The Effect of Multiplexing Delay on MMORPG TCP Traffic Flows

Jose Saldana
Communication Technologies Group (GTC), Aragon Institute of Engineering Research (I3A)
EINA, University of Zaragoza
Zaragoza, Spain
jsaldana@unizar.es

***Abstract*— Traffic optimization techniques, based on header compression, multiplexing and tunneling a number of concurrent flows, have been proposed in order to save bandwidth and to reduce the amount of packets per second in certain network segments. These techniques have been standardized for VoIP flows, but they can also be applied to the flows of online games. As a counterpart of the savings, the adding of a multiplexing delay is required in order to get a number of packets to be multiplexed together. The effect of this delay has already been explored for UDP-based games, showing that subjective quality can be maintained while using traffic optimization. In contrast, the present paper explores the issues arising when these optimization techniques are applied to TCP-based games. The interaction of the additional multiplexing delay and the TCP mechanisms is explored, and results are obtained in terms of round trip time increase and traffic overhead modification.**

***Index Terms*— online games; traffic optimization; multiplexing; application-limited TCP**

I. Introduction

Many emerging network services are based on small packets. The real-time requirements and the interactivity of these services make it necessary to divide the information into small pieces, which are sent at a high rate, as it happens with VoIP or online games. Furthermore, some non-interactive services also send small packets, as it happens with instant messaging, Machine to Machine (M2M) communications or the signaling of certain services.

These "small-packet services", as they are frequently called, present a very low payload-to-header ratio, since many payloads typically contain a few tens of bytes. As a consequence of the increasing use of small-packet services, network operators are witnessing a change on the packet size distribution of the traffic mix they have to manage, which also implies a reduction of the overall network efficiency.

As an additional problem, the global traffic of some of these services may present a degree of unpredictability, with the consequence of traffic surges appearing at certain moments (e.g., the release of a new game or new content of an existing one) or places (e.g., instant messaging during a sports event, a concert, etc.). This phenomenon is also known as "flash crowd".

In order to increase this low network efficiency, be it caused by a permanent or by an instantaneous traffic rush, traffic optimization techniques have been proposed for small-packet flows [1].

VoIP was the first small-packet service becoming popular, and a technique for optimizing the traffic when a number of flows share a common path was standardized in 2005 by the IETF [2]. More recently, the use of techniques based on Tunneling, Compressing and Multiplexing (TCM) has been proposed not only for VoIP, but also for other traffic flows, such as those generated by online games [3]. In [4], the expected savings for eight different UDP-based First Person Shooter (FPS) [5] games were calculated, showing that a bandwidth reduction of 30 or 35% can be obtained for client-to-server flows. In [6] the savings for three TCP-based Massively Multiplayer Online Role Playing Games (MMORPGs) were calculated, and the predicted bandwidth reduction was even higher (up to 54%). The cause is twofold: first, TCP headers are bigger than UDP ones; and second, the high rates of TCP ACKs, makes the compression rate be higher, since they do not include a payload.

These techniques provide a tradeoff, since they permit to reduce the bandwidth and the amount of packets per second, at the cost of the processing power required by header compression techniques. Another counterpart is the need of an additional "multiplexing" delay, in order to get the packets which will be included in the same bundle. A small processing delay will also appear. Thus, the effect of optimization techniques has to be quantified in order to establish the conditions in which subjective quality can be maintained.

If the application generates packets with a fixed rate (as it usually happens with VoIP), when a number of concurrent flows are multiplexed together, it is enough if the optimizer gets a packet from each flow, compresses its header and puts it into the multiplexed bundle (Fig. 1.a). Thus, the additional delay will present in this case an upper bound given by the Inter-Packet Time (IPT). However, online games do not usually send packets with a fixed cadence [5], [7], so a different mechanism has to be used in order to select the native packets that will travel into a multiplexed one. In this work we will use a multiplexing method capable of setting an upper bound for the additional multiplexing delay [4]: a period (*PE*) is defined, and all the packets arrived during the interval will travel within the same multiplexed bundle (Fig. 1 b).

