Influence of the Router Buffer on Online Games Traffic Multiplexing

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

***Abstract*—This work presents a study of the behavior of a tunneling, compressing and multiplexing technique, named TCM, which can obtain bandwidth savings when many users of online games share the same path, as it occurs e.g. in Internet cafés. The main characteristics of the technique are summarized, and some tests are carried out with real machines which emulate two different buffer policies, using traffic traces of a popular First Person Shooter game. The results show that the obtained bandwidth savings can be significant, so they can improve user's experience, and that the best parameters for the protocol have to be empirically obtained in each case.**

Keywords-online gaming, multiplexing, first person shooter, buffer sizing, Internet café

I. Introduction

Since the middle of the 1990s, Internet cafés have given many users the opportunity of accessing Internet services. Nowadays, they still represent an important connection way for the users in some countries [1]. The profile of their users has been studied [2], and gaming has been reported as an important activity. Two of the most popular genres are First Person Shooters (FPS) and Massive Multiplayer Online Games (MMOGs). Internet cafés are present all over the world, but they have a special significance in developing countries [3].

This scenario, where many computers share the same Internet connection, has a very big variability: different network technologies depending on the telecommunications infrastructure of the country, different routing equipment and network topologies, etc. Bandwidth is considered a scarce resource, which has to be well administrated.

It can be frequent that a group of people go to a café to play a FPS game. The traffic of these applications consists of a high rate of small UDP packets, so if a group of users shares the same connection, the router may experience problems in order to manage all the packets. Reference [4] presented a traffic characterization of FPS games, and concluded that, as network devices are mainly designed for bulk data transfers using TCP, the router processing capacity can be a new bottleneck added to the link speed limit. MMOGs are less bandwidth demanding [5], and their real-time constraints are not so hard, so they are not affected in the same way as FPS. This is the reason why we will consider FPS traffic in this work.

Game providers have to deal with very demanding customers, who do not tend to be loyal to a server, and always look for the best one [6]. So many servers have a limit of simultaneous players, which may be selected depending on the processing and network capacity. In the scenario we are considering, the number of simultaneous users should depend on the available bandwidth of the Internet access, and, in case of having an asymmetric one, on the uplink capacity.

The local agent can group packets from different users, multiplexing them into bigger ones, thus obtaining two benefits: a reduction of the number of packets per second the router has to manage, and bandwidth savings, as small packets present a significant overhead. This technique has been largely used for other multimedia services, like Voice over IP (VoIP). The local agent can be placed into different locations in the scenario: it can be placed into a local machine (Figure 1a); or into the computer of one of the players (Figure 1b); finally, it could even be embedded into the router (Figure 1c), being able to know the current traffic distribution of the access network and using that information to properly tune multiplexing parameters.

This technique not only can be used in Internet cafés, but there exist other scenarios where it can be applied, e.g. at the infrastructure of a game provider. If the service is only provided by a central server, it will represent a bottleneck, so some game *proxies* can be used in order to transfer workload to network borders as shown in Figure 1d. The same thing could be done in LAN parties, where large numbers of players share the same path.

In [7] a method named TCM (Tunneling, Compressing and Multiplexing) was presented. By the addition of small delays, it is able to obtain bandwidth savings of 30% for client-to-server traffic of many games, and up to 50% for certain ones. Another effect of the technique is that packet size increases depending on the number of merged packets.

In the present work we will study the effect of this technique depending on the router's buffer behaviour. On one hand, bandwidth saving will be beneficial, but on the other hand the increase of the packet size may impair the quality for certain buffer policies that have a higher discarding probability for big packets.

