Widening the Scope of a Standard: Real Time Flows Tunneling, Compressing and Multiplexing

Jose Saldana[1], Dan Wing[2], Julián Fernández-Navajas[1], José Ruiz-Mas[1], Muthu A. M. Perumal[2], Gonzalo Camarillo[3]
[1]Communication Technologies Group (GTC) - Aragon Institute of Engineering Research (I3A)
Dpt. IEC. Ada Byron Building. EINA, University of Zaragoza, 50018 Zaragoza, Spain
e-mail: {jsaldana, navajas, jruiz}@unizar.es

[2]Cisco Systems
771 Alder Drive, San Jose, CA 95035, US
e-mail: {dwing, mperumal}@cisco.com

[3]Ericsson Research
Jorvas, Finland
e-mail: gonzalo.camarillo@ericsson.com

***Abstract*—The use of the Internet for delivering real-time traffic flows makes it necessary to fragment the information into small pieces, thus making the traffic have a bad efficiency. In order to avoid this, the IETF defined TCRTP as a mechanism capable of compressing headers and multiplexing a number of RTP packets into a bigger one, which was sent using a tunnel. However, some new applications that can be considered real-time, generate UDP or TCP packets, but do not use RTP. Based on some research results deployed in the academic context, a proposal for widening the scope of this standard is being developed in collaboration with the industry. Different traffics could be considered, and this would imply the possibility of utilizing a number of header compression algorithms, and also different multiplexing and tunneling protocols. This new standard could be used by different enterprises, as network operators, network equipment developers and game providers.**

Keywords-multiplexing; header compression; real-time; tcrtp; standardization

I. INTRODUCTION

At the beginning of the Internet, the modification of the protocols was a feasible task. The last big change, which separated transport and addressing functionalities, by the substitution of NCP by TCP/IP, was performed on January 1, 1983, when the Internet barely included 400 nodes [1]. It was relatively easy to synchronize every node so as to perform the change. Many years later, the size of the Internet does not permit this kind of changes. As an example, we can consider the IPv4 to IPv6 transition [2], which is lasting too many years in spite of the problems that IPv4 presents, such as the shortage of IP addresses. As a consequence, the changes have to be taken step-by-step, and they cannot be imposed, but they have to reach a great consensus in order to be widely implemented. In fact, IETF uses the concept of "rough consensus" [3] in the working groups, in order to approve a new proposal as a standards track.

At the same time, in the last years we are witnessing the raise of new real-time services that use the Internet for the delivery of interactive multimedia applications. The most common of these services is VoIP, but many others have been developed, and are experiencing a significant growth: videoconferencing, telemedicine, video vigilance, online gaming, etc.

The first design of the Internet did not include any mechanism capable of guaranteeing an upper bound for delivery delay, taking into account that the first deployed services were e-mail, file transfer, etc., in which delay is not critical. RTP (Real-Time Protocol) was first defined by the IETF Audio/Video Transport Working Group in 1996 in order to permit the delivery of real-time contents. Nowadays, although there are a variety of protocols used for signaling real-time flows (SIP, H.323, etc.), RTP has become the standard par excellence for the delivery of real-time content.

RTP was designed to work over UDP datagrams. This implies that an IPv4 packet carrying real-time information has to include 40 bytes of headers: 20 for IPv4 header, 8 for UDP, and 12 for RTP. This overhead is significant, taking into account that many real-time services send very small payloads. It becomes even more significant with IPv6 packets, as the basic IPv6 header is twice the size of the IPv4 header. As an example, the header overhead for G.711 and G.729a RTP payloads carried in IPv4 and IPv6 packets are shown in Table I.