The contribution of the present work is the study of the effect of traffic optimization, when applied to TCP-based MMORPG flows. In this case, the question is not as simple as

978-1-4799-2355-7/14/$31.00 ©2014IEEE

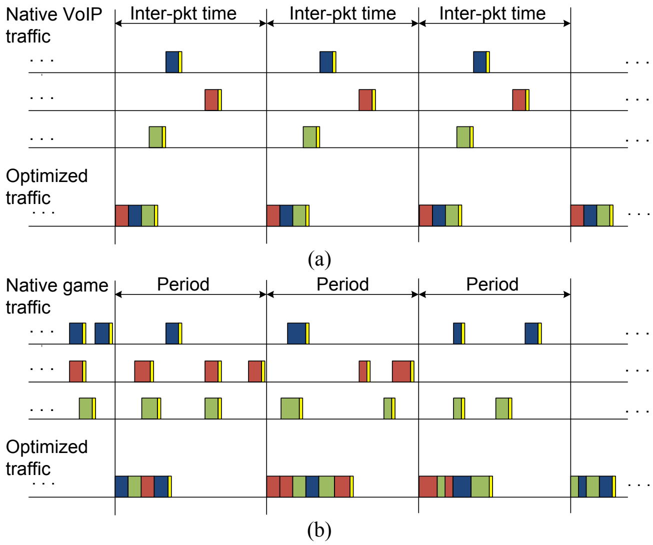


Fig. 1. Multiplexing scheme of a) three VoIP flows; b) three online game flows

in UDP flows, which are unidirectional and do not retransmit lost packets. When TCP is used, new problems arise, since the additional delay may modify the evolution of the parameters that govern the dynamics of the transport protocol, as e.g., Round Trip Time (RTT). The traffic of a popular MMORPG, namely *World of Warcraft* (WoW), will be used for that aim.

The rest of the paper is organized as follows: the related work is summarized in the next section; the traffic optimization method is described, and the maximum recommendable multiplexing period is estimated in section III. The simulation setup is explained in IV, and the results are presented and commented in V. The paper ends with the conclusions.

II. Related Work

In order to reduce the bandwidth of small-packet flows, traffic optimization techniques rely on header compression, based on the avoidance of the header fields that are the same for all the packets of a flow [8]. However, a packet with a compressed header cannot travel end-to-end unless tunneled, so header compression is combined with the multiplexing of a number of packets, which share a common tunneling header, thus obtaining significant savings [1], [2], [9].

As a counterpart of traffic optimization, new delay and jitter may appear. In [10] a mechanism using a period was proposed and compared, in terms of delay and jitter, with another one using a timeout. In [11] an adaptive multiplexing method was proposed for VoIP, able to maintain the voice quality in acceptable limits, according to the E-Model. The effect of traffic optimization on the subjective quality of UDP-based FPS games was explored in [12].

TCP is commonly used in certain game genres, because it avoids the loss of information, and also because it is easier to code. However, the fact of using TCP for an online game, considered as a (soft) real-time service, can be surprising in a first approach [13]. In general, the objective of TCP is to obtain the maximum throughput, while maintaining fairness and avoiding congestion. This is convenient for transferring a certain amount of data, so many TCP mechanisms assume that the throughput is limited by the network (network-limited traffic). However, many of these mechanisms lose their meaning when used by a game, i.e., an application that generates a limited bandwidth (application-limited traffic), being the interactivity the most important issue. In fact, some TCP mechanisms (delayed ACK, Nagle algorithm) may even deteriorate the player's experience [14].

The problem of the unfairness of TCP, (i.e. flows with lower RTTs get more throughput) has been largely studied [15], and different solutions and TCP improvements have been proposed [16]. This phenomenon has been mainly studied for network-limited traffic, in terms of the throughput obtained by each flow. However, when TCP is used for a real-time service, maximizing the throughput is not the main objective, since data are generated on the fly, according to the game dynamics and to the players' actions. Thus, when traffic optimization is employed, it may add a new delay and jitter to the flows, which may share the network with non-optimized ones. The agents that perform the optimization (e.g., network operators, service providers) may be interested on limiting the potential unfairness between optimized and non-optimized traffic flows.

All in all, the present work puts together these issues, studying the effect of traffic optimization techniques on the behaviour of TCP-based online games. We will specifically focus on the competition between optimized and non-optimized flows.

