Evaluation of Multiplexing and Buffer Policies Influence on VoIP Conversation Quality

Jose Saldana, Jenifer Murillo, Julián Fernández-Navajas, José Ruiz-Mas, Eduardo Viruete Navarro, José I. Aznar
Communication Technologies Group (GTC) – Aragon Institute of Engineering Research (I3A)
Dpt. IEC. Ada Byron Building. CPS Univ. Zaragoza. 50018 Zaragoza, Spain
e-mail: {jsaldana, jenifer.murillo, navajas, jruiz, eviruete, jiaznar}@unizar.es

***Abstract*-This work presents a study of RTP multiplexing schemes, which are compared with the normal use of RTP, in terms of ITU R-factor quality estimator. The bandwidth saving of the different schemes is studied, and some tests with VoIP traffic are carried out in order to compare R-factor using three different router buffer policies. Network delays are added using an adequate statistical distribution. The tests show the bandwidth savings of multiplexing, and also the importance of the packet size with time-limited buffer policies. The customer experience improvement which can be achieved is measured in terms of R-factor, showing that the use of multiplexing can be interesting in some scenarios, like an enterprise with different offices connected via Internet.**

***Keywords*-RTP multiplexing, QoS, VoIP, R-factor, buffer policies**

I. Introduction

The use of the Internet for voice transmission, called VoIP (Voice over IP) is growing as bandwidth increases. There are some wide deployed signalling protocols like SIP and H.323, and RTP is the most used protocol for media transport.

Voice information can be highly compressed by the use of a suitable codec, but RTP has the problem of its big overhead. For example, a G. 729a packet with 2 voice samples will have 20 bytes of information, plus 40 bytes for the IPv4, UDP and RTP headers. As a result, only one third of the bytes carry voice information. Of course, if IPv6 is used, the efficiency becomes even worse.

Some RTP header compression schemes have been proposed, but the IP and UDP headers can not be removed. CRTP [1] uses the repeatability of headers in order to compress them. But the problem is that it has to be applied in a hop-by-hop way, at every router in the path. ECRTP [2] introduced some extensions in the protocol in order to enhance its behaviour in scenarios with packet loss, packet reordering and long delays.

Other possible solution is to place multiple samples in one packet [3]. This can be achieved by bundling more voice samples of the same flow. The problem is that each sample will increase the delay in the sender. Finally, there exists the possibility of multiplexing samples of different conversations in the same RTP packet.

In 1998, the Audio/Video Transport Working Group of the IETF met to discuss different proposals for RTP multiplexing. There were different points of view, and many proposals. Finally, IETF approved in 2005 TCRTP as RFC 4170 [4], with the category of "Best Current Practice". We will explain it with more detail in next section, and also other proposals and drafts that are not standard solutions.

Multiplexing can be a good solution for systems where many voice flows share the same path, for example, an IP telephony system between different offices of the same enterprise. In this scenario, there exists the possibility of having simultaneous calls between two extremes in the same networks (Fig. 1). In this case, the office's router can act as a multiplexer. There is also another advantage: enterprises often have VPNs that protect traffic between offices. So these tunnels could also be used in order to transport VoIP calls.

But multiplexing has also some disadvantages, as we will see: it introduces new delays and processing charge, which are added to the ones caused by router buffers. Also the added delays can affect to final de-jitter buffers. Another problem is that when a packet is lost, all the multiplexed calls become affected. So there is a tradeoff: the more the number of multiplexed streams, the more the bandwidth efficiency, but with bigger delay and packet loss. These parameters are important in order to quantify customer experience, commonly estimated using the ITU E-Model [5]. R-factor is a measure which ranges from 0 (bad quality) to 100 (high quality). Medium quality is considered from R > 70. This work presents a study of different multiplexing schemes and how they affect user's perception of voice signal, depending on different router buffer policies.

This paper is organized as follows: section II discusses the related works about RTP multiplexing proposals and uses. The test methodology and the most significant parameters are presented in section III. The next section covers the measurements and results. The last section details the conclusions of the present work.

II. Related Work

A. RTP multiplexing proposals

RTP specification [6] includes the concept of translators and mixers. A mixer is an entity that receives stream from different sources, possibly changes the data format and forwards the combined stream. For example, in a multi-conference some voice streams can be added into one, and

978-1-4244-8788-2/11/$26.00 ©2011 IEEE

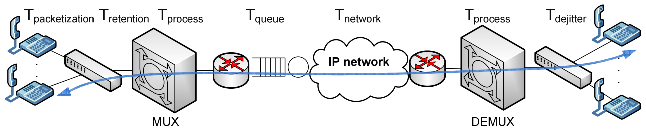

Figure 1. Multiplexing scheme.

then retransmitted to the receiver. A translator can re-encode some packets into one. But finally both of them send only a combined or translated RTP flow. In contrast, a multiplexer receives data from a number of sources and sends them all, combining them in the same packet.