In order to mitigate this bad network efficiency, the multiplexing of a number of payloads into a single packet can be considered as a solution. If we have only one flow, the number of samples included in a packet can be increased, but at the cost of adding new packetization delays. However, if a number of flows share the same path between an origin and a

TABLE I. EFFICIENCY OF DIFFERENT VOICE CODECS

IPv4	IPv6
IPv4 + UDP + RTP 40 bytes header	IPv6 + UDP + RTP 60 bytes header
G.711 at 20 ms packetization 25% header overhead	G.711 at 20 ms packetization 37.5% header overhead
G.729a at 20 ms packetization 200% header overhead	G.729a at 20 ms packetization 300% header overhead

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.

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

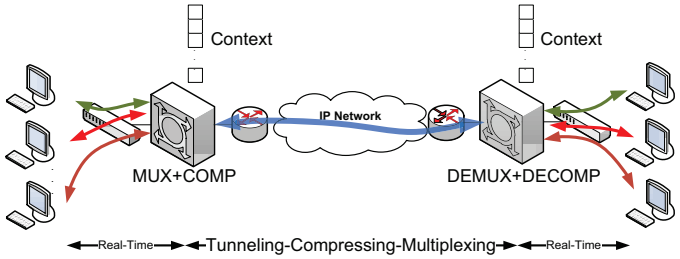

Figure 1. Tunneling, Compressing and Multiplexing scheme

destination (Fig. 1), a multiplexer can build a bigger packet in which a number of payloads share a common header. A demultiplexer is necessary at the end of the common path, so as to rebuild the packets as they were originally sent, making multiplexing a transparent process for the extremes of the flow.

The headers of the original packets can be compressed to save more bandwidth, taking into account that there exist some header compressing standards for RTP headers [4] [5] [6]. When IP, UDP and RTP headers are compressed together, tunneling can be used to relieve intermediate routers from the decompression and compression processing.

In the late 90's, the IETF Audio/Video Transport Working Group felt the need of standardizing some multiplexing method which could be able to reduce bandwidth in RTP flows. The group discussed different proposals, coming from academia and industry, and finally the approved standard included a RTP compressing protocol, a multiplexing one and a tunneling one [7]. The standard was approved as "best current practice" since it did not define any new protocol, but it only established a way to combine them.

But there are many real-time applications that do not use RTP. Some of them send UDP packets, e.g. First Person Shooter (FPS) online games [8], for which latency is very critical, as remarked in [9]. There is also another fact which has to be taken into account: TCP is getting used for media delivery [10]. For many reasons, such as avoiding firewalls, the standard IP/ UDP/ RTP protocol stack is substituted in many cases by IP/ TCP/ HTTP/ FLV.

There is also another kind of applications which have been reported as real-time using TCP: MMORPGs (Massively Multiplayer Online Role Playing Games), which in some cases have millions of players [11], thousands of them sharing the same virtual world. They use TCP packets to send the player commands to the server, and also to send to the player's application the characteristics and situation of other gamers' avatars. These games do not have the same interactivity of FPSs, but the quickness and the movements of the player are important, and can decide if they win or lose a fight. So in [12] these games have been defined as real-time using TCP.

Some recent research [13] has highlighted the usefulness of these techniques for non-RTP flows. Taking these facts into account, the spectrum of the multiplexing, compressing and tunneling standard for RTP can be widened so as to include not only RTP flows, but also UDP and even TCP ones.

Different scenarios of application can be considered for the Tunneling, Compressing and Multiplexing of Traffic Flows (TCMTF) solution: for example, the traffic of the users of an application in a town or a district can be multiplexed and sent to the central server (Fig. 2a). Also Internet cafés are suitable of having many users of the same application (e.g. a game) sharing the same access link (Fig. 2b).

Another interesting scenario is satellite communication links (Fig. 2c) that often manage the bandwidth by limiting the transmission rate, measured in packets per second (pps), to and from the satellite. Applications like VoIP that generate a large number of small packets can easily fill the limited number of pps slots, limiting the throughput across such links. As an example, a G.729a voice call generates 50 pps at 20 ms packetization. If the satellite transmission allows 1,500 pps, the number of simultaneous voice calls is limited to 30. This results in poor utilization of the satellite link's bandwidth as well as places a low cap on the number of voice calls that can utilize the link simultaneously. Multiplexing small packets into one packet for transmission would improve the efficiency. Satellite links would also find it useful to multiplex small TCP packets into one packet – this could be especially interesting for compressing TCP ACKs.