III. Description of the Traffic Optimization Method

Traffic optimization can be useful in certain network segments, when a number of flows share the same path. It can be permanent or dynamic, being only activated whenever a traffic rush occurs. The optimization does not affect the whole network path of a traffic flow, but only a segment shared by a number of flows of the same nature. After the segment in which packets are optimized, they are rebuilt exactly as they were when they arrived to the TCM ingress (Fig. 2).

Some examples of real scenarios where a number of flows of the same nature can be found, and therefore traffic optimization benefits can be obtained are:

- The aggregation network of an operator.
- The tunnels between different offices of a company where a VPN is established, which may include concurrent VoIP flows between the same pair of offices.
- All the small-packet flows generated in an Internet Café can be optimized in order to save bandwidth in the access link.
- In some wireless or satellite connections, multiplexing a number of flows before transmission can simultaneously reduce the required bandwidth and the amount of packets per second generated.

In all these cases, there is a common scheme, similar to the one presented in Fig. 2: packets can be aggregated in a certain part of the network path, but the ingress and egress of the optimization is never the endpoint itself, since a single host is not expected to generate a high number of small-packet flows

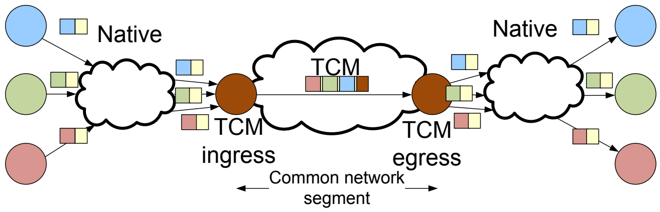


Fig. 2. General scheme of TCM optimization

with similar characteristics. As a consequence of this, the multiplexing delay will be seen by the two endpoints as a delay and a jitter added to those already present in the network. If we look at Fig. 1, it is easy to deduce that this multiplexing delay is sawtooth-shaped, since packets arriving at the beginning of a period will experience a delay equal to *PE*, and packets arriving at the end will experience a very low delay (Fig. 3).

Thus, the effect of multiplexing can be modelled as the sum of a fixed delay with the value of half the period:

$$delay_{mux} = PE\,/\,2\,, \tag{1}$$

and an additional variable delay, uniformly distributed in the interval ($-PE/2$, $PE/2$), which standard deviation, as obtained in [12], has this value:

$$stdev_{mux} = PE\,/\,\sqrt{12} \tag{2}$$

In order to have a first idea of the effect of multiplexing on subjective quality, we have used a MOS (Mean Opinion Score) model for WoW, deployed in [17], to build Fig. 4. This model obtains a MOS (i.e. an estimation of the subjective quality) as a function of network delay and jitter. So we have calculated the sum of $delay_{mux}$ plus the network delay, and we have also added $stdev_{mux}$ to the standard deviation of the network, using root-mean-square for the obtaining of the global jitter. Thus, the MOS is obtained for different values of network latency, adding the effect of the sawtooth-shaped multiplexing delay. The considered standard deviation of the network delay is 10 ms.

If we select the value of 3.5 as the acceptable MOS, we can see that when network delay is below 40 ms, multiplexing can be applied even with a period of 100 ms. However, if the network latency is 60, 80 or 100 ms, we must use a low multiplexing period if we want to maintain the subjective quality into acceptable levels. In [6] it was shown that significant bandwidth savings can be obtained even if a period of 50 ms is used, since the savings present an asymptotic behaviour and higher periods only provide marginal bandwidth

