Optimizing multiplayer gaming protocols for heterogeneous network environment

Zsolt Kenesi[1] Gábor Kiss[2] János Levendovszky[3] Sándor Molnár[4]

Abstract— **The paper is concerned with a novel adaptive game server protocol optimization to combat network latencies in the case of heterogeneous network environment. In this way, game playing becomes feasible for clients accessing the game via different networks, which can pave the way to securing a higher income for the game industry and service providers. The sever based game protocol is viewed as a number of arrival processes from the clients (which are characterized by different delays) and periodical updates sent by the server to the clients. Game quality is quantified by two measures: (i) the tail probability of the maximal idle time; and (ii) the probability of missing an update period. Our objective is to chose server update time subject to minimizing the probability of the maximal idle time exceeding a certain threshold (which threshold is associated with the "psycho-physically approved quality of the game"). In order to calculate the optimal server update time and the underlying tail distribution, statistical tools from large deviation theory are used. Furthermore, an adaptive on-line algorithm has been developed which can adjust the server update time by estimating the corresponding delay probability density functions based on past observations. Due to the new method, game quality can be significantly improved despite the wide range of client latencies which typically characterize the heterogeneous network environment. In this way, more players can be served which can further increase the business potential of network games. The performance of the new method has been evaluated by using measurements and extensive simulations.**

I. INTRODUCTION

Multiplayer network games form a rapidly growing segment of the computer game industry. The popularity of online gaming applications leads to increasing revenues in this market. According to a report from DFC Intelligence, the worldwide online game market is predicted to grow from USD3.4 billion in 2005 to over USD13 billion in 2011 [1]. During this time period North America is expected to challenge the current market leader, Asia, as becoming the leading region for online games. The subscription revenue, which is only one part of the online game business, was USD2 billion in 2005 and is expected to grow to USD6.8 billion by 2011. Additionally, many popular games like first-person shooters and sports or racing games are increasingly played online charge-free. The growth in this market and in the client base indicates a growing demand to access online game services in heterogeneous network environment. The network operators are also interested to provide a good online game access to increase their subscriber base and traffic. As a result, the extension of game services to heterogeneous networks has a great business potential, however it presents a technical challenge due to the wide range of delays associated with clients from different networking environment. In this paper we develop a new game server protocol optimization to compensate these delays.

Most of the online games (typically First Person Shooter (FPS) and Real Time Strategy (RTS)) support a lot of simultaneous players which requires increasing network and computational resources. Consequently, these games are designed for good network connections, e.g., in LAN environment. In the present paper, we investigate how to provide good quality for clients in heterogeneous environment when they can access the server not only via LAN but 3G mobile network as well.

When extending games to heterogeneous networks, latency and related quantities (such as jitter, packet loss) have long been identified as primary obstacles, which can fundamentally impair the gaming quality [2], [3]. In the literature several research studies have been conducted to analyze the effect of changing network parameters on the quality [4], [3]. To combat latencies and ensure tolerable gaming quality and fairness, a number of different latency compensation techniques were introduced which can be classified into three major groups (for more details see [5], [6]):

- predicting the game status (each client predicts the server response instead of waiting for the status information);
- introducing processing delays (the server delays the processing of client status information);
- applying time warping (the server applies a time-roll-back mechanism to consider game actions which arrive with large delays)

These mechanisms are either very complex (prediction algorithms are running on the client side) or they may undermine gaming consistency (e.g. in time warping, the rolled back game status may be in conflict with the game status already confirmed to the clients).

