

Influence of Online Games Traffic Multiplexing and Router Buffer on Subjective Quality

Jose Saldana, Julián Fernández-Navajas, José Ruiz-Mas, Eduardo Viruete Navarro, Luis Casadesus
 Communication Technologies Group (GTC) – Aragon Institute of Engineering Research (I3A)
 Dpt. IEC. Ada Byron Building. CPS Univ. Zaragoza. 50018 Zaragoza, Spain
 e-mail: {jsaldana, navajas, jruiz, evirute, luis.casadesus}@unizar.es

Abstract—This work presents a simulation study of the influence of a multiplexing method on the parameters that define the subjective quality for online games, mainly delay, jitter and packet loss. The results for an available subjective quality estimator from the literature are also shown. Two buffer implementations, each one with two buffer sizes, are tested in order to study the mutual influences of the router buffer and multiplexing on subjective quality. The results show that small buffers are more adequate to maintain delay and jitter in acceptable levels, but they increase packet loss. Multiplexing shows a clear advantage when using a buffer which size is measured in packets. A buffer with a limit in packets per second is also tested.

Keywords— buffer size, multiplexing, online gaming, FPS, subjective quality, MOS, QoS

I. INTRODUCTION

The use of the Internet for playing online games is becoming very popular in the last years. Developers release new titles, but some old ones still maintain their popularity. Some games to be played online have also been developed for consoles, thus increasing the number of potential players. Amongst them, First Person Shooters (FPS from now) are the ones with the tightest real-time requirements. Every player controls an avatar in a virtual world, who has to accomplish a mission or kill the enemies, using a weapon that can be improved according to the score. The movements are very fast, and delays are critical. The Internet has to quickly evolve so as to satisfy the stringent delay requirements of this kind of services.

FPSs use a client-server architecture. Clients generate high rates of small UDP packets which arrive to the server. Once it has calculated the next state, it sends it to the clients. If delay is not controlled, some bad effects can appear: “shooting around the corner” [1], unfairness caused by a hit lost in the network [2], etc. Some games have implemented mechanisms in order to tackle these problems: prediction, buffering or time distortion are some of them.

We will use the term KPI (Key Performance Indicators) referring to the network parameters that define the quality of a connection. They are mainly delay, packet loss and jitter. Players are usually behind residential Internet access networks, which may have a very low speed at the uplink. If background traffic is present, the uplink can be a bottleneck between the LAN and the backbone network, and the main contribution to

the delay can be caused by the router buffer [2]. But there is another problem, highlighted in [3] and [4], related to the amount of packets per second (pps from now): some routers were designed for managing big packets generated by traditional applications like web browsing, e-mail or P2P, but they may have problems when dealing with high rates of small packets, because of processing capacity limitations. As residential accesses use mid or low-end routers, this implies some limitations for traffic delivery.

The characteristics of real-time traffic, i.e. high rates of tiny packets, make it suitable for multiplexing. This solution, which was first used for VoIP, consists of merging a number of small packets into a bigger one. But in order not to add undesired delays, this is only useful when the traffic of many users shares the same link. Some scenarios, like Internet cafés or the network of a game provider, where many players share the same access were explained in [5].

Multiplexing mitigates two problems: first, the efficiency problem, as every packet has to include a number of headers which may represent a high percentage of its size. For example, an RTP VoIP packet includes 40 bytes corresponding to IP/UDP/RTP headers, or a client-to-server packet of a FPS game includes 28 bytes of IP/UDP ones. As the length of the payload is about some tens of bytes, the efficiency is very low.

The other problem that can be mitigated by multiplexing is the router limitation in pps. Logically, if a number of packets are included into a bigger one, this parameter is reduced.

On the one hand, multiplexing has advantages like bandwidth saving, but on the other hand it also introduces some impairments, like additional delay and jitter. In the present work we will study these effects from the perspective of subjective quality, taking into account the implementation and the limitations of the router.

The paper is organized as follows: section II discusses the related works. The test methodology is presented in section III. The next section covers the tests and results. The paper ends with the conclusions.

II. RELATED WORKS

A. Multiplexing real-time flows

The IETF defined TCRTCP [6] as RFC 4170, as a combination of protocols in order to multiplex many RTP

flows. First, ECRTMP header compressing is applied, and PPPMux is used to merge the packets into one, which is sent via an L2TP tunnel.