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

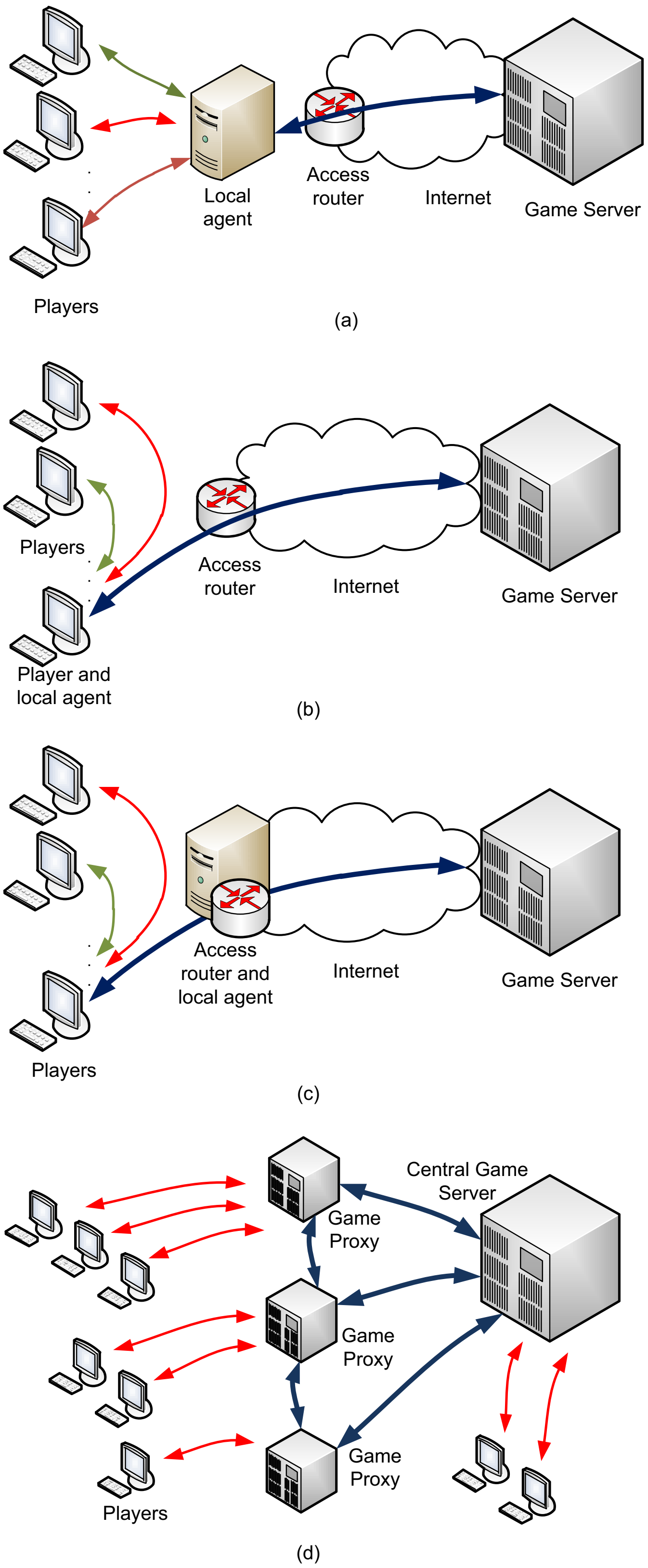


Figure 1. Different multiplexing scenarios: a) local agent as a machine in the local network; b) running into a player's machine; c) embedded into the router; d) traffic between proxies of a game provider.

The rest of the paper is organized as follows: next section presents the related works. Section 3 summarizes the system behaviour. Section 4 presents the tests and results, and the paper ends with the conclusions.

II. Related Works

The scenario we are considering may have very different technologies, and so will happen with the access router: there is a big variety of them. The problem of sizing the buffer of a router has been deeply studied. In the review presented in [8], Dhamdere and Drovolis explained that some years ago, the "rule of the thumb" of using the bandwidth-delay product was argued by the so-called "Stanford Model", which uses smaller buffers. In the same work, the authors also proposed a time-limited buffer, which discards the packets that spend more than a certain time in the queue. This buffer penalizes big packets, but it is interesting for real-time multimedia flows, as it maintains the delay under an upper bound. In this paper we will compare this approach with the use of bigger buffers.

Real-time services, like VoIP, video conference or online gaming, have very hard delay requirements, making the applications generate high rates of small packets, thus presenting a substantial overhead. Multiplexing solutions have been proposed and standardized [9] for the scenarios where many real-time flows share the same path, like it occurs in VoIP trunking. A big number of samples can be included into a single packet while only adding the retention delay corresponding to inter-packet time. So the bigger the number of flows, the better the bandwidth efficiency.

