This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE CCNC 2006 proceedings.

Buscar el Levante por el Poniente: In Search of Fairness Through Interactivity in Massively Multiplayer Online Games

Stefano Ferretti [(1)], Claudio E. Palazzi [(1,2)], Marco Roccetti [(1)], Giovanni Pau [(2)], Mario Gerla [(2)]

[1]*Dipartimento di Scienze dell'Informazione, Università di Bologna,*
Mura Anteo Zamboni 7, 40127 Bologna, Italia
{sferrett, roccetti}@cs.unibo.it

[2]*Computer Science Department,*
University of California Los Angeles, Boelter Hall, Los Angeles CA, 90095, USA
{cpalazzi, gpau, gerla}@cs.ucla.edu

***Abstract*— Ensuring fairness among players engaged in online games is a challenging task. Yet, it is a fundamental requirement that can make the difference between having customers that persist or desist in using this kind of application. Answering to this demand, we present here an event delivery mechanism among mirrored game servers able to effectively uplift the fairness degree during game sessions through the heterogenesis of ends in targeting interactivity. We also provide extensive results that sustain our claim.**

Keywords- Multiplayer Online Games; Fairness; Interactivity; Obsolescence; Networked Entertainment.

I. INTRODUCTION

Christopher Columbus's aim, when he sailed for finding Catai (ancient China) lands across the Atlantic Ocean, is perfectly described by the famous claim credited to him: "*Buscar el Levante por el Poniente*", to seek the East by way of the West. We humbly take inspiration from his genius to synthesize our work in the title of this paper. The analogy is represented by the fact that the scheme we here propose facilitates fairness by aiming at increasing the interactivity degree in Massively Multiplayer Online Games (MMOGs).

With the term *network fairness* we refer to the problem of guaranteeing the same possibility of victory to all the players, regardless of their subjective network conditions. This is obviously one of the well known key factors in designing MMOGs. Indeed, relative delays among players can be considered as important as absolute ones. If a player receives game state updates more quickly than another one, she/he will definitively be able to react more promptly and, thus, to overwhelm her/his adversary [1, 2].

Local lag and other similar algorithms have been proposed to ensure fairness (and consistency) among players in MMOGs [2-6]. The idea behind this kind of approach amounts to introducing artificial delays in the display of both generated and received game events. These delays are appropriately chosen for each client and depend on their subjective client-server latencies. The aim is that of having each game event simultaneously displayed, after a total amount of time since its creation, on all the players' screens.

Usually, this amount of time corresponds to the longest transmission latency experienced by the most unlucky player of the game. In practice, this kind of approach increases game delays and may jeopardize interactivity as in some case the unlucky client may be connected very far away from its server and/or through a slow connection.

Consequently, the efficiency and applicability of the local lag approach strongly depend on the network conditions and on the interactivity degree required by the game. Indeed, especially in the case of a highly interactive MMOG, servers should be optimally located to efficiently serve a large number of customers [1]. Yet, guaranteeing both interactivity and full fairness through local lag can sometimes be achieved only at the cost of impeding the access to some users whose connectivity is irremediably affected by large network delays.

A tradeoff relationship thus exists among scalability (especially in terms of geographical dispersion of the players), interactivity, and fairness. According to this, interactivity and fairness are traditionally seen as incompatible requirements in MMOGs.

Conversely, we claim now that upholding interactivity may be useful also to the aim of ensuring fairness. To demonstrate this, we have developed a novel mechanism named Fairness and Interactivity Loss Avoidance (FILA). Our scheme can be divided into two complementary sub-components. The first one exploits the semantics of the game to drop superseded events and speed up the delivery of game events. The second one takes advantage of this reduced transmission time to magnify the efficiency of a local lag-type algorithm in ensuring fairness without compromising interactivity.

The remainder of the paper is organized as follows. Section II presents our adopted scalable architecture. Section III provides fairness definition plus conditions for its achievement.

1-4244-0086-4/05/$20.00 ©2005 IEEE.

Section IV describes FILA's design. Section V explains our simulative environment and presents the experimental results. Finally, Section VI concludes this paper.

II. SCENARIO

A suitable architecture able to efficiently manage large-scale distributed games may make use of a constellation of mirrored Game State Servers (GSSs) which cooperate over the network, in a peer-to-peer fashion, to maintain replicas of the same game state [7]. Having multiple servers allows each client to connect in a client-server mode to the closest mirror, thus reducing the communication latency.