Other proposals for multiplexing RTP flows are [7] and [8]. They are mainly designed for VoIP, since this is a very popular service with a high overhead.

In [9] our group proposed TCM, which is an adaptation of TCRTMP scheme to the traffic of FPS games. It replaces ECRTMP by other compressing algorithms, since these games do not use RTP traffic. We will use this method in the current paper, in order to study the effect of the buffer behaviour on the different KPIs.

B. Subjective quality evaluation

The E-Model [10] is a widely accepted method which has shown useful for designing and dimensioning VoIP systems. It defines some impairment factors which are finally combined in a MOS formula. The network KPIs that have an influence on the calculation are delay and packet loss. Jitter is not directly considered, as de-jitter buffers are usually implemented in VoIP applications. They add a delay and also discard the packets that arrive too late to be correctly played.

Regarding online games, in [11] the authors studied the different delay upper bounds appeared in the literature, which are between 150 and 250 ms. In the same study, two FPS games were tested, obtaining a similar behaviour regarding delay, but packet loss was very different: while Halo stopped working when packet loss was around 4%, the users of Quake III did not experience a quality degradation even with a packet loss of about 35%. This means that this game has a very good loss concealment system. The effect of delay and jitter was also studied in [12] for two FPS, a sports and an RTS game. Its effect is very different depending on the considered game.

In [13] a MOS formula for Quake IV (G-Model) was developed, based on delay and jitter, which was measured as the standard deviation of the delay. Packet loss was not considered taking into account the very good method the game has to hide this to the players. Ref. [2] presented a formula with weighting factors for delay and jitter, which will be different for each game, and also considered packet loss. But, since this work was carried out inside a company, these parameters are not publicly available.

Finally, some works have studied the relationship between subjective quality and multiplexing, mainly for VoIP. Ref. [14] presented a multiplexing method based on E-Model for reducing overhead when using IPsec. Also in [15] our group compared the conversation quality [10] obtained when using native RTP, TCRTMP and the proposal of [7]. Bandwidth saving was shown as an important factor, but another conclusion was that packet size has to be taken into account, as some buffers penalize big packets, i.e. packet loss probability increases with packet size.

C. The influence of the router buffer

Since the work of Appenzeller et al. [16] argued the traditional rule of the thumb of using bandwidth-delay product

to calculate the size of the router buffer, there have appeared many papers related to this topic [17]. The problem has been mainly tackled for backbone routers managing a number of TCP flows. A tendency to reduce the size of the buffer can be observed, and some works [18] propose the use of *tiny* buffers (some tens of kB). In the current work we will not consider backbone routers, but mid or even low-end ones, as in our scenario we will find commercial access routers. Nevertheless, the idea of reducing the size of the buffer is also interesting for real-time UDP flows [19].

This topic is related to the two other ones that have been issued in this Section, i.e. the behaviour of the router buffer has a big influence on experienced quality, and of course multiplexing can strongly modify the traffic. Our group has studied the relationship between these three problems in some previous works: in [15] we compared the behaviour of VoIP multiplexing when using different buffers. In [5], the influence of game multiplexing was also studied, in terms of delay and packet loss, for a high capacity buffer and also for a time-limited one. The buffer size was found to be a very important parameter which has to be taken into account in order to select a correct value for the multiplexing period.

The current work tries to more deeply study these relationships, measuring not only delay and packet loss, but also jitter and subjective quality estimators.

III. TEST METHODOLOGY

We have used a Matlab simulation scenario (Fig. 1), where the multiplexed traffic of a number of users shares the same access link. Traces of different real parties have been properly merged in order to obtain the traffic of a number of users, and then they are multiplexed using TCM [9]. Finally, the packets are sent via a router which also receives Poisson background traffic with the next distribution [20]: 50% of the packets are of 40 bytes; 10% of 576 bytes, and 40% of 1500 bytes.

The total RTT (Round Trip Time) is measured as the sum of $delay_{mux}$, which includes the retention time plus the processing time in the multiplexer; $delay_{router}$ is the queuing delay at the router buffer which, as we have seen in the introduction, can be significant if background traffic is present; $delay_{network}$ includes the propagation time in the network, plus the processing times in the demultiplexer and game server.