When developing our novel latency compensation technique, we focus on a server side solution, as it has the advantage of running only one centralized algorithm and it avoids putting unnecessary computational overhead onto the clients. In our approach, latency compensation will be treated as an optimization problem on the server side tackled by a recursive optimization algorithm. We seek the optimal server

---

[1] Ericsson Hungary Ltd., Traffic Analysis and Network Performance Laboratory, H-1037 Laborc u. 1, Budapest, Hungary. Email: zsolt.kenesi@ericsson.com

[2] Budapest University of Technology and Economics, Dept. of Telecommunications, H-1117 Magyar tud. krt. 2, Budapest, Hungary. Email: kissg@hit.bme.hu

[3] Budapest University of Technology and Economics, Dept. of Telecommunications, H-1117 Magyar tud. krt. 2, Budapest, Hungary. Email: levendov@hit.bme.hu

[4] Budapest University of Technology and Economics, Dept. of Telecommunications and Media Informatics, H-1117 Magyar tud. krt. 2, Budapest, Hungary. Email: molnar@tmit.bme.hu

update time subject to the criterion of minimizing the tail probability of the maximal idle time. In this optimization process we also take into account the loss probability (i.e. too short update period for minimizing the idle time can increase the loss of ”slow” clients). In the forthcoming mathematical treatment, the p.d.f. of the measured delay processes will be approximated by radial basis functions and the tail probability of the maximal idle time is expressed analytically as a function of the server update time. The optimal service update time is found by performing gradient search on this function. Furthermore, by recursively updating the delay p.d.f. estimations with the measured delays, an adaptive sever update algorithm can be developed which helps to optimize the sever update time even in the case of clients with unknown latencies. In this way, the gaming quality can be improved and even heterogeneous clients can enjoy rather similar gaming experiences.

The paper is organized as follows: (i) Section 2 summarizes the effect of heterogeneous networks on the gaming quality;(ii) Section 3 describes the game protocol as an arrival processes and periodic updates; (iii) Section 4 introduces a statistical model to quantify gaming quality and to optimize the tail probability of the maximal idle time; (iv) Section 5 gives a detailed performance analysis based measurements and extensive simulations; (v) Section 6 draws some conclusions about the possible applications of the new methods; while (vi) the Appendix briefly summarizes the computational process and numerical tools used for the server update time optimization.

II. EFFECT OF NETWORK LATENCIES ON GAMING QUALITY

In this section the effect of delays originating from heterogeneous network accesses are analyzed.

In a LAN environment the latency is typically small and exhibits homogenous characteristics. However, in heterogeneous network environment the following attributes are to be taken into account:

- the clients experience different latency to a game server;
- the clients experience different latency in different game sessions;
- the clients experience a large delay variation within a game session.

These attributes are explained by the fact that delay and jitter values depend even in the internet quite much on the location of the nodes. The delay between different regions can be around 100-400ms. Thus, online game providers install servers in different regions to provide acceptable gaming quality. For example, clients connected to Quake3 servers in a 200ms latency ’radius’ of Internet in 2001 [7].

Delay and jitter can be an even more important issue in mobile networks. The general characteristics and classification of 2G and 3G delays are discussed in [8] and [9]. However, the measured delay and jitter varies at different operators based on their system version, configuration, traffic load, etc. The RTT values are typically in the order of 100ms in 2G and in the order of 10ms in 3G networks [10], [11], [6], [12]. Even though there is a continuous improvement of delay characteristics in mobile networks, the delay variation is still typically higher than in fixed environment.

There are publications about the tolerated delay and jitter for FPS games, claiming that there is 139ms defined as maximum delay for mobile real-time games in [2]. In [3] a delay bound of 150ms is defined for Halflife, whereas in [13] 300ms delay maximum is given for RTS games, e.g., Age of Empires. In [10] a maximum acceptable end-to-end delay was evaluated between 100ms and 200ms. In [4] the effect of latency on online Madden NFL Football was studied and the authors concluded that there is little impact from latency on client performance with latencies as high as 500ms. However, with latencies higher than 500 ms the performance can degrade by almost 30 percent.

III. CHARACTERIZATION OF SERVER UPDATE PROCESS

In this section the game protocol is described as arrival and update processes, i.e. game actions are arriving from the clients, whereas the sever sends periodical updates to the clients about the game status. The reason for applying periodical update process has been well discussed in the literature (for further details see [14]) and the periodicity is also supported by measuring different online games [10], [15].