A multiplexer-demultiplexer system has to be transparent for the communication ends: the packet sent from the origin and the packet received at the end have to be exactly the same, so the demultiplexer needs information in order to rebuild the original packet and deliver it to its destination.

We will first explain TCRTP [4]. This standard does not define a new protocol, but combines some of them. Its protocol stack can be seen in Fig. 2 [3]. First, ECRTP, which is a header compression scheme, compresses IP, UDP and RTP headers into a new header. Next, PPP multiplexing is used and finally the packets are sent with a PPP and L2TP tunnelling scheme. The use of a tunnel makes it possible to use ECRTP end-to-end, avoiding the need of being applied on each router of the path.

Another non standard option was presented by Sze et al. [7]. As can be seen in Fig. 3, it includes a number of RTP packets in a single UDP packet. The RTP headers are compressed, so context-mapping tables are required in the multiplexer and demultiplexer in order to rebuild the original RTP packets. From now, we will refer to this multiplexing scheme as *Sze.* In [8] a similar solution had been proposed but without the compression of RTP headers.

There exist other multiplexing proposals in the literature. For example, [9] presented a system that adapts the throughput in response to congestion; GeRM [3] proposed the idea of including multiple RTP payloads, each one with a compressed header, into a single RTP packet; ref. [10] assembles audio samples from different users into an RTP payload, using a 2 byte MINI-Header in order to identify users. Nevertheless, we will use TCRTP and *Sze* proposals, as we consider that they include the most representative ideas of RTP multiplexing.

B. RTP Multiplexing Uses

Multiplexing reduces both bandwidth and the number of packets. Based on measurements of commercial routers, [13] discovered that, in certain conditions, the maximum call load is bounded by the router capacity rather than the link capacity, i.e. the number of packets per second a router can manage is limited. So they recommended the consideration of both packet throughput and bit throughput. Multiplexing schemes certainly reduce the number of packets, alleviating the router's workload as they divide it by a factor of *k*. For the rest of the paper, *k* represents the number of multiplexed flows.

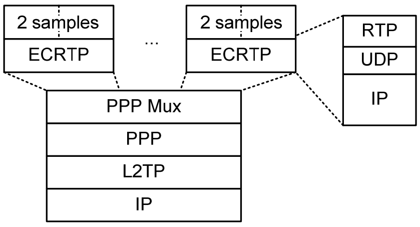

Figure 2. TCRTP protocol stack.

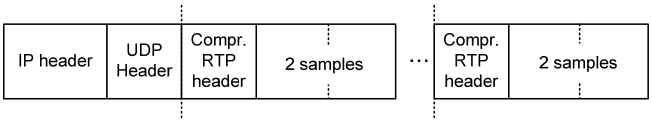

Figure 3. Alternative compression scheme (*Sze*).

VoIP applications are being widely used, and bandwidth consumption is a concern for researchers. We have found some works in the literature interested on RTP multiplexing. Ref. [11] presents a study of an adaptive multiplexing system based on E-model. They mainly use multiplexing in order to decrease overhead caused by IPSec.

Ref. [7] presented a system that multiplexes RTP packets in one UDP. Each RTP packet has a mini-header. They use some tables which are necessary to rebuild packets at the demultiplexer. They also study the delays that appear when a multiplexing scheme is used. They conclude that multiplexing can increase bandwidth efficiency by as much as 300%.

Multiplexing has also been proposed in other levels. In [12] two different systems are compared: sample application-layer aggregation, in which many RTP samples are included in a RTP packet, and the use of a performance enhancing proxy at IP level, putting together complete VoIP packets from multiple flows.

The present work not only considers the bandwidth reduction achieved by multiplexing, but also evaluates the advantages of multiplexing schemes, in terms of conversation quality, taking into account the effect of router buffer policies.

C. Buffer size and buffer policies

Using certain buffer policies, the packet size may have an impact in the percentage of discarded packets. As multiplexing increases the packet size, big packets are expected to be discarded in a bigger percentage than small ones. On the other hand, multiplexing saves bandwidth, so, with the same background traffic, multiplexed RTP may have better results than simple RTP. This is the reason why we have carried out measurements with different buffers. Our main objective is to know in which cases multiplexing represents an improvement to the QoS.