Since the system only multiplexes client-to-server traffic, there is no demultiplexing delay when the traffic arrives to the client. The delay of the router is only considered for the uplink,

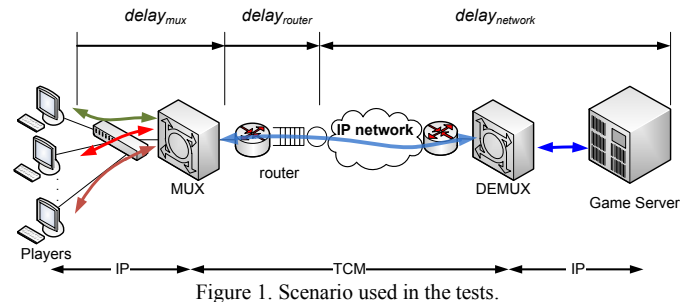


Figure 1. Scenario used in the tests.

as in the downlink packets pass from a slower to a faster network, so queuing delay will be negligible.

A period PE is defined in the multiplexer [9] in order to periodically send a multiplexed packet including all the native game packets which have arrived. As can be seen in Fig. 2 this will add a delay, and also jitter, since the added delay will be different depending on the moment in which the packet arrives: if it arrives at the beginning of a period, it will be delayed much more than if it arrives at the end.

Logically, if a big value is set for the period, bandwidth saving will increase but, on the other hand, retention time will grow. This fact implies a series of tradeoffs: if PE is increased, bandwidth is reduced, so the buffer will discard less packets. But on the other hand, retention time will grow, and so will do the jitter. But there is still another factor which has to be taken into account: packet size. Multiplexed packets are bigger than native ones and this will have an influence depending on the implementation of the router buffer. The reduction in terms of pps can also have an influence in some cases.

Multiplexing delay will be half the period in average, considering that the packet arrivals will be uniformly distributed during the period. In order to calculate the standard deviation, we have to take into account that the variance of a uniformly distributed variable in an interval (a,b) is $(b-a)^2/12$. As the added delay varies from 0 to PE , the standard deviation of the delay added at the multiplexer will be:

$$stdev_{mux} = PE / \sqrt{12} \quad (1)$$

Root-mean-square can be used to calculate the addition of standard deviations of uncorrelated variables. In order to obtain the total standard deviation of the delay, we have to take into account that $delay_{mux}$ and $delay_{router}$ are correlated, since the variation of PE will modify the total traffic offered to the router, and consequently the delay added by the buffer. So the standard deviation has been calculated since a packet arrives to the multiplexer until it leaves the router buffer.

$$stdev_{mux+router} = \sqrt{stdev_{mux}^2 + stdev_{router}^2 + cov_{m+r}} \quad (2)$$

We have observed that $stdev_{mux+router}$ is smaller than the root-mean-square of $stdev_{mux}$ and $stdev_{router}$ so this implies that the covariance (cov_{m+r}) of $delay_{mux}$ and $delay_{router}$ is negative. This happens when higher than average values of a variable correspond to lower than average values of the other variable. In our case, this could be expected as, when we use higher values of PE , the variance at the multiplexer grows, but the total traffic offered to the router is reduced, so the variance of queuing delay will be reduced. We have considered that network delay is independent of the two other ones.

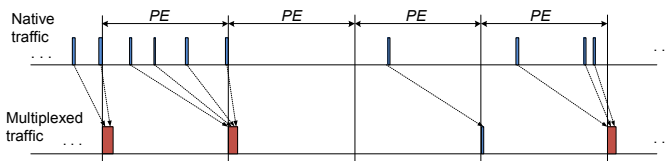


Figure 2. Multiplexing method.

Propagation delay with an average of 30 ms and a variance of 5 has been added. So the total jitter has been calculated as:

$$stdev_{total} = \sqrt{stdev_{mux+router}^2 + stdev_{network}^2} \quad (3)$$

IV. TESTS AND RESULTS

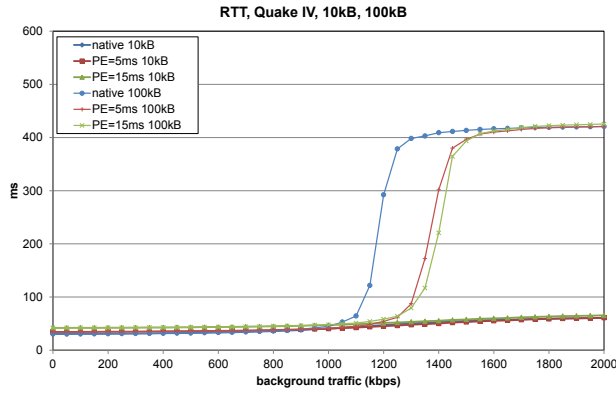
Two buffer implementations have been used: one defines its size in bytes (*byte-sized*) and the other in packets (*packet-sized*). In fact, some manufacturers measure the buffer size in bytes and others do in packets. For example, in [21] the routers of two manufacturers are compared, and one of them gives this information in packets, while the other gives it in milliseconds, which is equivalent to bytes, as they are related by the line speed. We will use two sizes (*small* and *big*) for each implementation: 10 and 100 kB for *byte-sized*, and 16 and 166 packets for *packet-sized*. The sizes are roughly equivalent, considering average packet size of 600 bytes.

In this work we will use Quake IV, since a MOS formula has been developed for it [13]. The number of flows used in the tests is 20. Each user generates 40.7 kbps (64 pps, 79.5 bytes avg.), so the total game traffic is 814 kbps. Measurements have been carried out with native traffic traces, and also with multiplexed ones setting PE to 5 and 15 ms.

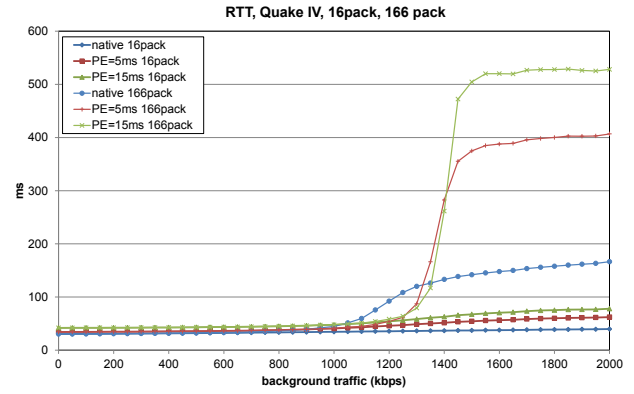
Fig. 3 shows RTT, packet loss, jitter and MOS for each buffer. First, we can observe that, when the link capacity is exceeded, the delay grows (Fig. 3.1). For *small* buffers, it does not grow to unacceptable values, but for *big* ones we observe that the delay grows dramatically, and also a peak appears in the jitter graphs (Fig. 3.2). We observe that this growth appears later when multiplexing is used. Bandwidth saving of multiplexing is about 200 kbps. This saving could become more significant if a narrower buffer was used. The jitter goes down again after the peak because the buffer is always full, so buffering delay will have very little variations. Fig. 3.2.2. shows a smaller peak for the native traffic jitter than Fig. 3.2.1., because in this case the average packet size is smaller, as a big number of native packets are present, so queuing delay will also be smaller.

Regarding packet loss (Fig. 3.3), we can first observe that *small* buffers start losing packets before, while *big* ones start exactly when the bandwidth limit is reached. Other effect is related to packet size: if the buffer is *byte-sized*, then big packets have a bigger probability of being discarded. This is the reason why multiplexing with high values of PE obtains worse results. If the buffer is *packet-sized*, then all the packets have the same probability of being discarded, so packet loss is increased with respect to *byte-sized* buffers. In this case, packet loss is significantly reduced when multiplexing. This advantage is caused by the pps reduction.

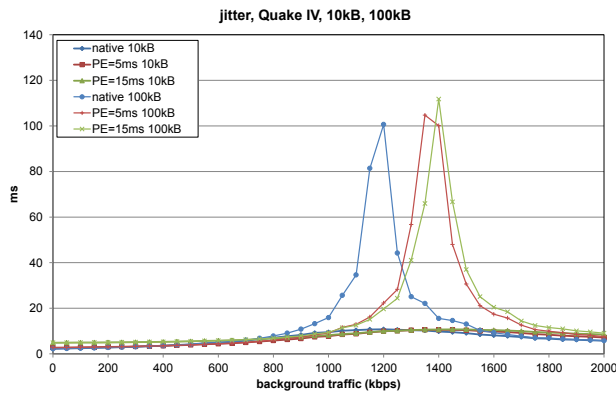
Although the model used for the MOS does not consider packet loss, it does have an effect on perceived quality, which may be mitigated by concealment methods [11]. As a conclusion, we can say that *small* buffers are more adequate to maintain delay and jitter in acceptable levels, but they increase packet loss.