In conclusion, the development or a new standard with a wider scope seems necessary, and can be interesting for many enterprises: developers of VoIP systems can include this option in their solutions; or game providers, who can achieve bandwidth savings in their supporting infrastructures. Other fact that has to be taken into account is that the technique not only saves bandwidth but also reduces the number of packets per second, which sometimes can be a bottleneck for a satellite link or even for a network router [8].

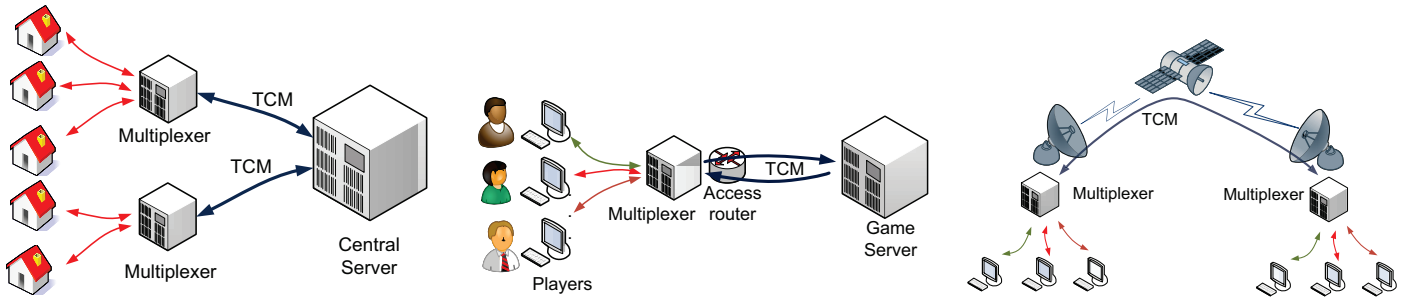

Figure 2. Scenarios where the scheme can be used: a) aggregating traffic of a district; b) traffic of different gamers in an Internet café; c) satellite link

The rest of the paper is organized as follows: the context and the history of the standardization of TCRTP, and a brief description, are included in the next section. Some examples of the achievements that can be performed when using the new proposal are summarized on Section III. Section IV presents the new proposed standard in more detail. The paper ends with the conclusions.

II. Context, History and Description of RFC 4170

In 1999, there was a desire to send voice over an IP network with bandwidth efficiency on par with voice over ATM. Already existent was Compressed RTP (cRTP) [4], which compresses RTP, UDP, and IP headers. However, cRTP only works on a link and could not accommodate significant delay, reordering, or packet loss, characteristics of most any network. cRTP was extended to support such network characteristics and standardized as ECRTP (Enhanced cRTP) [5]. When operating on routers (rather than on endpoints themselves), ECRTP needs to be tunneled.

One way to tunnel ECRTP is described in RFC4170 ("TCRTP") [7], which combines several technologies shown in Fig. 3. First, it runs ECRTP over PPPMux to gain the ability to multiplex several ECRTP payloads into one IP packet. A similar technique was used by voice over ATM. PPP cannot run natively over a network, so PPP is run over an L2TP tunnel. Finally, this is all run over IP. A common perception of tunneling is that it makes packets larger, because tunneling simply runs one protocol over another. However, TCRTP always saves bandwidth because it compresses the IP, UDP, and RTP headers from 40 bytes to 26 bytes.

The first header compression techniques had been developed many years ago, when Van Jacobson defined VJHC, which was able to jointly compress TCP/IP headers, on 1990 [14]. Later, IPHC [15] was defined, as the IETF felt necessary to deploy a standard also able to compress UDP and IPv6 headers. At the same time, cRTP was defined, being able to compress RTP headers too. The improvement of this technique so as to perform well in high-RTT and lossy links was ECRTP, the compression technique used by TCRTP.