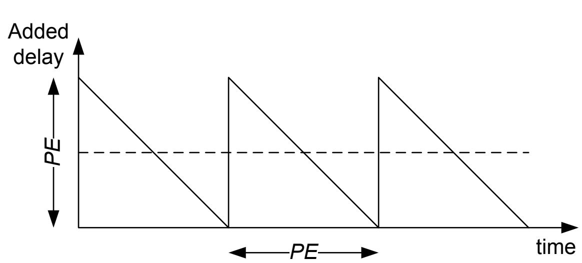


Fig. 3. Sawtooth-shaped delay caused by multiplexing

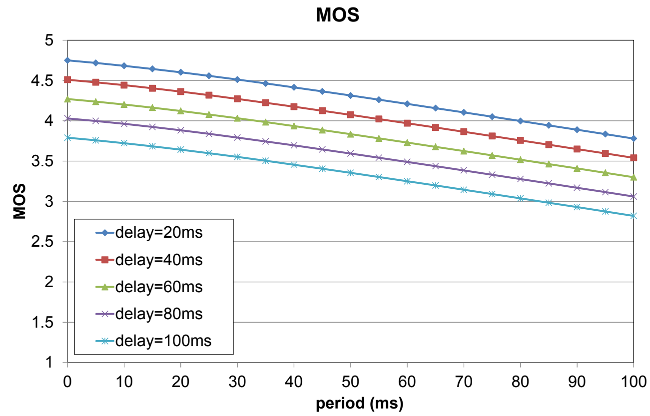


Fig. 4. MOS as a function of network delay and multiplexing period

reductions. All in all, this graph gives us an idea of the values of the period that can be acceptable, so we will limit the period to 50 ms in the subsequent tests.

IV. Scenarios of Interest: Modelling and Simulations Setup

The aim of the tests presented in this section is to study the effect of multiplexing delay on the dynamics of TCP-based MMORPG flows. So we will first explain the traffic model used, and we will then present the simulation scenario.

A. MMORPG TCP Traffic Generation

Different statistic traffic models have been developed for some popular MMORPGs. Some of them simply propose a statistical model for the client traffic and other for the server [18]. However, advanced models, looking for more accuracy, define different categories of player's activities, creating a model for each of them [7], [19]. These models reflect the wide range of activities that can be deployed in an MMORPG, some of them more interactive (e.g., *Player vs Player Combat*) and some of them more relaxed (e.g., *Trading*).

In this study we use the traffic model for WoW proposed in [7]. We have selected it because it includes five different activities (*Trading, Questing, Dungeons, Raiding, Player vs Player Combat*), and this will allow us to deploy some tests comparing the effect of multiplexing delay, depending on the interactivity of each of them.

We will only multiplex client-to-server traffic. The reason is twofold: first, client-to-server connections generate smaller packets, which are more suitable to be optimized [6], [18]; second, in many of the scenarios of interest, the uplink is more restrictive than the downlink (e.g., DSL).

B. Simulation Scenario

We have built a dumbbell NS2 simulation scenario (Fig. 5), able to add a sawtooth-shaped delay to some flows, in order to compare the results with the case where no multiplexing delay is present. The characteristics of the scenario are:

- A number of MMORPG sessions are established between A (clients) and A' (servers). The same happens with B and B'.
- Nodes C-C' are used to create background traffic.

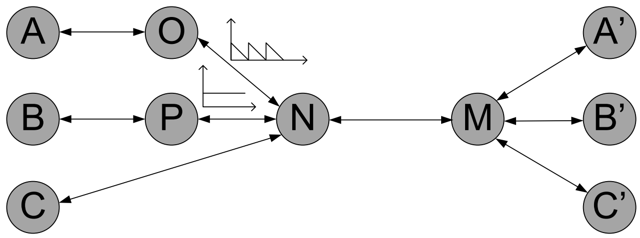


Fig. 5. NS2 simulation scenario

- The One Way Delay (OWD) of the bottleneck is set to 20 ms. Links A-O, O-N, B-P and P-N have an OWD of 5 ms. The rest of the links have a latency of 10 ms.
- The connection O-N is able to add sawtooth multiplexing delay, on the client-to-server direction.
- A fixed delay (*PE*/2) is added at the P-N link
- The default bandwidth of the links is 10 Mbps. The bandwidth of the bottleneck uplink is 1 Mbps.
- NS2 *FullTCP* is used for the WoW flows, since it is bidirectional, in order to imitate the real traffic of the game, which uses piggybacking, sending the ACKs in data packets [18]. The NS2 parameter *segsperack_* (segments received before generating an ACK) is set to 0 in order to emulate the behaviour of the game, which sets to 1 the *PUSH* bit, so as to ask TCP to send the packet as soon as possible [18]. TCP SACK is used for background TCP connections.
- By default we will use *Questing* traffic, since it is a very common activity in MMORPGs.