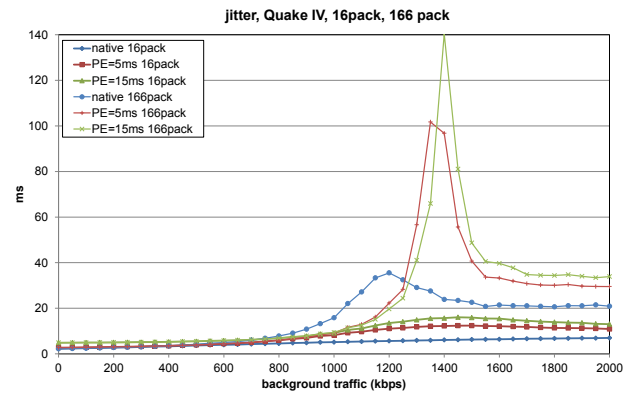
(1.1)



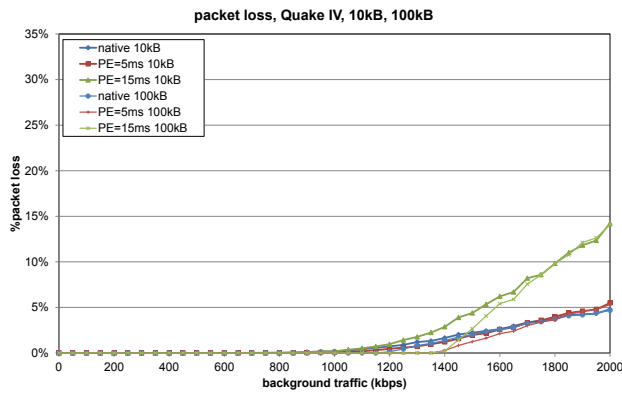
(1.2)



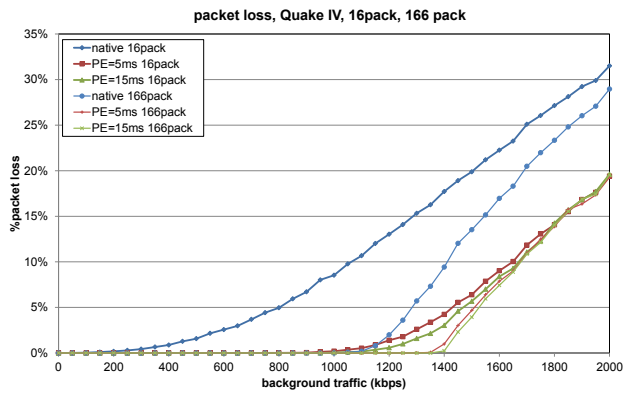
(2.1)



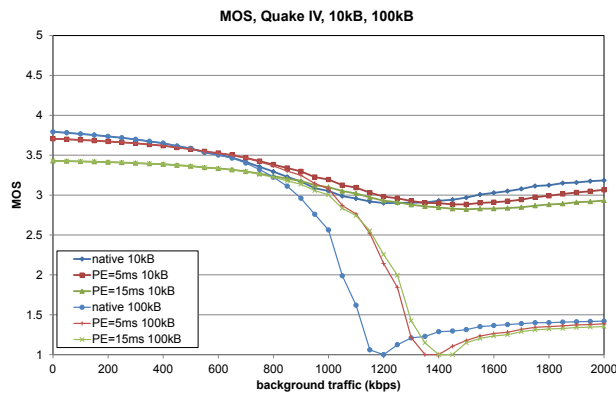
(2.2)