Regarding to the traffic of FPS games, it can be said that, although there are many different titles developed by a number of companies, they have similar traffic patterns [10]: from client to server, each player generates a high rate of small packets (some tens of bytes). These traffics are typically independent of the number of players, as each one only sends to the server the actions of a player [4], [11]. On the other hand, the packets sent from the server to the clients are bigger, as they include the information of the rest of the players, so their size depends on the number of users. In [7] a multiplexing, compressing and tunneling method was proposed and tested for client-to-server game traffic. It adapts the scheme of [9], but it uses different compressing algorithms, as RTP is not present.

And finally, MMOG games can also obtain benefit by grouping packets: a recent study of the traffic of a popular title [12] has concluded that P2P propagation schemes are not suitable for that kind of games. Another conclusion is that message aggregation before transmission can reduce both bandwidth and latency in client-server and P2P schemes.

III. Summary of System Behaviour

We will now make a brief summary of TCM and the savings it can achieve. As seen in Figure 1, the main idea is to add a local agent that merges into a bigger packet the ones that have arrived during a period of time named T_{period} (Figure 2). When a single packet has arrived, it is sent without modification, as the tunnel would add a bigger overhead. Figure 3 shows the scheme of a multiplexed packet: a header compression protocol, like IPHC or ROHCv2 is applied to the IP/UDP headers; next, PPPMux is used and finally the multiplexed packet is sent using an L2TP tunnel.

Logically, two new delays are added: first, a retention time which average will be half the period. Second, a processing time, that is expected to be small. In fact, in a similar case [13], it was about 1ms. The transmission delays of the local network are considered negligible, as LANs are usually faster than the Internet.

The bandwidth relationship (*BWR*), i.e. the quotient of the multiplexed and native bandwidths for TCM was found to be [7]:

$$BWR = \frac{\Pr(k=1)}{E[k]} + \Pr(k>1)\,\frac{CH}{E[k](NH+E[P])} + \\ + Pr(k>1)\,\frac{E[k\,|\,k>1]}{E[k]}\,\frac{MH+E[RH]+E[P]}{NH+E[P]} \qquad (1)$$

Where k represents the number of packets arrived during a period. CH, MH, RH and P are the different header and packet sizes represented on Figure 3, and NH refers to the size of a normal IP/UDP header (28 bytes for IPv4 and 48 for IPv6). The first term expresses the case when a single packet has arrived to the multiplexer. The second one expresses how the multiplexed packets share the common header, and gets smaller as $E[k]$ grows. Finally, the third term expresses the asymptote for *BWR* for the cases when the period and the number of players are big enough, so $Pr(k>1) \approx 1$ and $E[k|k>1] \approx E[k]$.

In order to carry out a battery of tests, a concrete game has to be selected. The chosen title has been Half Life Counter Strike 1.6. Although it was developed many years ago, it is still very popular and representative of FPS traffic. Another reason is the availability of many studies of its traffic [10], [11], [14]. Figure 4 shows the theoretical behaviour of *BWR* for different numbers of players, and it illustrates the asymptotic behaviour. For this game, a bandwidth saving of 30% can be achieved using IPv4.

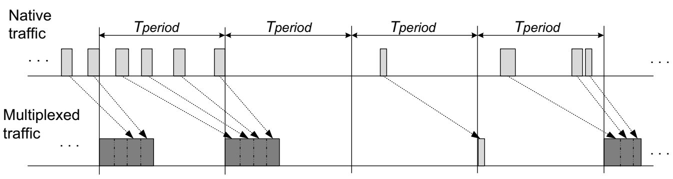


Figure 2. Multiplexing method.

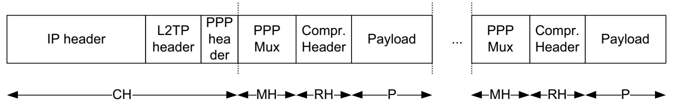


Figure 3. Scheme of a TCM packet.

IV. Tests and Results

The aim of the presented tests is the study of the mutual influence of the buffer policy and TCM. Although the buffer does not appear to have a direct influence on TCM, as packets are first multiplexed and then sent to the router, there exists a relationship: the bigger the period, the bigger the packets sent to the buffer. The behaviour of the packets in the router will depend on buffer policies, which will add different delays and losses depending on packet size. On the other hand, as TCM achieves bandwidth savings, it will reduce the traffic arrived at the router.