For many years, the "rule of the thumb" for buffer sizing was the use of "Bandwidth-Delay Product" (BDP), i.e. the buffer size should be the product of the RTT (Round Trip Time) and the capacity of the link [14]. But in the last years, Appenzelder et al. [15] proposed the so-called "Stanford Model", which reduces the buffer size by dividing it for the square root of the number of TCP flows. After that, there has appeared a lot of literature studying optimal buffer sizes. Most of them are centered in the study of the behavior of a number of TCP flows in core routers. Ref. [16] presents a comparative study of different buffer size policies. One of them is the approach that controls the maximum queuing delay at the target link. In this paper we will use this approach, as VoIP has very strict time constraints, taking into account that delay is one of the parameters that determine R-factor [17]. In the present study we will use this buffer policy, comparing it with a very big buffer with a simple FIFO policy, and also with the case of a dedicated bandwidth for VoIP traffic.

III. Test Methodology

A. *Traffic generation*

Background traffic is generated with D-ITG [18]. We have used the next distribution: 50% of the packets are of 40 bytes, 10% of the packets are of 576 bytes, and the rest 40% are of 1500 bytes [19]. UDP has been used instead of TCP, in order to avoid flow control, thus obtaining always the same traffic. VoIP traffic is also generated using D-ITG, which permits different statistics in both inter-packet time and packet size.

Multiplexed RTP traffic has been characterized with statistical models in order to obtain a realistic behavior. Ref. [20] presents a comparison of CRTP and ECRTP for VoIP applications over satellite links. The obtained values show that for ECRTP, 97.3% of the packets have a COMPRESSED_ RTP header, while 2.6% have a COMPRESSED_UDP one. The percentage of FULL_HEADER packets is very small (0.0033 %), and we will consider it negligible. We have modeled TCRTP's behavior in terms of packet size, adding the correspondent number of extra bytes for each COMPRESSED _UDP packet, according to a binomial distribution depending on the number of multiplexed packets *k*. These extra bytes correspond to a timestamp and absolute IPID, which have to be updated. We have used the same statistics to model the behavior of *Sze*, in order to compare the two multiplexing methods in the same conditions.

For each measurement, 400 seconds of real traffic have been sent in a scenario similar to Fig. 1. Later, the first and last 20 seconds have been discarded in order to get a stationary behaviour. No silence suppression has been used.

RTCP is a protocol that works with RTP, but in [6] it is said that its traffic must not exceed 5% of RTP traffic. This is the reason why in this paper we will not consider RTCP multiplexing, i.e. RTCP will work normally between the extremes of the communication.

B. *System delays*

We will summarize the different delays that have to be considered in our system. They are illustrated in Fig. 1:

- Packetization delay: It depends on the codec. In this work we always use G.729a with two samples per packet, so delay will be 25 ms: 10 ms for each sample and 5 ms corresponding to the look-ahead time.

- Retention time: The multiplexer has to wait in order to receive one packet from each RTP font. In this study we will assume that the RTP sources are connected to a high speed LAN, so retention time can be considered equivalent to the time between packets (20 ms), as an upper bound.

- Process time in the mux/demux: Ref. [7] built a software prototype of their multiplexing scheme, running under Linux. They observed that the processing mux/demux times caused by packet transmission and header manipulation were below 1 ms. As the packets are bigger when multiplexing, store & forward delay will be increased a little. In this work we will add 5 ms in order to take into account processing time in the mux and demux, and also store & forward and propagation times in local networks.

- Queuing delay at the origin router's buffer: The pass from a high speed LAN to the Internet access network supposes a bottleneck that has to be taken into account. This delay will strongly depend on the buffer policy implemented at the router.

- Network delay: The packet arrival times are captured after the router, and then a different network delay is added to each packet, using a statistical distribution. We have used the model proposed in [21], which is based on the results of some global measurement projects [22]. The model consists of a fixed minimum delay depending of the geographical distance between the two nodes, plus a log-normal distributed delay that is applied to each packet. In our case we have considered an intra-region scenario, and we have used values extracted from [23]: 20 ms of minimum One Way Delay (OWD), and for the log-normal distribution, the average was 20 ms with a variance of 5. We have not considered the network to increase packet loss.

- Queuing delay at the destination router's buffer: It is considered negligible, because we are passing from an Internet access to a high speed LAN.