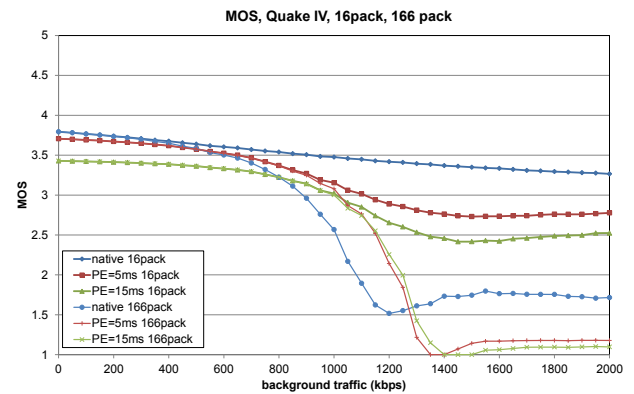
(3.1)



(3.2)



(4.1)



(4.2)

Figure 3. RTT, jitter (delay standard deviation), packet loss and MOS for 10 kB, 100 kB, 16 packets and 166 packets buffers

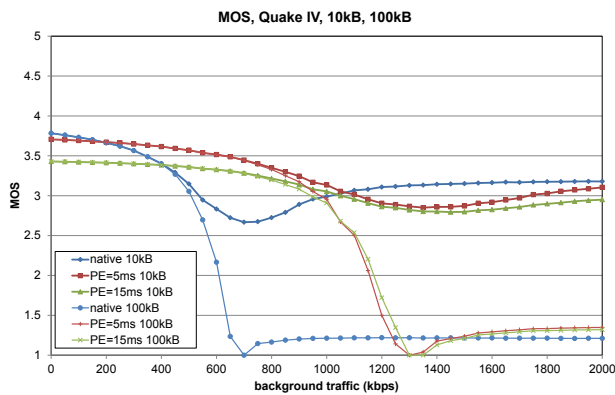


Figure 4. MOS for byte-sized buffers with a limitation of 2000 pps.

Fig. 3.4 shows the MOS. The used formula [13] is only based on delay and jitter, since this game has a very good loss concealment method. Normally acceptable MOS values are considered above 3.5 or even 3. For *small* buffers, the graphs first go down because of the jitter peak, and after that they grow a little. For *10 kB* buffer the difference is small, whereas for *16 packets* buffer multiplexing is worse than native traffic. When using *big* buffers, the graphs go down later for multiplexed traffic, so it can support a bigger amount of background traffic, because of bandwidth saving.

Finally, Fig. 4 illustrates another limitation of the router, i.e., the number of pps it can manage. If compared with Fig. 3.4.1, we see that the behaviour becomes clearly worse if the router has a limitation of 2000 pps. Although the maximum offered amount of pps is 1652, this limitation has a very bad effect on MOS. This illustrates that, in addition to the bandwidth limit of our connection, the limitation of pps has to be considered.

V. CONCLUSIONS

This work has studied the influence of a game traffic multiplexing method on the parameters that define the subjective quality, mainly delay, jitter and packet loss. The results for an available subjective quality estimator from the literature have also been shown. Two buffer implementations, each one with two buffer sizes, have been tested in order to study the mutual influences of the router buffer, multiplexing and subjective quality.

The results show that small buffers are more adequate to maintain delay and jitter in acceptable levels, but they increase packet loss. Buffers which size is measured in packets increase packet loss, as all packets have the same probability of being discarded, and in this case multiplexing shows a clear advantage because of the reduction in pps. A buffer with a limit in packets per second has also been tested, and in this case multiplexing has shown a clear advantage.

ACKNOWLEDGMENT

This work has been partially financed by CPUFLIPI Project (MICINN TIN2010-17298), MBACToIP Project, of Aragon I+D Agency and Ibercaja Obra Social, Project of Catedra Telefonica, Univ. of Zaragoza.