The traffic traces were obtained from [15], including only active game traffic. The traces were combined to obtain a trace of 20 players as said in [7]. Figure 5 shows the scenario used for the tests. First, the traffic of the game and the background traffic are sent from a machine running JTG [16] traffic generator, which is able to send the traces exactly as they were originally, as it reads the packet sizes and inter-departure times from a file. So a traffic model has not been necessary. The size distribution of background traffic is: 50% of the packets are of 40 bytes, 10% of 576 bytes, and 40% of 1500 bytes [17]. 810 seconds of traffic have been sent for each point of the graphs.

The traffic of the game and the background traffic share the same access link, which is emulated by a machine running Linux tool *tc* (Traffic Control). It limits the bandwidth at eth level and allows us to define the size of the buffer. The bandwidth limit has been set to 1Mbps. The *burst* parameter of *tc* has been set to 5000 bytes. The buffer size is defined by limiting the maximum delay of a packet, which is the same as limiting the buffer size, as the two parameters are related by the link speed. We have used two different buffers: first, a high capacity buffer (its maximum delay is 500 ms), and a time-limited buffer with a maximum delay of 50 ms.

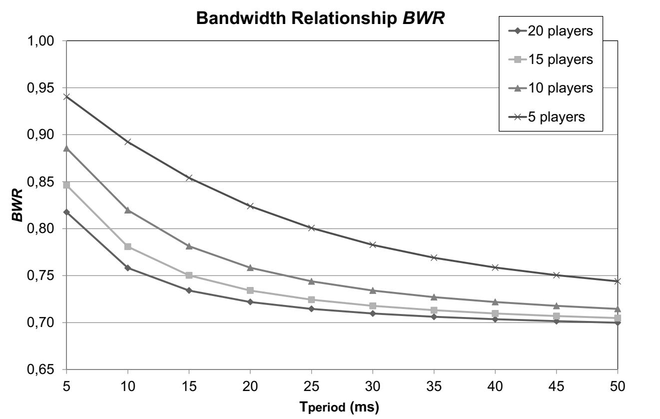


Figure 4. Bandwidth relationship for Counter Strike 1.6. using IPv4

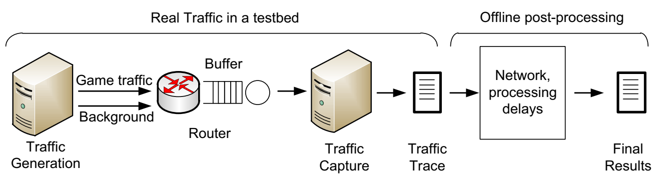


Figure 5. Measurement scheme.

The packets are collected by another machine, and then it is processed in order to add the network delay, which is the sum of a fixed delay of 20 ms corresponding to the geographical distance, and a lognormal one with an average of 20 ms and a variance of 5 [18]. A processing delay is also added. In [13] a multiplexing scheme for VoIP was implemented in real hardware, and its processing delay was about 1 ms. In order to include the effect of multiplex and demultiplex, we have added a fixed delay of 5 ms.

In [14] a study of Half-life was conducted and a conclusion is that players would not play when latencies are above 225-250 ms. More recent studies [19] have concluded that acceptable quality can be perceived with 200 ms of delay for certain titles. So the delays added by TCM can be assumed by the players. Regarding to packet loss, the behaviour depends on the game: while some of them stop working with packet loss about 4%, others can work properly with this parameter about 35% [19].

Next, we will present some graphs of One Way Delay (OWD) and packet loss for both buffers, using different amounts of background traffic in order to saturate the access router. Logically, multiplexing will only be interesting when the traffic of the game has to compete with big amounts of background traffic. For each buffer we have used three traffics: the *native* one, in which no multiplexing is applied, other using T_{period}=25 ms, and a third one with T_{period}=50 ms.

Figure 6 shows the results for the high capacity buffer. It should be noticed that a small increase of the delay is introduced when multiplexing, due to retention (half the T_{period}) and processing time in the multiplexer. The native bandwidth is 319 kbps at eth level. When the total traffic exceeds the limit, the delays grow up dramatically. It can also be seen that the bandwidth saving (about 120 kbps) is translated into a bigger amount of background traffic that can be supported while maintaining acceptable delays.