Each GSS receives game events from its engaged players and forwards them to all the other GSS peers. Moreover, it gathers all the game events received from the other GSS peers and sends game state updates to its clients. In essence, this approach collects the advantages of both centralized and fully distributed architecture.

Recent studies have demonstrated that exploiting the semantics of the game can be put to good use to augment interactivity whilst preserving consistency of the game state viewed by the various nodes in the system [8, 9, 10, 11]. Some events, in fact, can lose their significance as time passes: new actions may make the previous ones irrelevant. For example, where there is a rapid succession of movements by a single agent in a virtual world, the event representing the last destination supersedes the older ones.

Based on this concept, a notion of *obsolescence* was defined as the relation between two received events e' and e'', generated at different times $t' < t''$, by which the content of event e'' supersedes e' and the need for its processing. Of course, e' can be defined as obsolete by the arrival of e'' only if it is not correlated to other events concurrent with e'' [8]. In simple words, think to an event which cannot be considered as obsolete as further events may come into the picture that correlate it to the final game state.

The notions of obsolescence and correlation can be used to improve interactivity along two parallel directions: i) discarding obsolete game events to speed up the processing of fresher ones at receiving GSSs, and ii) providing a delivery of events to receiving GSSs based on correlation order rather than total order.

III. SYSTEM MODEL AND FORMAL PRELIMINARIES

As previously mentioned, we have *network fairness* in a MMOG when all the players on the network simultaneously receive every game event. Yet, it is very hard to demonstrate the existence of such a strict property since it would require contemplating all the possible cases in terms of network configuration, traffic load, players' dispersion, accidental malfunctioning, etc. for all the events during the game. Moreover, measuring network fairness would only result in binary outcomes: either achieved or not.

Therefore, to find a useful measure to help one in comparing different fairness preserving algorithms, a parameter which evaluates each transmitted game event is needed. We hence introduce the concept of *event-related fairness* which represents a situation when a single event is simultaneously displayed by all the clients. From here on, when we mention the fairness degree we intend the percentage of game events that were delivered achieving event-related fairness.

The following further definitions are needed with the aim of presenting our fairness preserving approach. As each game event travels from player to player, we call *Overall Latency* (*OL*) the amount of time elapsed since the generation of a game event by a player to its delivery at the adversary. We consider *OL* as comprised of two different values: the *Network Traversal Latency* (*NTL*) and the *Last Hop Latency* (*LHL*), i.e., $OL = NTL + LHL$.

Obviously, both *NTL* and *LHL* are measured at the receiving GSS, as depicted in Fig. 1. Following this model and considering the set C of clients simultaneously playing in the same virtual arena, the event-related fairness condition for a certain event e is satisfied if the following equation holds (where D is a unique amount of time),

$$OL_i(e) = D \quad \forall i \in C. \tag{1}$$

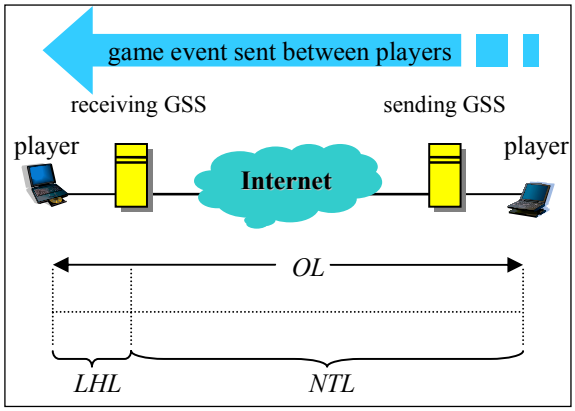


Figure 1. Delays definition.

Further, every class of games has a game specific Game Interactivity Threshold (GIT) that represents the maximum OL endurable before displaying a game event on each player's screen if one wishes to preserve interactivity. Obviously, the interactivity requirement is satisfied whenever OL is smaller than GIT. It is interesting to mention that the GIT value for fast paced games (i.e. vehicle racing, first person shooter) typically corresponds to 150-200ms but can be increased in case of slow paced games (i.e. strategic, role play game) [1, 4, 12, 13].