REFERENCES

- [1] M. Oliveira, T. Henderson. "What online gamers really think of the Internet?". In Proc. 2nd workshop on Network and system support for games (NetGames '03). ACM, New York, NY, USA, 185-193, 2003.
- [2] Ubicom White Paper, "OPScore, or Online Playability Score: A Metric for Playability of Online Games with Network Impairments", 2005, <http://www.kevingee.biz/wp-content/uploads/2011/04/IP3K-DWP-OPSCORE-10.pdf>
- [3] J. Yu, I. Al-Ajarmeh. "Call Admission Control and Traffic Engineering of VoIP", In Proc. Second International Conference on Digital Telecommunications, IEEE ICDT 2007.
- [4] W. Feng, F. Chang, W. Feng, J. Walpole: "A Traffic Characterization of Popular On-Line Games". IEEE/ACM Trans. Netw., pp. 488-500, 2005.
- [5] J. Saldana, J. Fernández-Navajas, J. Ruiz-Mas, J. I. Aznar, E. Viruete, L. Casadesus, "Influence of the Router Buffer on Online Games Traffic Multiplexing," Proc. Int. Symposium on Performance Evaluation of Computer and Telec. Systems SPECTS 2011, The Hague, Netherlands, June 2011.
- [6] B. Thompson, T. Koren, D. Wing. RFC 4170: "Tunneling Multiplexed Compressed RTP (TCRTP)", November 2005.
- [7] H.P. Sze, S. C. Liew, J.Y.B. Lee, D.C.S.Yip. "A Multiplexing Scheme for H.323 Voice-Over-IP Applications", IEEE J. Select. Areas Commun., Vol. 20, pp. 1360-1368, September. 2002.
- [8] A. Trad, H. Afifi, "Adaptive Multiplexing Scheme for Voice Flow Transmission Across Best-Effort IP Networks", INRIA Research Report 4929, Sep. 2003.
- [9] J. Saldana, J. Murillo, J. Fernández-Navajas, J. Ruiz-Mas, J. I. Aznar, E. Viruete, "Bandwidth Efficiency Improvement for Online Games by the use of Tunneling, Compressing and Multiplexing Techniques," Proc. Int. Symposium on Performance Evaluation of Computer and Telec. Systems SPECTS 2011, The Hague, Netherlands, June 2011.
- [10] "The E-model, a computational model for use in transmission planning", ITU-T Recommendation G.107, 2005.
- [11] S. Zander, G. Armitage, "Empirically Measuring the QoS Sensitivity of Interactive Online Game Players". In Proc. Australian Telecommunications Networks & Applications Conference (ATNAC 2004), Sydney, Australia, Dec. 2004.
- [12] M. Dick, O. Wellnitz, L. Wolf. "Analysis of factors affecting players' performance and perception in multiplayer games". In Proc. 4th ACM SIGCOMM workshop on Network and system support for games (NetGames '05). ACM, New York, NY, USA, 1-7, 2005.
- [13] A. F. Wattimena, R. E. Kooij, J. M. van Vugt, O. K. Ahmed, "Predicting the perceived quality of a first person shooter: the Quake IV G-model". Proc. 5th ACM SIGCOMM Workshop on Network and system support for games (NetGames'06), ACM, New York, USA, 2006.
- [14] R. M. Pereira, L.M. Tarouco: "Adaptive Multiplexing Based on E-model for Reducing Network Overhead in Voice over IP Security Ensuring Conversation Quality", in Proc. Fourth international Conference on Digital Telecommunications, Washington, DC, 53-58, July 2009.
- [15] J. Saldana, J. Murillo, J. Fernández-Navajas, J. Ruiz-Mas, E. Viruete, J. I. Aznar, "Evaluation of Multiplexing and Buffer Policies Influence on VoIP Conversation Quality", Proc. CCNC 2011-3rd IEEE Workshop on Digital Entertainment, Networked Virtual Environments and Creative Technology (DENVECT), pp 1147-1151, Las Vegas. Jan 2011.
- [16] G. Appenzeller, I. Keslassy, and N. McKeown. "Sizing router buffers", In SIGCOMM '04, pages 281-292, New York, USA, 2004. ACM Press.
- [17] A. Vishwanath, V. Sivaraman, and M. Thottan. "Perspectives on router buffer sizing: recent results and open problems," SIGCOMM Comput. Commun. Rev. 39, 2, 34-39, March 2009.
- [18] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, T. Roughgarden. "Part III: routers with very small buffers," SIGCOMM Comput. Commun. Rev. 35, 3, pp 83-90, July 2005.
- [19] A. Vishwanath and V. Sivaraman. "Routers with Very Small Buffers: Anomalous Loss Performance for Mixed Real-Time and TCP Traffic," In IEEE IWQoS, Netherlands, 2008.
- [20] Cooperative Association for Internet Data Analysis "NASA Ames Internet Exchange Packet Length Distributions".
- [21] J. Sommers, P. Barford, A. Greenberg, W. Willinger, "An SLA perspective on the router buffer sizing problem," SIGMETRICS Perform. Eval. Rev. 35, 4, pp 40-51, March 2008.