These techniques are based on the fact that many header fields are the same for every packet in a flow. Other improvement is the use of *delta* compression so as to reduce the number of bits required by a field, by transmitting only the difference between the value of a field in a packet and in the previous one. Bandwidth can be saved, but at the cost of defining a *context,* which is a set of variables that has to be synchronized between the sender and the receiver (Fig. 1). So we can conclude that, in addition to the drawback of needing a tunnel, another counterpart of header compression is the possibility of context desynchronization, which would imply bursts of corrupted packets. So the compressing techniques define different mechanisms so as to grant the synchronization.

Since ECRTP and TCRTP were standardized, there has been additional interest in header compression. In 2006, the IETF formed the Robust Header Compression (ROHC) Working Group which created specifications for header compression over links for RTP, UDP and TCP [6]. These

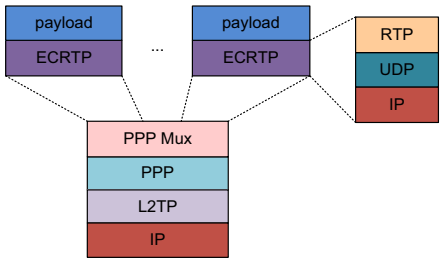

Figure 3. TCRTP protocol stack

specifications were extended to compress headers over other networks. For most RTP flows, ROHC is more bandwidth efficient than ECRTP, at the cost of implementation complexity, which would imply more processing and memory resources. It has a special care of context synchronization, defining some states at the compressor and decompressor. Once the effort for designing ROHC has been conducted, it is worth including it in the compressing and multiplexing standard.

Regarding multiplexing, although the sharing of the common header is beneficial, it also presents some counterparts: first, additional delays will appear, since the multiplexer will have to wait in order to obtain the number of packets that it will merge into a bigger one. Second, an extra jitter will also be added, taking into account that some packets will spend more time than others at the multiplexer's queue. Finally, the packet size will be increased, and this can have an influence on packet loss, as shown in [16]. In order not to harm subjective quality when using multiplexing techniques, different quality estimators have been proposed [17].

Other multiplexing proposals apart from TCRTP can be found in the literature: Sze *et al* [18] proposed the inclusion of a number of RTP packets into a single UDP one, also compressing RTP headers. Other proposal [19] presented a similar solution, but it did not include header compression. In [20] a method to adapt the transmission rate while multiplexing was proposed. GeRM proposed the inclusion of a number of compressed RTP packets inside a bigger RTP one [21]. Finally, [22] proposed the assembly of audio samples from different flows into a single RTP packet, using a 2-byte mini header so as to identify each one.

III. Examples of the Improvements that can be Achieved

In this section we present some examples of the improvements provided by the proposed standard. First, we will present some measurements which were carried out in [16]. The bandwidth saving has been measured as the quotient of TCRTP bandwidth divided by native one. Fig. 4 shows that significant savings can be achieved when multiplexing different numbers of G.729a voice flows, depending on the number of samples per packet (1, 2 or 3 samples, which means 10, 20 or 30 bytes of payload).

It can be seen that the saving presents an asymptote. This implies that, when the number of flows to multiplex is high, the difference in bandwidth will be small, while packet size will maintain its increase. As a consequence, if the router buffer

penalizes big packets, it will be more interesting, in terms of subjective quality, to group the flows into a number of tunnels [23] than establishing a single tunnel including all the flows.

As an example of the gaining which can be achieved for FPS games, which send UDP packets, we present (Fig. 5) the bandwidth saving for a game (*Counter Strike 1*), as a function of the number of players and the multiplexing period. IPHC was used for compressing the headers. The subjective quality can be maintained in certain conditions [24].