More precisely, the protocol is described as follows:

- there is a client population $i = 1, ..., N$;
- the server updates the status of the game periodically, after each update period (the duration of this period is denoted by T);
- a packet (client action) arriving later than an update period is discarded and regarded as a loss;
- the clients send new packets (their gaming actions) upon receiving updates from the server;
- the arrived packets have to wait till the next update to be validated and this waiting time is the client idle time.

The operation of the protocol is depicted on Figure 1 - A. Our concern is to choose an optimal T which guarantees an acceptable quality for each client participating in the game despite the different network latencies.

It can be easily seen that the loss can indeed be minimized by increasing T, since there is enough time to receive the packets of each client (even though some of them may have large delays). However, in this case the idle time is also increased, as the client with small access delay has to remain idle for a long period, till the packets generated by large-delay clients are received. As a result, the protocol in this form cannot cope with heterogeneous clients and a wide variety of delays.

Thus, an extended protocol is introduced which enables the server to accept packets arriving in a period of length $2T$, whereas sends update with a period of either T (the packet of each participant is received within the interval $[0, T]$), or $2T$ (there are some participants whose packets are received within the interval $[T, 2T]$). With this extension the loss can be further decreased, however ”fast” clients do not suffer from large idle periods.

This extension is described as follows:

- there is a client population $i = 1, ..., N$;
- the server updates the status of the game periodically;

- a packet (client action) arriving later than two update period $2T$ is discarded and regarded as a loss;
- the server sends an update after T if the packets of all participants arrived earlier than T, or sends an update after $2T$ if there is at least one client packet received within the interval $[T, 2T]$;
- the clients send new packets (their gaming actions) upon receiving updates from the server;
- the arrived packets have to wait till the next update to be validated and this waiting time is the client idle time.

The operation of the extended protocol is depicted by Figure 1 - B.

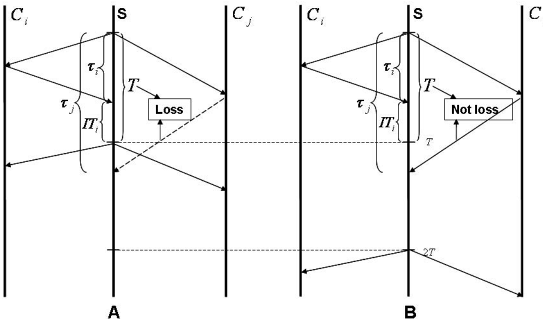

Fig. 1. Protocol model based on T or $2T$ period

When evaluating the game quality we consider three main parameters in a game session: loss probability, average idle time and the tail probability of maximal idle time.

Figure 2 and Figure 3 show these quality measures (loss rate, average idle time and tail probability) as a function of the server update period T, in the case of applying the single T protocol and the extended $2T$ protocol, respectively. In the simulated case, there are three clients with RTT values measured in Internet with 100byte packets participating in the game. The statistics of the client RTTs applied in the simulation are given as follows: minimum values are between 112.4-125.6msec, average values are between 138.2-154.8msec, while the maximum values are between 1237.7-1261.5msec.

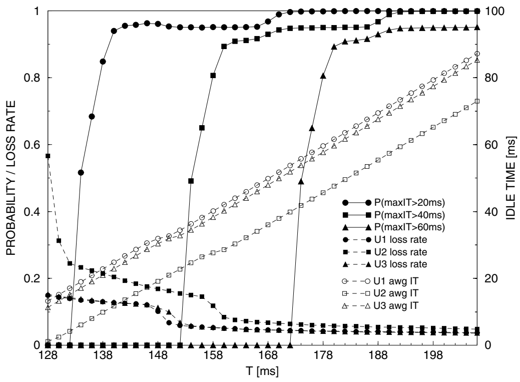

Fig. 2. Quality measures by T server period

It can be clearly seen that there is a great role for optimization as the tail probability of the maximal idle time exceeding a specific threshold (i.e. 20ms, 40ms and 60ms) exhibits a definite minimum with respect to T. Further analysis of the figures will be given in Section 5. Thus, in the next sections our objective is to develop a formal model and optimization algorithm to find this specific T for clients with heterogeneous network environments in the case of the extended protocol.