The scheme is similar to the one presented in Fig. 2: multiplexing delay only affects packets during a certain part of the network path, but the ingress (O) and egress (N) of the optimization tunnel are never located in the endpoints.

V. Tests and Results

In this section we will use the proposed scheme in order to explore the effect of multiplexing delay on TCP dynamics. The main questions we are trying to answer are: how does multiplexing delay impair the game traffic?; to what extent is fairness between multiplexed and non-multiplexed flows granted?. So we will compare flows affected by a sawtooth-shaped delay, against non-delayed or uniformly delayed flows, sharing the same bottleneck.

First of all, we have to find some figures of merit able to express our results. When studying TCP, it is frequent to use the throughput obtained by each flow as the most interesting magnitude, taking into account that the main aim of TCP's design is to obtain the maximum possible throughput, while maintaining fairness and avoiding network congestion. As an example, when "RTT unfairness" between different flows is measured [16], the results are mainly reported in terms of throughput.

However, we should remember that in this case we are not studying network-limited flows, in which a certain amount of data have to be transmitted as fast as possible. In contrast, application-limited flows transmit the information while it is generated by the player, so the critical magnitude here is the delay. We cannot forget that we are using TCP for a (soft) real-time service in which interactivity matters, and it may have an influence on the result of the game (players usually talk about "ping" as the RTT). In this sense, the amount of retransmitted packets will also be important, since a retransmission causes a significant delay to the game traffic.

As a consequence, we will express the results in terms of the next magnitudes:

1) *The RTT parameters estimated by TCP.* In order to govern TCP dynamics (e.g., retransmissions), RTT samples are calculated and updated frequently, according to the network conditions. They are used to compute two different parameters, namely *smoothed RTT* and *RTT variation,* which are subsequently used so as to obtain the value of *Retransmission TimeOut* (RTO) [20]. If the timeout expires, the packet is retransmitted. As a consequence, we will present some of the results in terms of these RTT parameters, taking into account that the multiplexing delay may affect them.

2) *The retransmission overhead, i.e., the relationship between the number of bytes generated by the application, and the number of bytes sent by TCP.* If some packets are lost, then TCP will retransmit them. In that case, the estimated RTT may have different values if an additional multiplexing delay is present. We will define the Retransmission Overhead (*RO*) as the relationship between the amount of bytes sent by TCP retransmissions (headers and ACKs are not counted) and the amount of bytes generated by the application:

$$RO = \frac{bytes_{TCP\_retrans}}{bytes_{application}} = \frac{bytes_{TCP}}{bytes_{application}} - 1 \qquad (3)$$

As an example, if 1,000 bytes are generated by the application, and 1,100 bytes are finally sent by TCP, then the value of *RO* is 0.1 (i.e., 10% overhead).

A. MMORPG vs FTP Traffic

In these tests we will observe the effect of the sawtooth-shaped multiplexing delay when MMORPG flows compete between them, in the presence of FTP background traffic. These flows can coexist in a common network segment included in their paths. We establish 50 game sessions between A and A', which experience a multiplexing delay with a period *PE*. 50 additional game sessions are established between B and B', and a fixed delay of *PE*/2 is added to them; finally, one background FTP connection, which tries to get as much throughput as possible, is established from C to C' (uplink). The simulation time is 200 sec. The values of RTT parameters are calculated this way: the average value of each parameter is first calculated for each flow, between sec. 100 and 200. Then, the average value of the results obtained for each flow is presented. The 95% confidence intervals are also obtained.

In Fig. 6 a, we see that the difference between multiplexed and non-multiplexed flows cannot be appreciated in terms of *smoothed RTT*, since the average delay is *PE*/2 in both cases. However, an increase can be observed in the *RTT variation* of the optimized flows (Fig. 6 b), which means that TCP detects a higher value of the jitter in the flows affected by the multiplexing delay.