Finally, we will present some tests which have been deployed using the traffic of an MMORPG (*World of Warcraft*). As we have said, these games use TCP to transmit the movements of the player to the server. In order to reduce the delay, the bit *push* of the header is always set to 1, and this makes the protocol send the packet immediately. As a consequence, the generated packets are small [12]. By the use of simulations with the traffic model presented in the same paper, preliminary bandwidth savings results have been obtained using different values for the period and the number of players (Fig. 6), using IPHC compression. It can be observed that savings higher than the ones obtained for FPSs can be achieved, but on the other hand, the period and the number of players have to be bigger. This is not a problem, since the interactivity of these games is not as critical as in FPSs. These values of the period can be used while maintaining an acceptable subjective quality [25]. Regarding the number of players, it is not a problem, since in these games the virtual scenario can be shared by thousands of simultaneous players.

IV. Proposal of a Widened Standard

The contact between academy researchers and the industry has permitted the first steps toward the formal definition of this proposal, which will be presented in the context of the Transport Area Working Group in the IETF 83 meeting in Paris (March 2012). The main idea is to make it cover a wider scope, being able not only to save bandwidth for RTP, but also for real-time flows using other protocols. Fig. 7 shows the scheme of this new proposal. It includes different possibilities for real-time traffic. The "far right side" of the scheme is the current TCRTP standard, which only multiplexes RTP content.

Regarding compression, a number of options have to be considered: as the standards are able to compress different headers, the one to be used could be selected depending on the traffic to compress, and also taking into account the availability of processing and memory resources. In addition, the proposal would also include the possibility of having a null header compression, for clients that do not want to compress traffic, taking into account the need of storing a context for every flow and the problems of context desynchronization in certain scenarios. Further analysis will be required in order to study the tradeoffs between compressing rate and processing and memory requirements in each case.

With respect to multiplexing, a simpler mechanism, rather than PPPMux, could also be considered as a possibility, but it would have to be newly defined; and also a simpler tunneling protocol, which could be used instead of L2TP. GRE could be the selected option [26].

There is another topic which has to be considered: although many voice codecs generate samples at a fixed rate, in other services inter-packet time does vary. So in this case, we have to define a policy so as to decide which packets are multiplexed and when. This can be done using a fixed number of packets or a maximum packet size. Nevertheless, in order to set an upper bound for the added delay, multiplexing policies, based on a period or a timeout, could be considered more adequate. A comparative between them was presented in [27].