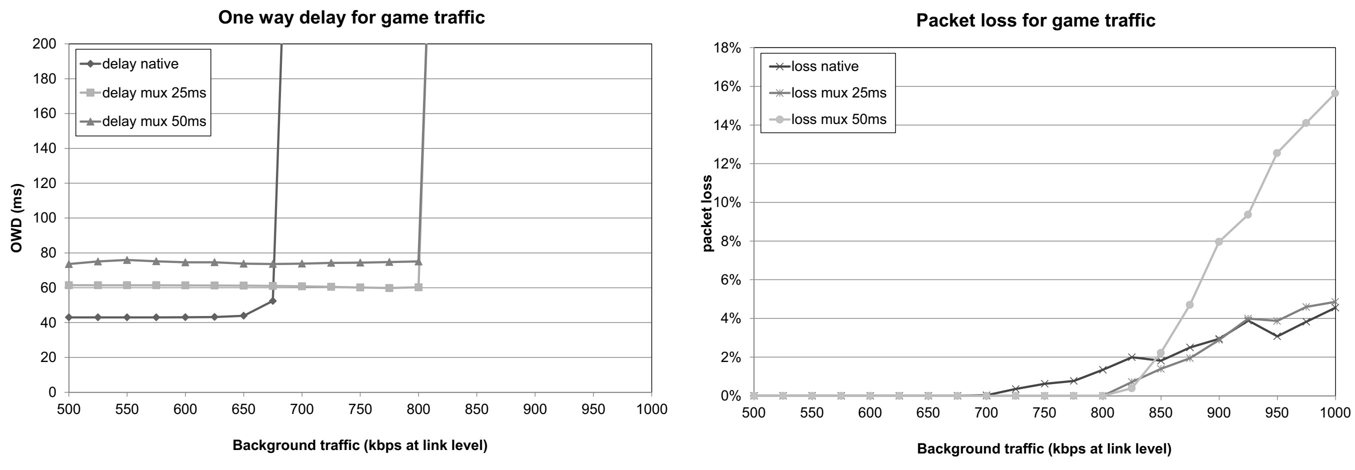
Figure 7 has been obtained using the time-limited buffer. If we compare it with Figure 6, we can see that the effects on the delay are the same as the ones observed for high capacity buffer, but the use of the time-limited one has the advantage of maintaining delay below 160 ms despite the amount of background traffic. There is a zone of the graph where the delay obtained when multiplexing is smaller than the native one, so this is an interesting result taking into account the hard real-time constraints of FPS games.

We can observe an interesting phenomenon: for the native traffic, the packet loss rate decreases as the background traffic increases from 850 to 925 kbps. This happens because the bandwidth limit is reached, so the first packets being discarded are the big ones (1500 bytes), as they have a bigger probability of not having place at the queue. This represents a benefit for native packets, as they are very small. But multiplexed graphs do not show this effect, because their packets are bigger.

There is another remarkable effect regarding to packet loss: the use of T_{period} =25 ms achieves better results than native traffic, due to bandwidth saving, but it is also better than the use of T_{period} =50 ms. We can discuss this surprising result looking at the *20 players* graph of Figure 4: the values of *BWR* for 25 and 50ms are very similar, as they are near the asymptote. In fact, the difference in terms of bandwidth is smaller than 6 kbps. But if we calculate the average packet size, we can see that in the first case it is 608 bytes, and in the second one it is 1192 bytes. So, as the buffer policy penalizes big packets, it will be better not to use a big period.

But above 925 kbps of background traffic, it can be seen that native traffic has less packet loss than multiplexed ones for both buffers. The cause is that smaller packets have less probability of being discarded. So there are some situations in which multiplexing can increase packet loss. So multiplexing will affect the delay and packet loss of the game in a different manner depending on the router's buffer policy.

Next, we will analyze the results for background traffic. Figure 8 shows background traffic packet loss for both buffers. Discontinuous lines represent the high capacity buffer values. The obtained graphs are very similar. Bandwidth saving is translated into a smaller packet loss probability, so multiplexing will always be beneficial if we want to avoid harming background traffic. The buffer policy does not have a direct influence on packet loss for background traffic.