This increase in the *RTT variation* is translated into a higher *Retransmission Overhead* (Fig. 6 c) for the multiplexed flows. As a first result, we see that for the lowest values of *PE,* the overhead is not very significant, and the differences are really small. However, as the multiplexing period grows, the unfairness between multiplexed and non-multiplexed flows gets more significant. This yields a first conclusion: if the value of *PE* is 40 ms or higher, the unfairness may be noticed by the players.

B. *MMORPG vs UDP Traffic*

In this subsection we will study the effect of the multiplexing delay, when multiplexed and non-multiplexed game flows compete, in the presence of Constant Bit Rate (CBR) UDP traffic. For that aim, 50 MMORPG flows are established between A and A', adding a multiplexing delay; 50 delayed flows affected by a delay of *PE*/2 are set between B and B'; different amounts of UDP traffic are sent from C to C', following this packet size distribution: 50% of the packets are of 40 bytes; 10% are of 576 bytes; and the remaining 40% are 1,500 bytes packets [21].

We have set *PE* to 30 ms for all the tests, and we will vary the amount of background traffic. It should be remarked that the aggregate bandwidth of the MMORPG flows is roughly 300 kpbs, so congestion may appear when background traffic approaches 700 kbps (the bandwidth in the uplink is 1 Mbps).

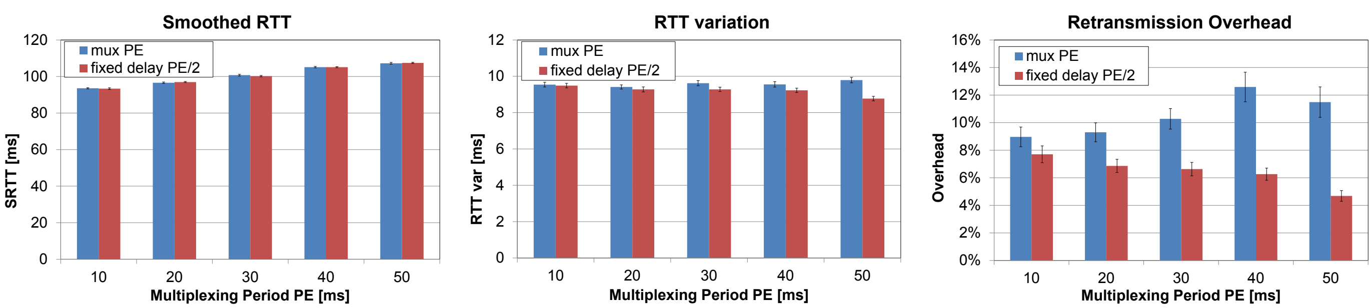
The results are shown in Fig. 7. In contrast to what happened with FTP, in this case the background traffic does not reduce its rate as a consequence of congestion. Although the value of the *smoothed RTT* (Fig. 7a) is not very different from the results obtained with FTP (compare with the third column set of Fig. 6a), the difference between multiplexed and non-multiplexed flows is stressed in terms of *RTT variation* (see the third column set of Fig. 6b, and all the columns of Fig. 7b). In normal conditions, the overhead is low (Fig. 7c), but when the offered traffic gets above the bottleneck capacity, the overhead rises to very high values, up to 32 %. We can conclude that multiplexing may produce unfairness between players in case of severe network congestion.

C. *Dependance with Player Activities*

All the previous tests have been carried out using the traffic model corresponding to *Questing* activity. As reported in [7], a game may generate very different traffic patterns depending on the activity that the player is performing at a certain moment. Thus, in this subsection we present some tests with the aim of studying the different influence of multiplexing for the different game activities.