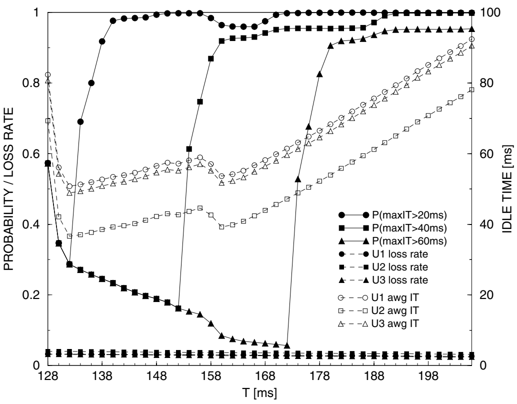

Fig. 3. Quality measures by T server period (extended protocol)

IV. SERVER UPDATE TIME OPTIMIZATION BY STATISTICAL TOOLS

In this section we embark on optimizing the server update time by using a formal approach and statistical considerations. The underlying model and the optimization method are discussed in the following two paragraphs.

A. The model

To model the gaming problem we introduce the following notations:

- the server works in a synchronous fashion and the state-update period is denoted by T;
- there is a client population $i = 1, ..., N$ and each client accesses the server with a random delay denoted by $\tau_i\,, i = 1, ..., N'$, where N' denotes the subset of clients whose delay is smaller than $2T$;
- $f_i(t)$ denotes the probability density function of random variable τ_i (the delay of client i), whereas $F_i(t)$ is the corresponding probability distribution function;
- random variable ξ represents the maximum delay among the clients, i.e. $\xi := \max_i \tau_i$;
- the idle time for client i within a server update period is denoted by η_i.

In order to develop an analytical model the following assumptions have been made:

- $f_i(t), i = 1, ..., N$ are known (later this assumption will be relaxed by devising adaptive schemes based on measurements);
- the client misses an update if the access delay is longer than $2T$.

With this assumption the idle time for client i is given as

$$\eta_i := \begin{cases} T - \tau_i & \text{if} \quad \xi \le T \\ 2T - \tau_i & \text{if} \quad T < \xi \le 2T. \end{cases} \tag{1}$$

The objective is to optimize the server update time T in order minimize the probability that the maximum idle time is larger then a predefined quantity A, where A refers to the quality of the game. More precisely, gaming optimization amounts to solving the following problem:

$$T_{opt} : \min_T P\left(\max_i \eta_i > A\right) \tag{2}$$

B. *Optimization of the server update time*

In order to carry out this optimization task, one has to express $P\left(\max_i \eta_i > A\right)$ as a function of the idle time. This dependence is derived by using the union bound first:

$$P\left(\max_i \eta_i > A\right) \leq \sum_{i=1}^{N} P\left(\eta_i > A\right). \tag{3}$$

Furthermore, the $P\left(\eta_i > A\right)$ probability can be expanded by using the conditional probabilities as follows:

$$\begin{aligned} P\left(\eta_i > A\right) = P\left(\eta_i > A|\tau_i \neq \xi\right) P\left(\tau_i \neq \xi\right) + \\ +P\left(\eta_i > A|\tau_i = \xi\right) P\left(\tau_i = \xi\right) \end{aligned} \tag{4}$$

When analyzing the expression above we must distinguish the following cases:

1) the access delay of client i is smaller than the maximum access delay, i.e. $\tau_i < \xi$;
2) the access delay of client i is the maximum access delay, i.e. $\xi = \tau_i$.

Analyzing the first case, the event $\eta_i > A$ can occur under the following assumptions:

- $\xi < T - A$;
- $T - A < \xi < T$;
- $T < \xi < 2T - A$;
- $2T - A < \xi < 2T$.