(a) (b)

Figure 6. High capacity buffer: a) One Way Delay b) packet loss

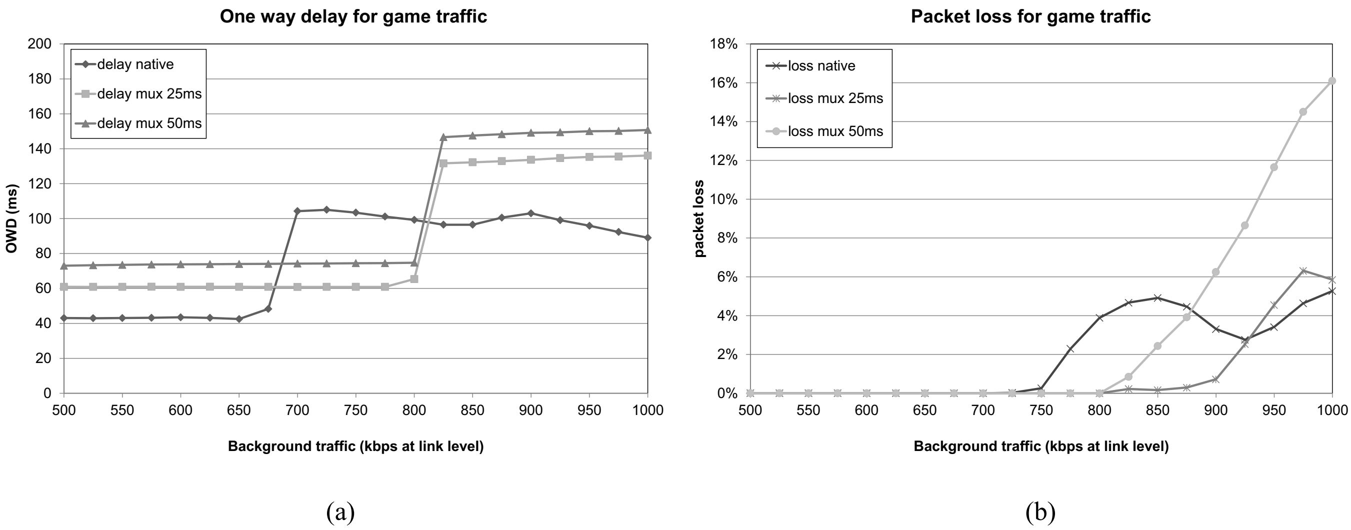


Figure 7. Time-limited buffer: a) One Way Delay b) packet loss

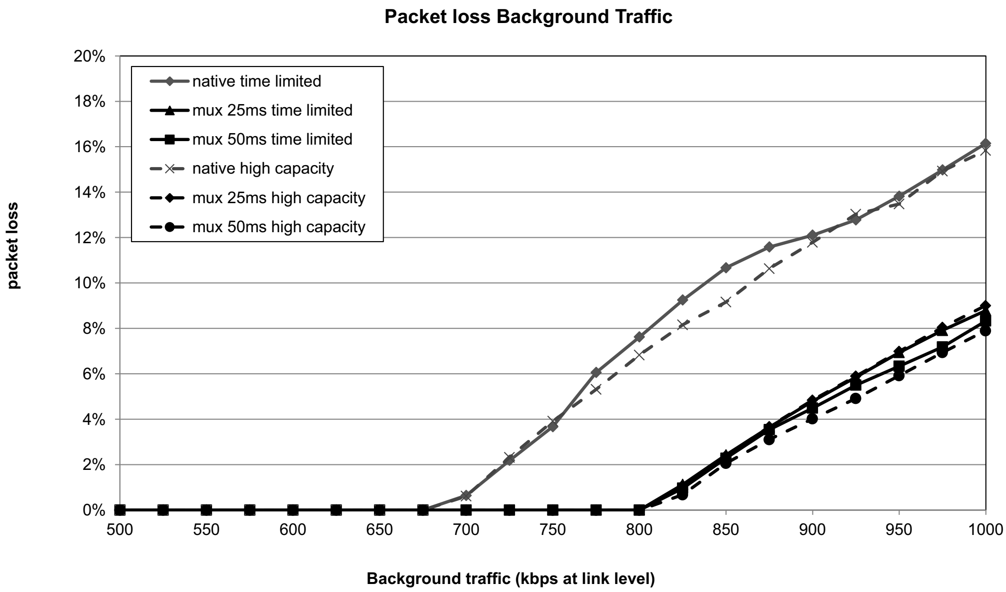


Figure 8. Packet loss for background traffic for both buffers

V. CONCLUSIONS

A comparison of the behaviour of native and TCM multiplexed flows of FPS, depending on router buffer size, has been carried out, showing that the best multiplexing solution is not always the one that achieves the best bandwidth saving. Packet size has to be considered too, as some policies penalize big packets.