In the following, we present a mechanism (FILA) which is aimed at preserving fairness while achieving interactivity. It steps through two phases. The first phase implements a proactive control scheme whose aim is that of keeping $OL_i(e) < GIT$, $\forall\ i \in C$. This is accomplished by dropping obsolete events. In the second phase, a local lag-type algorithm is employed to add an appropriate artificial delay δ_i to each different $OL_i(e)$ so that

$$OL_i(e) + \delta_i = GIT\ . \tag{2}$$

If our mechanism is successful in satisfying (2), then the event-related fairness condition holds. Needless to say, if a given event traveling from a player to another one surpasses

GIT, then no artificial delays will be added to that event and the event-related fairness requirement will be not satisfied.

IV. FILA: ACHIEVING FAIRNESS THROUGH INTERACTIVITY

With FILA, game events are orderable: they are marked at their creation with a generation timestamp and then sent to the destination. Obviously, this requires the maintenance of a global conception of time among all the GSSs, which can be achieved as discussed at length in [8, 14, 15].

The first phase of our scheme takes inspiration from Active Queuing Management techniques [16] adapting their discarding algorithm to consider the game event delay at GSSs, instead of the queue size at routers. In essence, upon every new game event arrival, each GSS determines the *NTL* of the relative event.

This *NTL* is utilized to calculate a sample employed to update the value of a variable named *avgOL*. This variable represents the average *OL* that the considered game event is expected to have when it will finally reach all the players engaged by that GSS.

With FILA, all the game events are regularly processed and forwarded while *avgOL* is smaller than a threshold named *tmin* (an alert threshold whose value is smaller than GIT). As soon as *avgOL* exceeds *tmin*, instead, the GSSs drop obsolete events with a certain probability p which is directly proportional to *avgOL*, while neither processing nor forwarding them. Finally, if *avgOL* exceeds the subsequent GIT, then *p* is set equal to 1, and all obsolete events waiting for being processed are discarded. Interested readers can find a detailed rationale of the design choices related to a similar algorithm in [8].

The value of *avgOL* at iteration *n* is computed according to the following low pass filter:

$$avgOL_n = avgOL_{n-1} + w \times (sample_n - avgOL_{n-1}). \qquad (3)$$

In *(3)*, *w* is a coefficient that determines how close the *avgOL* sequence follows the sample trajectory. Instead, sample is computed as follows:

$$sample = GTD + \min\{DUB, \max_{i \in C\_GSS}\{LHL_i\}\}. \qquad (4)$$

This formula represents an estimation of the maximal latency $\max_{i \in C\_GSS}\{LHL_i\}$ experienced to reach the most unlucky player *i* belonging to the set of all the players connected to that GSS (*C_GSS*).

However, we cannot let some irremediably delay-affected client to excessively impact on *(3)* raising the values of *avgOL*. In this case, we would discard an oversized amount of game events with no real advantages. For this reason, *(4)* includes a global value as a maximal limit for *LHL*. We term this limit *Delay Upper Bound* (*DUB*).

As *DUB* represents a global value within the system, an open problem remains on how to appropriately set it. We report here on a heuristic we use to dynamically compute *DUB*. The formula for its computation is as follows:

$$DUB = GIT - \max\{NTL\}, \qquad (5)$$

where $\max\{NTL\}$ represents the largest among the *NTLs* experienced over all the connections within the entire network. Obviously, each GSS has to communicate back to all the other peers the largest *NTL* experienced at that server. This allows a global knowledge of the worst *NTL* value endured by each GSS.

Finally, the highest among these maximum *NTLs* can be univocally determined by each of the GSSs and used to determine the global *DUB*. Summarizing, each time a given GSS receives a new game event from some player connected to one of its peers, it computes the new value of sample as in *(4)* and feeds *(3)* with it to update the discarding probability *p*.

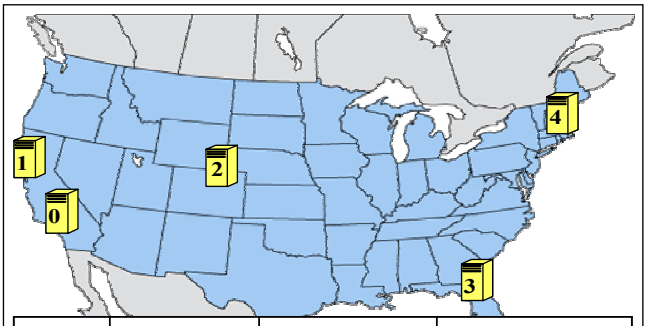
The second and final part of our scheme is simply in charge of equalizing the delay differences among players with a local lag-type scheme that appropriately computes the value of δ_i so as to satisfy *(2)*.