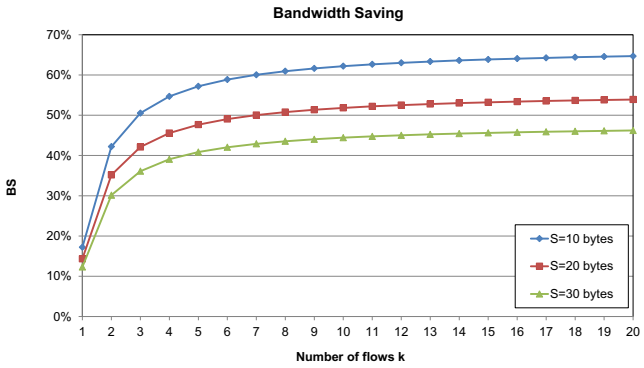

Figure 4. Bandwidth saving using TCRTP for G.729a VoIP flows

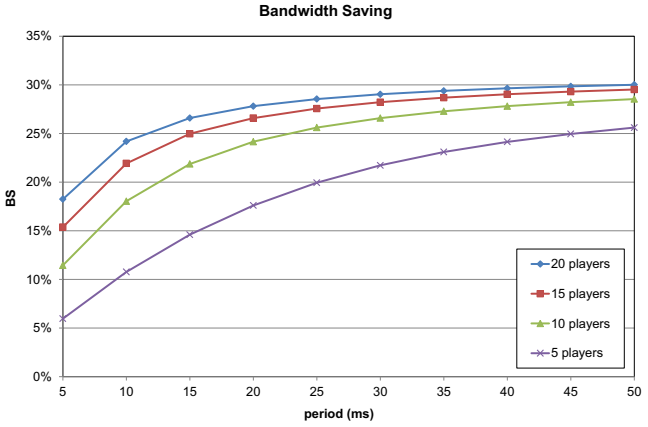

Figure 5. Bandwidh saving for *Counter Strike 1* using the proposed method

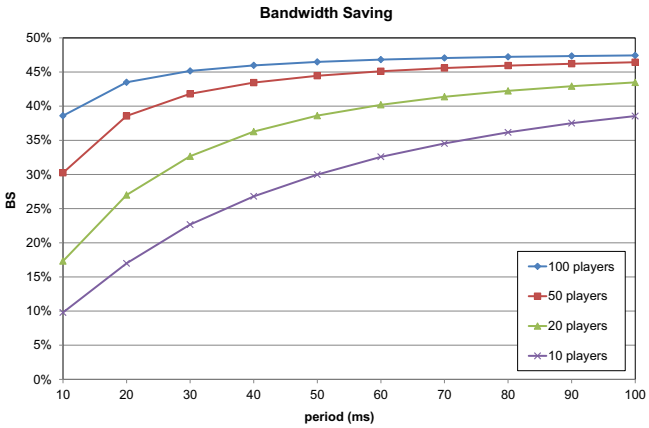

Figure 6. Bandwidh saving for *World of Warcraft* using the proposed method

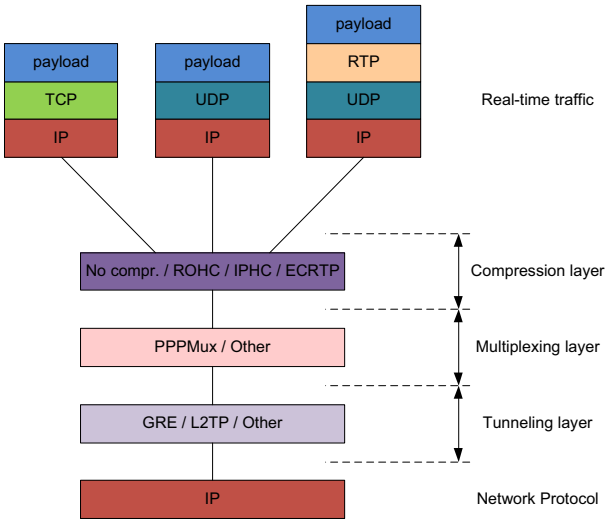

Figure 7. Protocol stack of the new proposed standard

The mechanisms necessary to establish the tunnel between endpoints will have to be defined by the standard, and also the negotiation protocol which could be used in order to decide the option to use in each layer.

V. CONCLUSION

In this article the problem of improving bandwidth efficiency in real-time flows has been first highlighted. Different scenarios of application have been identified. Next, different techniques that can be used for improving the efficiency, as multiplexing and header compression, have been described, and also the standardized solution for multiplexing RTP flows. The history of this standard has been summarized. Next, some results obtained using real-time applications that do not send RTP packets have been presented, mainly showing the bandwidth savings that can be achieved.

A proposal for widening the scope of the current standard has been presented, showing the need of defining a policy to decide which packets will be included in each multiplexed one. The collaboration between academia and industry has been shown as an interesting way to improve the existing standards, demonstrating that the two domains can walk in the same way.

REFERENCES

[1] M. Handley, "Why the Internet only just works," BT Technology Journal 24, 3, pp 119-129, Jul. 2006.

[2] L. Colitti, S. H. Gunderson, E. Kline and T. Refice, "Evaluating IPv6 adoption in the Internet," Proc. 11th Int. Conference on Passive and Active Network Measurement, pp 141–150. Springer-Verlag, Apr. 2010.

[3] S. Bradner, RFC 2418, "IETF working group guidelines and procedures," Sep. 1998.