Taking into account the first assumption $\xi < T - A$ the probability $P\left(\eta_i > A \mid \tau_i \neq \xi\right)$ can be rewritten as follows:

$$P\left(\eta_i > A \mid \tau_i \neq \xi\right) = P\left(\tau_i < T - A \tau_i \neq \xi\right) \Rightarrow$$

$$\int_0^{T-A} P\left(\tau_i < t|\xi = t\right) f_\xi\left(t\right) dt = \int_0^{T-A} P\left(\tau_i < t\right) f_\xi\left(t\right) dt =$$

$$= \int_0^{T-A} F_i\left(t\right) f_\xi\left(t\right) dt \tag{5}$$

When the assumption $T - A < \xi < T$ holds than

$$P\left(\tau_i < T - A\,|\eta_i \neq \xi\,\right) = P\left(\tau_i < T - A\right) \int_{T-A}^{T} f_\xi\left(t\right) dt =$$

$$= F_i\left(T - A\right) \int_{T-A}^{T} f_\xi\left(t\right) dt \tag{6}$$

The same line of reasoning can be applied when $T < \xi < 2T - A$ yielding

$$P\left(\tau_i < 2T - A\,|\eta_i \neq \xi\,\right) = \int_T^{2T-A} P\left(\tau_i < t|\xi = t\right) f_\xi\left(t\right) dt =$$

$$= \int_T^{2T-A} P\left(\tau_i < t\right) f_\xi\left(t\right) dt = \int_T^{2T-A} F_i\left(t\right) f_\xi\left(t\right) dt \tag{7}$$

or in the case of $2T - A < \xi < 2T$ one can obtain

$$P\left(\tau_i < 2T - A\,|\eta_i \neq \xi\,\right) = P\left(\tau_i < 2T - A\right) \int_{2T-A}^{2T} f_\xi\left(t\right) dt =$$

$$= F_i\left(2T - A\right) \int_{2T-A}^{2T} f_\xi\left(t\right) dt. \tag{8}$$

If the condition $\xi = \tau_i$ holds then

$$P\left(\eta_i > A|\tau_i = \xi\right) = P\left(\xi < T - A\right) + P\left(T < \xi < 2T - A\right) =$$

$$= F_\xi\left(T - A\right) + F_\xi\left(2T - A\right) - F_\xi\left(T\right) \tag{9}$$

The probabilities of the conditions $\xi = \tau_i$ and $\xi \neq \tau_i$ are given as

$$P\left(\tau_i = \xi\right) = P\left(\tau_i > t \bigcap_{j \neq i} \tau_j < t\right) = \int_0^\infty \left[f_i\left(t\right)\right] \prod_{\substack{j \\ j \neq i}}^{N} F_j\left(t\right) dt,$$

$$P\left(\tau_i \neq \xi\right) = 1 - P\left(\tau_i = \xi\right), \tag{10}$$

respectively. Thus, the probability $P\left(\eta_i > A\right)$ can finally be expressed as

$$\begin{aligned} & P\left(\eta_i > A\right) = \\ & = \left[\int_0^{T-A} F_i\left(t\right) f_\xi\left(t\right) dt + F_i\left(2T - A\right) \int_{T-A}^{T} f_\xi\left(t\right) dt + \right. \\ & \left. + \int_T^{2T-A} F_i\left(t\right) f_\xi\left(t\right) dt + F_i\left(2T - A\right) \int_{2T-A}^{2T} f_\xi\left(t\right) dt \right] * \\ & * \left[1 - P\left(\tau_i = \xi\right)\right] + \\ & + \left[F_\xi\left(T - A\right) + F_\xi\left(2T - A\right) - F_\xi\left(T\right)\right] * \\ & * \left[\int_0^\infty \left[f_i\left(t\right)\right] \prod_{\substack{j \\ j \neq i}}^{N} F_j\left(t\right) dt\right] \end{aligned} \tag{11}$$

One can see that this formula does indeed depend on T, which prompt us to perceive $P\left(\eta_i > A\right)$ as a function of T denoted by $P\left(\eta_i > A\right) = \Psi(T)$. In this way, optimizing the update period on the server reduces to a search problem tackled by the following recursion:

$$T(k+1) = T(k) - \Delta \mathrm{sgn}\left(\Psi(T(k)) - \Psi(T(k-1))\right). \tag{12}$$

One must note that if the density functions are known then finding T_{opt} is an off-line task which can be carried out prior to the game. Thus any search technique drawn from the tools of classical optimization theory can be used. However, in reality the densities $f_i(t)\; i = 1, ..., N$ are not known a priori, which prompts us to develop an adaptive technique based on only the observed delays working in an on-line during the game session. **Remark:** As mentioned earlier, one may consider not only the tail probability of the maximal idle time as the quality measure of the game but the average idle time and the loss rate, as well. Our method can easily be extended to include these measures into the optimization process.