The aim of the next section is to demonstrate how the combination of phase one and two is effective to ensure fairness and interactivity.

V. SIMULATION ASSESSMENT AND RESULTS

It is well known that MMOG service providers should appropriately position their game servers in such a way that their target player market would be located within a circle having 150-180ms of latency diameter [1]. Following this rule and aimed at creating a configuration able to factually support a highly interactive MMOG, we have simulated a constellation of five GSSs deployed across U.S.A. by choosing optimal market locations.

Clients are supposed to be distributed all over the North American continent connecting through various access technology and thus enduring different access delays. We have focused our attention on the event receiving aspect of a single GSS (GSS_0), pretending that the other GSSs are sending events to it (without any loss of generality).



GSS	Avg latency	City	Road distance
0	0 ms	Los Angeles	0 km
1	20 ms	San Francisco	602 km
2	35 ms	Denver	1650 km
3	50 ms	Orlando	3870 km
4	65 ms	Boston	4896 km

Figure 2. Game servers deployment.

Inspired by the literature [12], the *NTL* values were chosen based on a lognormal distribution whose approximate average was obtained by means of repeated runs of the ping application. More in detail, game events coming from clients connected to the sending GSSs (i.e. GSS_1–GSS_4) and traveling towards GSS_0 experience average latencies as reported in Fig. 2, with a standard deviation of 10ms. Further, several scenarios were considered where the values of $\max_{i \in C\_GSS} \{ LHL_i \}$ were chosen for each GSS within the following set [25ms, 50ms, 75ms, 100ms, 125ms, 150ms]. This choice simply derives from the consideration that clients should be located within a circle having a maximum latency diameter of 150ms. We assumed to have 10 clients connected to each GSS, engaged in a fast-paced game, and generating a new action every 300ms in average. The average game event size (200 Bytes) was inspired by literature about games as well [17]. This results in a flow of game events having 30ms of inter-departing time.

The probability that an event makes obsolete preceding ones was set to 90%. This represents a realistic scenario for a vast plethora of possible games (e.g. adventure, strategic, vehicle race, flight simulator, etc.), where most of the events are just independent movements. In other words, critical (correlated) game events that cannot become obsolete have to be considered only sporadically, such as during collisions or shots, and may represent even less than the 10% of the whole set of game events.

As a confirmation of this claim, an extensive study of players' behavior on Quake 3 is presented in [1]. In that paper, a measure of the average number of kill actions per minute as a function of the median ping time between client and server is reported. Using those measures we provided in [9] a numerical explanation that demonstrates how 10% of correlated events and 90% of obsolescence probability may represent a realistic scenario for interactive MMOGs. As to the parameters in the FILA algorithm, we have set w = 1/8 for all the simulations. The alert threshold tmin was equal to *GIT* – 100ms.

Each experiment was identically replicated to compare the outcomes of FILA against the regular local lag algorithm. In [2], Zander *et al.* demonstrated that there is a statistically significant difference between the mean kill rates of player groups which are affected from diverse client-server latencies. In essence, lower latencies results in higher mean kill rates and thus in unfairness. Coherently, we have chosen to evaluate as a performance parameter the percentage of events that were delivered by GSS_0 to *all* of its players, thus achieving event related fairness.

Fig. 3, 4, 5, and 6 show, respectively, four different sets of experiments, obtained varying the *GIT* from 150ms to 300ms. Each set of experiment was comprised of six different experiments. Each experiment consisted in the transmission of about 4000 game events which experienced, in the worst case, a maximal overall latency whose value is reported on the x-axis of each provided chart.

The leftmost graphs of Fig. 3, 4, 5 and 6 show the percentage of game events that GSS_0 was able to deliver to all of its engaged players in time to be simultaneously delivered with an *OL* lower than *GIT*. It hence represents the amount of events which satisfied condition (2) and were thus fairly processed by all the clients. As can be seen from these graphs, having a higher *GIT* improves the efficacy of both the evaluated schemes since larger local lags can be utilized. However, regular local lag algorithm experiences a premature performance decrease when the maximal overall latency increases even if it is still far from the *GIT*. Instead, FILA ensures a good fairness degree for a larger set of overall latencies.