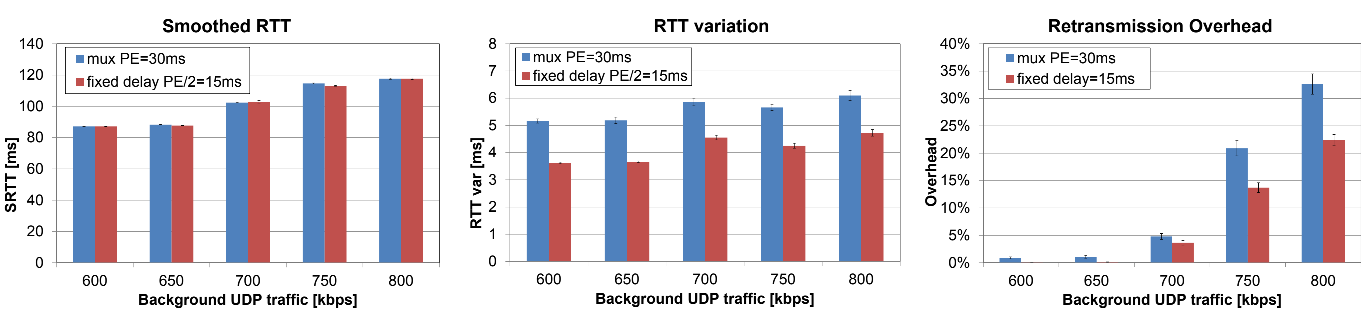
We have used the same scheme as in subsection *A*: 50 A-A' flows with a multiplexing period of 30 ms share the bottleneck with 50 B-B' flows with a fixed delay of 15 ms, and with an FTP upload connection. The statistics of the game flows correspond to *Trading, Questing, Dungeons, Raiding* and *Player vs Player Combat.* The results in terms of overhead are shown in Fig. 8.

It can be observed that the effect of multiplexing is more noticeable when the interactivity of the game is higher, as it happens in *Player vs. Player* and *Raiding* [7]. In these two activities, the amount of packets per second generated is higher, since a number of players are involved in the action, frequently organized in different teams who have to cooperate or to fight each other. The effect is very low in *Trading* and *Questing,* where the player mainly acts alone, and in *Dungeons,* where the number of players is lower.



(a) (b) (c)

Fig. 6. *Smoothed RTT*, *RTT variation* and Retransmission Overhead, with multiplexing delay *PE*, a fixed delay *PE*/2, and an FTP connection



(a) (b) (c)

Fig. 7. *Smoothed RTT*, *RTT variation* and Retransmission Overhead with multiplexing delay *PE*=30ms, a fixed delay of 15 ms, and UDP traffic

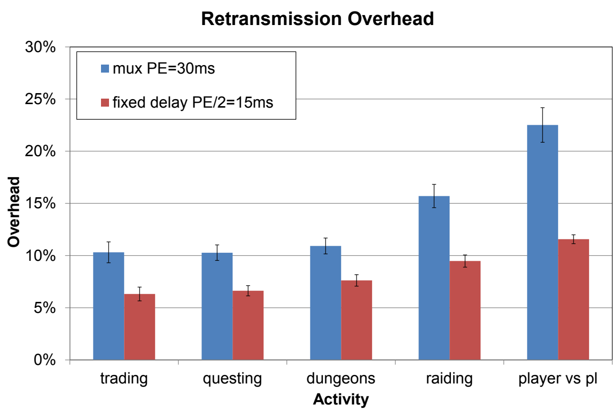


Fig. 8. Retransmission Overhead for different activities. *PE*=30 ms

VI. Conclusions

This paper has presented a study of the effect of traffic optimization techniques, when applied to the traffic of TCP-based online games. As a counterpart of bandwidth savings, traffic optimization requires the adding of a multiplexing delay, which has been characterized as a constant delay and a jitter. First, the maximum values of the multiplexing period, so as not to impair subjective quality, have been calculated. Then, a simulation scenario has been built, in which flows affected by traffic optimization share a link with other flows, in order to compare their performance, mainly measured in terms of RTT and retransmission overhead. Different tests using FTP and CBR background traffic have been deployed, showing the impairments caused by traffic optimization. Some recommended limits for the multiplexing period have been found. It has also been shown that multiplexing stresses the unfairness between flows if network congestion is severe.

Finally, the effect of multiplexing has been evaluated, depending on the different activities that the player can perform in the game. It is shown that the multiplexing delay is more noticeable for the most interactive activities.

VII. Acknowledgements

This work has been partially financed by the European Social Fund in collaboration with the Government of Aragon; CPUFLIPI Project (MICINN TIN2010-17298); TAMA Project, Government of Aragon; Ibercaja Obra Social; Project of Catedra Telefonica, University of Zaragoza.

