break point at 0.7 year of the real world and 2.25 minute of the gaming environment. If the data series are divided by these limits and replotted the two qq-plots it could be seen that the distributions are similar to each other in these intervals as they fall onto a linear.

*The potential reason* for the short gaming session durations (see Figure 5) can be the difference of experience between the average or weaker than the average players. A weaker player is usually not able to withstand for long the attacks of a more experienced player. Comparing the tail of the duration of the real world battles and RTS sessions it is clear that renewal resources models the real world battles better than Warcraft 3 non-renewal resources as there is no such cut-off in real world cdf as in the case of Warcraft 3.

The exponential tail of the distributions in Figure 5 can be interpreted in the following way: when two armies fight against each other, when they both reach a 20% loss, both retreat and continue at a later time. A longer tailed distribution can occur in such a case that during the battle when one of the armies feel a disadvantageous situation it instantaneously retreats with as small loss as possible, so there is not significant loss in the armies thus the chance of a new battle remains the same as it was before the battle.

The break point of the qq-plot in Figure 6 supports the idea that people tend to take part in only short duration battles as the long battles are expensive and risky. Other reason for the preferred short duration battles both in gaming and real world environment can be due to the difficulties in controlling the armies in the long run after the battle started and the armies smashed together. To overcome this, the armies are regularly retreated to reorganize the scattered soldiers both in the real world and gaming world environment.

## 7. Conclusion

Complex game server architecture is required by the growing number of Massively Multiplayer Online Real Time Strategy (MMORTS) games to make the enormous number of state information transmission feasible. The architecture design is supported by network traffic simulations based on accurate player behavior characterization.

The major contribution of this paper is to make these simulations possible by extracting the necessary parameters with a method identifying battles from passive measurements. The proposed method is based on the analysis of game traffic characteristics. The method grabs the increase in bandwidth in the gathering of forces phase of the battle and the fallback in bandwidth due to the mass loss of units during the battle. The proposed method was validated in several steps and proved to be a general method to work well with the majority of games in the RTS game class.

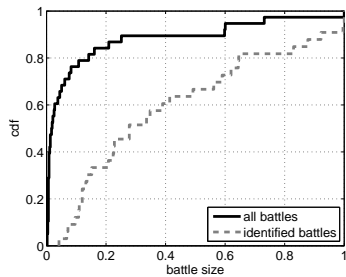


Figure 4. Distribution of the power level of the audio of all the battles comparing to the recognized ones

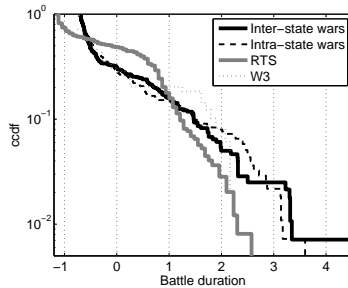


Figure 5. Normalized CCDF of battle durations

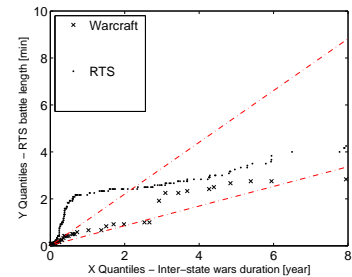


Figure 6. The qq-plot of real world battle durations and in-game battle durations

The suggested method was also applied on traffic traces obtained from live operational broadband networks in order to analyze the player behavior in real network situation. It was found that the distribution of battle durations shows similarities in the gaming and real world environment. We demonstrated that the gaming environment models the real world well but it is not a complete copy of that. This means that parameters obtained from real world can be used to some extent but the most exact results can be obtained by the investigation of the gaming environment.

Another significant contribution of the paper is in connection with traffic modeling and traffic profiling work. We showed that the player interaction can so strongly influence the traffic characteristics of a game that it can not be regarded as an approximately fix rate UDP stream as it is used in previous works but more detailed game genre specific traffic models are needed.

There are plenty of possible directions for further work. An option would be to compare several other parameters of the gaming environment with the real world. The battles and tactics can be analyzed in a lower and more detailed level of the gaming environment and can be compared to real world events. We are also aware that the number of wars in the gaming world and consequently the battles that we found are relatively low thus, the analysis of a larger dataset containing high number of wars in the gaming world is a candidate for further work.

**Acknowledgements.** The third author is supported by János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

## References

- [1] M. Ye and L. Cheng, "System-performance modeling for massively multiplayer online role-playing games," *IBM Syst. J.*, vol. 45, no. 1, pp. 45–58, 2006.
- [2] F. R. Cecin, J. V. Barbosa, and C. F. R. Geyer, "A communication optimization for conservative interactive simulators," in *IEEE Communications Letters Vol. 10, No. 9*, 2006, pp. 686–688.
- [3] Y. Lee, S. Agarwal, C. Butcher, and J. Padhye, "Measurement and Estimation of Network QoS Among Peer Xbox 360 Game Players," in *PAM*, ser. Lecture Notes in Computer Science, M. Claypool and S. Uhlig, Eds., vol. 4979. Springer, 2008, pp. 41–50.
- [4] W. Feng, D. Brandt, and D. Saha, "A long-term study of a popular mmorpg," in *NetGames '07*. New York, NY, USA: ACM, 2007, pp. 19–24.
- [5] M. Claypool, D. LaPoint, and J. Winslow, "Network Analysis of Counter-strike and Starcraft," *IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Apr 2003.
- [6] T. Beigbender, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool, "The Effects of Loss and Latency on User Performance in Unreal Tournament 2003," in *NetGames '04*. New York, NY, USA: ACM Press, 2004, pp. 144–151.
- [7] "List of gaming traces, 2007," <http://perform.wpi.edu/downloads/>.
- [8] S. Fernandes, R. Antonello, J. Moreira, and C. Kamienski, "Traffic Analysis Beyond This World: the Case of Second Life," in *NOSSDAV '07*, Urbana, Illinois, USA, June 2007.
- [9] M. Claypool, "The Effect of Latency on User Performance in Real-time Strategy Games," *Comput. Netw.*, vol. 49, no. 1, pp. 52–70, 2005.
- [10] "Cossacks, 2005, [Apr, 2009]," <http://www.cossacks.com/>.
- [11] "Matlab, [Apr, 2009]," <http://www.mathworks.com/>.
- [12] "TA Spring, [Apr, 2009]," <http://spring.clan-sy.com/>.
- [13] "Spring Replays, [Febr, 2009]," <http://replays.unknown-files.net/>.
- [14] "Command and Conquer 3, [Apr, 2009]," <http://www.commandandconquer.com/default.aspx>.
- [15] G. Szabó, I. Szabó, and D. Orincsay, "Accurate traffic classification," in *Proc. IEEE WOWMoM*, Helsinki, Finland, June 2007.
- [16] "Correlates of War Project, [Apr, 2009]," <http://www.correlatesofwar.org/>.