Taking into account the high variety of routers that can be found in the scenario, the presented measurements illustrate the need for particularizing this problem for every concrete case: each network will have different delays, different behaviour regarding to packet loss, different distributions and amounts of background traffic, and a different number of packets per second the router is able to manage. So TCM technique can help us to adapt our traffic to the network behaviour: if the network is better prepared for low rates of big TCP packets, we can modify the traffic in order to adapt it to the underlying technology.

This work is part of a project which is currently focusing its research into many related topics: the characterization of commercial buffers, the study of the traffic of other popular games, and the different possibilities of locating the multiplexer and demultiplexer, in order to provide game manufacturers with a flexible technique which can be useful to improve the gaming experience in scenarios where many users share the same path.

REFERENCES

[1] S. H. Batool and K. Mahmood, "Entertainment, communication or academic use? A survey of Internet cafe users in Lahore, Pakistan," Information Development vol. 26. pp 141-147, May 2010.

[2] M. Gurol and T. Sevindik, "Profile of Internet Cafe users in Turkey," Telematics and Informatics, Vol. 24, Issue 1, pp 59-68, Feb. 2007.

[3] B. Furuholt, S. Kristiansen, F. Wahid, "Gaming or gaining? Comparing the use of Internet cafes in Indonesia and Tanzania," The Intern. Inform. & Library Review, Volume 40, Issue 2, pp 129-139, Jun. 2008.

[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] K. Chen, P. Huang, C. Lei: Game traffic analysis: An MMORPG perspective. In Proc. International workshop on Network and operating systems support for digital audio and video (NOSSDAV'05), pp. 19-24. ACM, New York, 2005.

[6] C. Chambers, W. Feng, S. Sahu, D. Saha: "Measurement-based Characterization of a Collection of On-line Games," In Proc. 5th ACM SIGCOM conference on Internet Measurement (IMC'05). USENIX Association, Berkeley, 2005

[7] 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. International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS 2011, The Hague, Netherlands Jun. 2011.

[8] A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing," Comput. Commun. Rev., vol. 36, no. 1, pp. 87-92, Jan. 2006.

[9] B. Thompson, T. Koren, D. Wing. "RFC 4170: Tunneling Multiplexed Compressed RTP" Nov. 2005.

[10] 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.

[11] T. Lang, G. Armitage, P. Branch, H. Choo, "A Synthetic Traffic Model for Half-Life," In Australian Telecom, Networks and Applications Conference (ATNAC) Melbourne, 2003.

[12] J. L. Miller, J. Crowcroft, "The Near-Term Feasibility of P2P MMOGs," in Proc. of International Workshop on Network and Systems Support for Games (NetGames), 2010.

[13] H. Sze, C. Liew, J. Lee, D. Yip: A Multiplexing Scheme for H.323 Voice-Over-IP Applications. IEEE J. Select. Areas Commun. Vol. 20, pp. 1360-1368, 2002.

[14] T. Henderson, "Latency and User Behaviour on a Multiplayer Game Server", Lect. Notes Comp. Science, Springer Berlin/ Heidelberg, pp 1-13, vol 2233, 2001.

[15] L. Stewart, P. Branch: HLCS, Map: dedust, 5 players, 13Jan2006. Centre for Advanced Internet Architectures SONG Database, http://caia.swin.edu.au/sitcrc/ hlcs_130106_1_ dedust_5_fragment.tar.gz

[16] J. Manner, JTG, http://www.cs.helsinki.fi/u/jmanner/software/jtg/, last access 5/4/2011.

[17] Cooperative Association for Internet Data Analysis: NASA Ames Internet Exchange Packet Length Distributions.

[18] S. Kaune, K. Pussep, C. Leng, A. Kovacevic, G. Tyson, R. Steinmetz. "Modelling the internet delay space based on geographical locations". In 17th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP 2009), Feb 2009.

[19] S. Zander, G. Armitage, "Empirically Measuring the QoS Sensitivity of Interactive Online Game Players," Australian Telecommunications Networks & Applications Conference 2004 (ATNAC2004), Sydney, Australia, Dec. 2004.