References

[1] C. Perkins, Rtp: Audio and Video for the Internet. Addison-Wesley Professional, 2003.

[2] B. Thompson, T. Koren, D. Wing. RFC 4170: "Tunneling Multiplexed Compressed RTP (TCRTP)," November 2005.

[3] J. Saldana et al, "Tunneling Compressed Multiplexed Traffic Flows (TCMTF)," draft-saldana-tsvwg-tcmtf-05, Jul. 2013.

[4] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, J. I. Aznar, E. Viruete, L. Casadesus, "First Person Shooters: Can a Smarter Network Save Bandwidth without Annoying the Players?," IEEE Communications Magazine, vol. 49, no.11, pp. 190-198, Nov. 2011.

[5] S. Ratti, B. Hariri, S. Shirmohammadi, "A Survey of First-Person Shooter Gaming Traffic on the Internet," IEEE Internet Computing, pp. 60-69, Sep./Oct. 2010.

[6] J. Saldana, L. Sequeira, J. Fernandez-Navajas, J. Ruiz-Mas, "Traffic Optimization for TCP-based Massive Multiplayer Online Games," Proc. SPECTS 2012, July 2012, Genoa, Italy.

[7] M. Suznjevic, I. Stupar and M. Matijasevic, "A model and software architecture for MMORPG traffic generation based on player behavior," Multimedia Systems, 2012.

[8] E. Ertekin, C. Christou, "Internet protocol header compression, robust header compression, and their applicability in the global information grid," IEEE Communications Magazine, vol. 42, 2004, pp. 106-116.

[9] 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.

[10] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, J. I. Aznar, L. Casadesus, E. Viruete, "Comparative of Multiplexing Policies for Online Gaming in terms of QoS Parameters," IEEE Communications Letters, vol.15, no.10, pp.1132-1135, Oct. 2011.

[11] R. M. Pereira, L.M. Tarouco, " Adaptive Multiplexing Based on E-model for Reducing Network Overhead in Voice over IP Security Ensuring Conversation Quality," Proc. Fourth international Conference on Digital Telecommunications, Washington, DC, pp 53-58 , July 2009.

[12] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, E. Viruete Navarro, L. Casadesus, "Online FPS Games: Effect of Router Buffer and Multiplexing Techniques on Subjective Quality Estimators," Multimedia Tools and Applications, Springer 2013.

[13] C. Griwodz and P. Halvorsen, "The fun of using TCP for an MMORPG," Proc. international workshop on Network and operating systems support for digital audio and video (NOSSDAV '06). ACM, New York, NY, USA, 2006.

[14] C-C. Wu K-T. Chen, C-M. Chen, P. Huang, and C-L. Lei, "On the Challenge and Design of Transport Protocols for MMORPGs," Multimedia Tools and Applications, Vol. 45, No. 1, pp. 7-32, 2009.

[15] R. Jain, "A delay based approach for congestion avoidance in interconnected heterogeneous computer networks," Computer Comm. Review, ACM SIGCOMM, vol. 19, Oct 1989, pp. 56–71.

[16] G. Marfia, C.E. Palazzi, G. Pau, M. Gerla and M. Roccetti, "TCP Libra: Derivation, analysis, and comparison with other RTT-fair TCPs", Computer Networks, 2010, 54(14), 2327-2344.

[17] M. Ries, P. Svoboda, M. Rupp, "Empirical study of subjective quality for Massive Multiplayer Games," Proc. 15th Int. Conf. on Systems, Signals and Image Processing, pp.181-184, 2008.

[18] P. Svoboda, W. Karner, M. Rupp, "Traffic Analysis and Modeling for World of Warcraft," in Proc. ICC, Urbana-Champaign, IL, USA 2007.

[19] Y. Wu, H. Huang, D. Zhang, "Traffic Modeling for Massive Multiplayer On-line Role Playing Game (MMORPG) in GPRS Access Network," in Proc. Int. Conf. Communications, Circuits and Systems, pp. 1811–1815, 2006.

[20] T. Issariyakul, E. Hossain, Introduction to Network Simulator NS2. Springer, 2011.

[21] Cooperative Association for Internet Data Analysis, "NASA Ames Internet Exchange Packet Length Distributions".