Obviously, in those configurations where the maximal overall latency is close to (or surpasses) *GIT*, both schemes cannot overwhelm network conditions, thus achieving poor fairness (and interactivity). Even in this case, however, FILA behaves better than the regular local lag algorithm. FILA pays these better results with the drops of some obsolete events.

Specifically, the rightmost charts of Fig. 3, 4, 5, and 6 reveal on the y-axis the percentage of game events which were discarded by FILA. In all the considered cases, less than 20% of the game events were dropped and these events were exclusively obsolete ones.

Results turn out to be even better if we focus only on those cases where the overall latency is not irremediably high with respect to *GIT*. Considering the configurations when the maximal overall latency is lower than *GIT* by 35ms or more, we find that FILA always guarantees more than 86% of fairly delivered game events with less than 15% of dropped events.

VI. CONCLUSION

The ever increasing number of MMOG subscriptions demand for new technology aimed at solving the key problems in online games and ensuring a pleasant experience to customers. In this context, fairness among players has been shown to be as important as other issues (i.e. interactivity, consistency, and scalability).

We have hence designed FILA, an event delivery scheme enforced among mirrored game servers, which exploits the notions of obsolescence to ensure fairness while achieving interactivity. As to the event dropping activity, it should be mentioned that FILA drops only obsolete events. This reduces delivery delays without causing inconsistencies in the game evolution.

As only superseded events are discarded, there is no risk that different dropping percentages at different servers could result in some unfairness. This is a further prominent result of our scheme. We provided experimental outcomes that demonstrate the efficacy of FILA with various latencies.

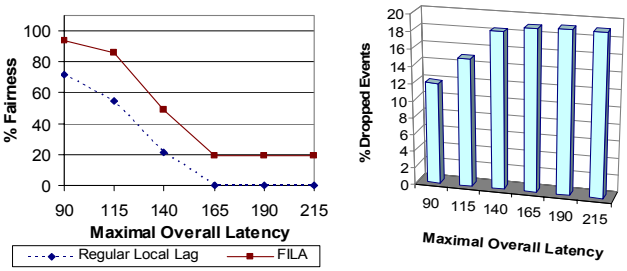


Figure 3. Fairness improvement (left) and dropped events (right) with *GIT*=150ms.

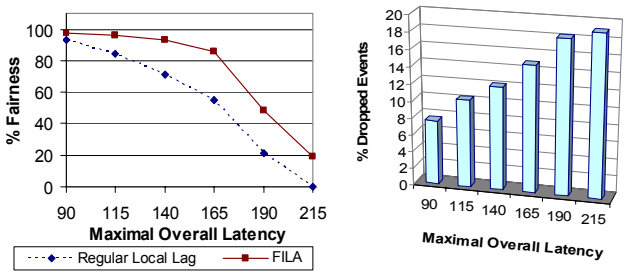


Figure 4. Fairness improvement (left) and dropped events (right) with GIT=200ms.

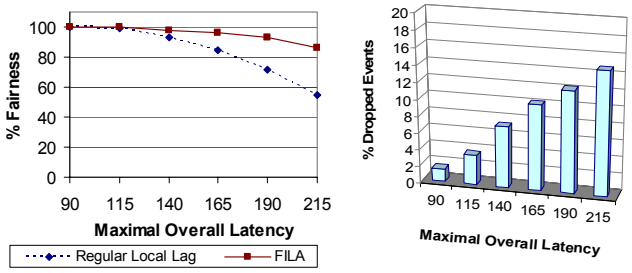


Figure 5. Fairness improvement (left) and dropped events (right) with GIT=250ms.

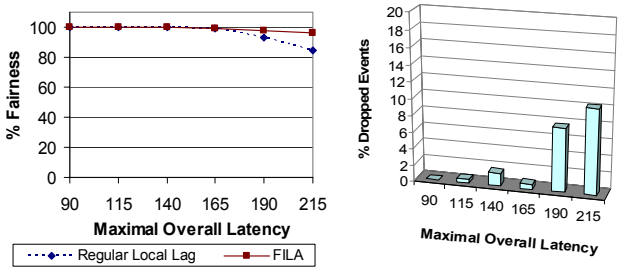


Figure 6. Fairness improvement (left) and dropped events (right) with GIT=300ms.