[4] S. Casner and V. Jacobson, RFC 2508, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links," Feb. 1999.

[5] T. Koren, S. Casner, J. Geevarghese, B. Thompson and P. Ruddy, RFC 3545, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering," Jul. 2003.

[6] L-E. Jonsson, G. Pelletier and K. Sandlund, RFC 4995, "The RObust Header Compression (ROHC) Framework," Jul. 2007.

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

[8] W. Feng, F. Chang, W. Feng and J. Walpole, "A Traffic Characterization of Popular On-Line Games," IEEE/ACM Trans. Networking, pp. 488-500, 2005.

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

[10] G. Marfia and M. Roccetti, "TCP At Last: Reconsidering TCP's Role for Wireless Entertainment Centers at Home," IEEE Transactions on Consumer Electronics, Vol. 56, N. 4, pp. 2233-2240, Nov. 2010.

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

[12] P. Svoboda, W Karner and M. Rupp, "Traffic Analysis and Modeling for World of Warcraft," ICC '07. IEEE International Conference on Communications, pp.1612-1617, 24-28, Jun. 2007.

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

[14] V. Jacobson, RFC 1144, "Compressing TCP/IP Headers for Low-Speed Serial Links," Feb. 1990.

[15] M. Degermark, B. Nordgren and D. Pink, RFC 2507, "IP Header Compression," Feb. 1999.

[16] J. Saldana, J. Murillo, J. Fernández-Navajas, J. Ruiz-Mas, E. Viruete and J. I. Aznar, "Evaluation of Multiplexing and Buffer Policies Influence on VoIP Conversation Quality," Proc. CCNC 2011. The 8th Annual IEEE Consumer Communications and Networking Conference, pp 1147-1151, Las Vegas. Jan. 2011.

[17] F. Kuipers, R. Kooij, D. De Vleeschauwer and K. Brunnstrom, "Techniques for measuring Quality of Experience," Wired/Wireless Internet Com., Springer-Verlag Berlin/Heidelberg, pp. 216-227, 2010.

[18] H.P. Sze, S. C. Liew, J.Y.B. Lee and D.C.S.Yip, "A Multiplexing Scheme for H.323 Voice-Over-IP Applications," IEEE J. Select. Areas Commun, Vol. 20, pp. 1360-1368, Sep. 2002.

[19] T. Hoshi, K. Tanigawa and K. Tsukada, "Proposal of a method of voice stream multiplexing for IP telephony systems," Proc. IWS '99, pp. 182–188, Feb. 1999.

[20] A. Trad and H. Afifi, "Adaptive Multiplexing Scheme for Voice Flow Transmission Across Best-Effort IP Networks," INRIA Research Report 4929, Sep. 2003.

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

[22] B. Subbiah and S. Sengodan. draft-ietf-avt-mux-rtp-00.txt, "User Multiplexing in RTP payload between IP Telephony Gateways," Aug. 1998.

[23] J. Saldana, J. Murillo, J. Fernández-Navajas, J. Ruiz-Mas, E. Viruete and J. I. Aznar, "Influence of the Distribution of TCRTP Multiplexed Flows on VoIP Conversation Quality," Proc. CCNC 2011. The 8th Annual IEEE Consumer Communications and Networking Conference, pp 711-712, Las Vegas. Jan. 2011.

[24] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, E. Viruete Navarro and L. Casadesus, "Influence of Online Games Traffic Multiplexing and Router Buffer on Subjective Quality," in Proc. CCNC 2012- 4th IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology (DENVECT), pp. 482-486, Las Vegas. Jan 2012.

[25] M. Ries, P. Svoboda and M. Rupp, "Empirical study of subjective quality for Massive Multiplayer Games," Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on, pp.181-184, 25-28, June 2008.

[26] D. Farinacci, T. Li, S. Hanks, D. Meyer and P. Traina, RFC 2784, Generic Routing Encapsulation (GRE), March 2000.

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