C. Server update optimization algorithm

Based on the previous section we have arrived at a server update time optimization algorithm which can be performed on the game server. The computational process of the algorithm is summarized by Figure 4:

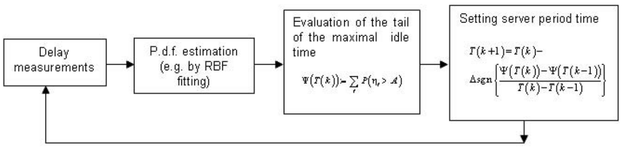

Fig. 4. Mechanism of server period time optimization

The calculations needed by equation (4) requires to measure the delays and estimate the distributions in a recursive manner, running simultaneously with the game.

The delay density functions are estimated by RBF approximation [16] according to the following steps:

- Delay measurement of client i:
  $\tau_i(k) := (t_i(k), k = 1, ..., K)$
- Calculating histograms:
  $Hist(k)$
- P.d.f function by Radial Basis function:
  $f(t, \mathbf{w}) = \sum_{j=1}^{K} w_j e^{-\frac{\left(t-t_j^i\right)^2}{2\sigma^2}}$
- Fitting the model:
  $\mathbf{w}_{opt}^{(i)} = \min_w \sum_{n=1}^{K} \left(Hist\left(t_k^i\right) - \sum_{j=1}^{K} w_j e^{-\frac{\left(t_k^i - t_j^i\right)^2}{2\sigma^2}} \right)^2$
- Using optimal $\mathbf{w}$ for p.d.f estimation:
  $f\left(t, \mathbf{w}_{opt}\right) = \sum_{j=1}^{K} w_{j,opt}^{(i)} e^{-\frac{\left(t-t_j^i\right)^2}{2\sigma^2}}$

After the RBF approximation $P\left(\eta_i > A\right)$ and recursion (11) can be calculated for each client. Recursion (11) is operating based on a gradient search which can get stuck in local optima. But simulations demonstrated (as it can be seen in Figure 5) that $P\left(\max \eta_i > A\right)$ has only one global minimum. Furthermore, in order to avoid stopping in local minima recursion (5) was started from many different initial points.

D. Adaptive server update time optimization

In the present section, we investigate an on-line approach when delay measurements during the game are taken into account. Thus, the aim is to update the delay density estimates based on the current measurements by implementing a recursive estimation $f_i\left(t, k+1\right) = \Psi\left(f_i\left(t, k\right), t_k^{(i)}\right), i = 1, ..., N$, where k refers to the fact that the p.d.f. is estimated after observing the first k measurements and $t_k^{(i)}$ denotes the kth observation of the delay of client i. Now we use a histogram estimation given as follows:

$$f_i(t, k) := \sum_{l=1}^{L} n_l(k) I_l(t) \tag{13}$$

where $n_l(k)$ is the relative frequency of the samples falling in to the interval $\Delta t_l := t_l - t_{l-1}$ and

$$I_l(t) := \begin{cases} 1 \text{ if } t \in \Delta t_l \\ 0 \text{ otherwise} \end{cases}$$

When a new measurement is taken about the delay of client i in the course of the game, the corresponding density is updated as

$$f_i\left(t, k+1\right) = \sum_{l=1}^{L} n_i(k+1) I_l(x), \tag{14}$$

where

$$n_l(k+1) = \frac{n_l(k) N(k) + s_l(x)}{N(k)+1}$$

and

$$s_l(x) = \begin{cases} 1 \text{ if } t_k^{(i)} \in \Delta t_l \\ 0 \text{ otherwise} \end{cases}$$

Performing recursion (11) in each new measurement, the server update time is optimized recursively by plugging the updated p.d.f.-s into expression (3) and (4), respectively. In this way, the server can optimize the update time based on the newly obtained delay information in the course of the game.