ACKNOWLEDGMENT

This work is partially supported by the Italian Ministry for Research via the ICTP/E Grid Initiative and the Interlink Initiative, the National Science Foundation through grants CNS-0435515/ANI-0221528, and the UC-Micro Grant Micro 04-05 private sponsor STMicroelectronics.

REFERENCES

[1] G. Armitage, "An Experimental Estimation of Latency Sensitivity in Multiplayer Quake 3", in *Proc. of ICON*, Sydney, Australia, 2003, pp.137-141.

[2] S. Zander, I. Leeder, G. Armitage, "Achieving Fairness in Multiplayer Network Games through Automated Latency Balancing", in *Proc. of ACM SIGCHI ACE2005*, Valencia, Spain, 2005 pp. 117-124.

[3] L. Gautier and C. Diot: "Design and Evaluation of MiMaze, a Multiplayer Game on the Internet", Proc. IEEE Multimedia (ICMCS'98), Austin, TX, USA, 1998, pp. 233-236.

[4] L. Pantel, L. C. Wolf, "On the Impact of Delay on Real-Time Multiplayer Games", in *Proc of NOSSDAV 02*, Miami, FL, USA, 2002, pp. 23-29.

[5] M. Mauve, J. Vogel, V. Hilt, W. Effelsberg, "Local-lag and Timewarp: Providing Consistency for Replicated Continuous Applications", *IEEE Transactions on Multimedia*, vol. 6, no. 1, 2004, pp. 47-57.

[6] S. Kim, F. Kuester, K. H. Kim, "A Global Timestamp-Based Approach to Enhanced Data Consistency and Fairness in Collaborative Virtual Environments", *Multimedia Systems*, vol. 10, no. 3, 2005, pp. 220-229.

[7] E. Cronin, A. R. Kurc, B. Filstrup, S. Jamin, "An Efficient Synchronization Mechanism for Mirrored Game Architectures", *Multimedia Tools and Applications*, vol. 23, no. 1, 2004, pp. 7-30.

[8] C.E. Palazzi, S. Ferretti, S. Cacciaguerra, M. Roccetti, "On Maintaining Interactivity in Event Delivery Synchronization for Mirrored Game Architectures", *Proc. of NIME'04,* Dallas, TX, USA, 2004, 157-165.

[9] C.E. Palazzi, S. Ferretti, S. Cacciaguerra, M. Roccetti, "Interactivity-Loss Avoidance in Event Delivery Synchronization for Mirrored Game Architectures", accepted with minor revisions in *IEEE Transactions on Multimedia*.

[10] C.E. Palazzi, S. Ferretti, S. Cacciaguerra, M. Roccetti., "A RIO-like Technique for Interactivity Loss Avoidance in Fast-Paced Multiplayer Online Games", *ACM Journal of Computers in Entertainment, ACM Press, 3,* 2, (2005), 1-11.

[11] S. Ferretti, M. Roccetti, "A Novel Obsolescence-based approach to Event Delivery Synchronization in Multiplayer Games", *International Journal of Intelligent Games and Simulation, 3,* 1, (2004), 7-19.

[12] M. S. Borella, "Source Models for Network Game Traffic", *Computer Communications*, Elsevier, vol. 23, no. 4, 2000, pp. 403-410.

[13] N. Sheldon, E. Girard, S. Borg, M. Claypool, E. Agu, "The Effect of Latency on User Performance in Warcraft III", in *Proc. of NetGames 2003*, May 2003. pp. 3-14.

[14] D. L. Mills, "Internet Time Synchronization: the Network Time Protocol", *IEEE Transactions on Communications*, vol. 39, no. 10, 1991, pp. 1482-1493.

[15] P. Ramanathan, K. G. Shin, R. W. Butler, "Fault Tolerant Clock Synchronization in Distributed Systems", *IEEE Computer*, vol. 23, no. 10, 1990, pp. 33-42.

[16] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, 1993, pp.397-413.

[17] J. Farber, "Network Game Traffic Modelling", in *Proc. of NetGames 2002*, Braunschweig, Germany, 2000,pp.53-57.