V. PERFORMANCE ANALYSIS

Extensive simulations have been carried out to test the performance of the server optimization method. The delay distribution functions were estimated by making more than 60000 measurements.

Figure 5 depicts the calculated tail probabilities against the measured ones as a function of the server update time T. In this figure, The approximation with the union bound is denoted by Mod1, whereas the tail calculated by assuming client delay independence is denoted by Mod2. As demonstrated by the figure the quality measure (tail probability of the maximal delay) has indeed a sharp minimum with respect to the server update time T. Furthermore, one can see that even though both Mod1 and Mod2 upper bound the real tail, the corresponding T values of the curves more or less coincide which implies that our optimization provides nearly the exact optimum.

As was mentioned before, the server update time can be optimized adaptively by algorithm 12. Figure 6 shows the convergence of adaptive server update time optimization running on 500 length ping sequence for three different cases: 3 users with mean delay of 106ms (case 1), 173ms (case 2) and 60ms (case 3). As exhibited by the curves, the server update time has quite fast convergence speed, i.e. within 100 measurement the optimal T is reached.

The simulation results clearly demonstrated that using 40-50ms (which is currently implemented in most of gaming servers [10], [15]) is very far from the optimal server update time. Based on our measured RTT values, our optimization method pointed out that optimal server update time falls in

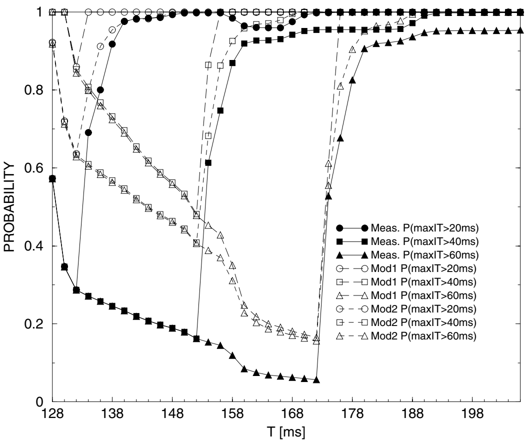

Fig. 5. Comparison of measurements and model results

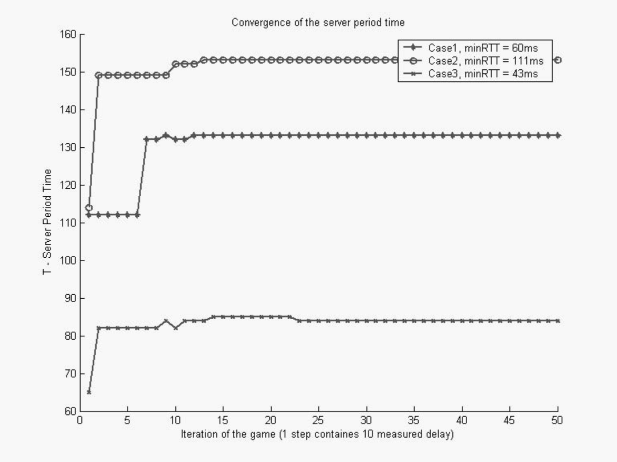

Fig. 6. Convergence of the server period time

the range of 110-150msec depending on the measured RTT values (for further numerical details see Figure 6).

In addition, it has been proven that an adaptive server update time optimization is needed to handle the various latencies being typical in heterogeneous network environment. Consequently gaming quality can significantly be improved by our method and the optimal sever update time can be achieved with a fast convergence speed.

VI. CONCLUSIONS

In our preliminary analysis we found that real-time (FPS and RTS) games typically designed for high-speed LAN or Internet connection are rather susceptible to latency. Thus latency originating from heterogeneous network environment can have a significant negative impact on gaming quality. As a result, traditional protocols are not efficient in heterogeneous environment, where latency and jitter is typically higher and changing in a broader range than in high-speed fixed network environment.

We viewed game protocol as arrival and update processes running on the server and found that the loss, average idle time and the probability that the idle time is higher than a specific value greatly depend on the choice of server update period T. Thus, server update time optimization proved to be an efficient tool to compensate latencies.

We have developed a statistical model to express the tail probability of maximal idle time as function of the server update period. Based on this model the server update time can be adaptively optimized. Our new solution works in server side, and has a fast convergence speed.

By decreasing the maximum idle time means with the new method, the game provider can support much more clients from heterogeneous network environment. Therefore, the method proposed in the paper can contribute to achieving higher revenues from network games. Moreover, finding the optimal server update time can further increase the perceived game quality and satisfaction.

VII. ACKNOWLEDGEMENT

This paper describes work partially undertaken in the context of the E-NEXT - IST FP6-506869 project, which is partially funded by the Commission of the European Union. The views contained herein are those of the authors and do not necessarily represent those of the E-NEXT project.

REFERENCES

[1] DFC Intelligence, "The Online Game Market," June 2006.

[2] C. Schaefer, T. Enderes, H. Ritter, and M. Zitterbart, "Subjective Quality Assessment for Multiplayer Real-Time Games," *In Proc. of the 1st Workshop on Network and System Support for Games (NetGames), Braunschweig, Germany*, pp. 74–78, April 2002.

[3] T. Henderson, "Latency and user behaviour on a multiplayer games server," *In Proc. of the 3rd International Workshop on Networked Group Communication (NGC), London, UK*, pp. 1–13, November 2001.

[4] J. Nichols and M. Claypool, "The Effects of Latency on Online Madden NFL Football," *In Proc. of the 14th international workshop on Network and operating systems support for digital audio and video, Cork, Ireland*, pp. 146–151, 2004.

[5] G. Armitage, M. Claypool, and P. Branch, "Networking and Online Games: Understanding and Engineering Multiplayer Internet Games," *John Wiley and Sons Ltd*, 2006.

[6] J. Korhonen, O. Aalto, A. Gurtov, and H. Laamanen, "Measured Performance of GSM HSCSD and GPRS," *In Proc. of the IEEE International Conference on Communications (ICC)*, January 2001.

[7] G. Armitage, "Sensitivity of Quake3 Players To Network Latency," *Poster session, SIGCOMM Internet Measurement Workshop, San Francisco*, pp. 199–204, November 2001.

[8] "3GPP TS 22.060 General Packet Radio Service, Service description Stage 1."

[9] "3GPP TS 23.107 Quality of Service concept and architecture."

[10] M. Busse, B. Lamparter, M. Mauve, and W. Effelsberg, "Lightweight QoS-Support for Networked Mobile Gaming," *In Proc. of the 3rd Workshop on Network and System Support for Games, NETGAMES 2004, Portland, Oregon, USA*, pp. 85–92, August 2004.

[11] A. Gurtov, M. Passoja, O. Aalto, and M. Raitola, "Multi-Layer Protocol Tracing in a GPRS Network," *In Proc. of the IEEE Vehicular Technology Conference (Fall VTC 2002), Vancouver, Canada*, September 2002.

[12] R. Chakravorty and I. Pratt, "Performance Issues with General Packet Radio Service," *Journal of Communication and Networks (JCN)*, 2002.

[13] M. Terrano and P. Bettner, "1500 archers on a 28.8: Network programming in Age of Empires and beyond," *In Proc. of the 15th Games Developers Conference, San Jose, CA, USA*, March 2001.

[14] W. Feng, F. Chang, W. Feng, and J. Walpole, "Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server," *In Proc. of 2nd Internet Measurement Workshop, Marseille, France*, November 2002.

[15] J. Färber, "Network Game Traffic Modelling," *In Proc. of the 1st Workshop on Network and System Support for Games (NetGames), Braunschweig, Germany*, April 2002.

[16] M. J. D. Powell, "The theory of radial function approximation in 1990," in *Advances in Numerical Analysis*, vol. II. Oxford: Clarendon Press, 1992, pp. 105–210.