- De-jitter buffer of the destination application: It adds a new delay and also increases packet loss, as every packet that does not arrive in time to be reproduced will be equivalent to a lost packet. As we want to avoid the use of a concrete implementation, de-jitter buffer losses have been calculated by an approximation suggested by Cole et al. in [24]:

$$loss_{\text{de-jitter buffer}} \sim \mathrm{P}\ \{\ l > bg\} \qquad (1)$$

Where l is the difference between OWD of consecutive packets, b is half of the buffer size, and g represents inter-packet generation time. This approximation supposes a static buffer. By the use of adaptive schemes, better results can be obtained. This is the reason why in some graphs we have included values of R-factor smaller than 70. De-jitter buffer size has been calculated in each case to maximize R-factor, obtained with the analytical expression also proposed in [24].

IV. Results

As we have previously said, three different router buffer policies are being tested. For each one of them, we will compare simple RTP with two multiplexing schemes: TCRP and *Sze*. In the figures we will also include a graph called "1 RTP", which represents the behaviour of a single RTP flow. It can be a help in order to compare the results with the best case.

The queuing delays at the origin router's buffer have been obtained by the use of a testbed [25] that emulates the queues using *tc* Linux tool. It permits to set up different buffer sizes with some parameters as latency limit, buffer limit or the size of bursts. *tc* takes into account level-2 headers (Ethernet in our case) to calculate bandwidth limit, so traffic amounts have to be properly corrected.

We will compare multiplexed schemes with RTP, but not with CRTP or ECRTP, because these protocols operate link by link, so they are not adequate for our Internet scenario.

A. Dedicated buffer

First, we will see what happens if a bandwidth is reserved to VoIP packets. We will send different number of RTP flows using a *tc* limit of 200 kbps dedicated bandwidth. So we can expect the system to behave well while the VoIP bandwidth is smaller than the limit. Fig. 4 shows R-factor as a function of the number of flows *k*. It can be seen that using simple RTP only 6 flows are supported, while TCRTP supports up to 17 and *Sze* maintains R above 70 until 20 flows. The overhead of the two multiplexing schemes is shared by all the flows, but in the case of simple RTP the bandwidth simply increases by a factor of *k*. As TCRTP uses a tunnel, its bandwidth is bigger than *Sze's* one and it can support less flows.

B. Big buffer

Next, we will study multiplexing behaviour when a big buffer is used. We will assume a single FIFO buffer with a very big size. In our case we have used an 800 ms limited queue. So if bandwidth limit is reached, it will grow indefinitely, delaying packets above the required limits for VoIP. The bandwidth limit in this case is 1 Mbps. Fig. 5 shows R-factor as a function of background traffic, in the case of 15 VoIP flows. It can be seen that the behaviour is similar to the one obtained with dedicated bandwidth: when the limit is reached, R-factor gets unacceptable.

C. Time-limited buffer

Finally, a time-limited buffer has been tested. We consider it interesting because in real-time and interactive services, like VoIP, the delay has to be maintained under a limit in order to provide a service similar to traditional telephony.

The connection bandwidth is 1 Mbps. The buffer has only one queue and every packet that spends more than 80 ms in it is discarded, so big packets have more probability of being dropped than small ones. This is an advantage for voice packets, as they are small, but it is a disadvantage for multiplexed packets, as they are bigger than non-multiplexed ones, and they will be dropped in a bigger percentage. With this buffer policy R-factor is expected to go down slower than with the others, as voice packets have this advantage. There are two simultaneous effects: multiplexing saves bandwidth, but at the cost of generating bigger packets and thus having more dropping probability. So it can be interesting to study in which cases one effect is more important than the other.

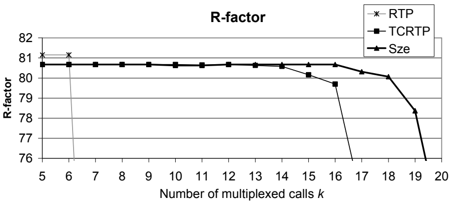

Figure 4. Comparative using 200 kbps of dedicated bandwidth.

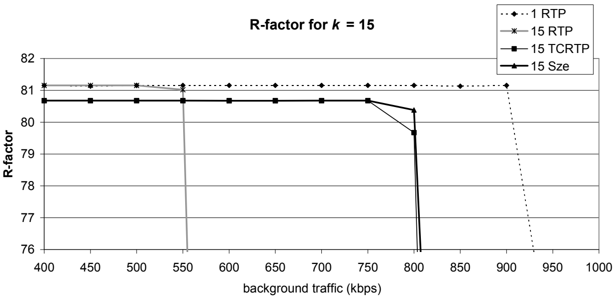

Figure 5. R-factor for big buffer.

Fig. 6 shows the comparative for k = 10. It can be seen that for small background traffics, simple RTP behaves a little better than multiplexing schemes. This is mainly caused by the additional delays introduced by multiplexing.

But when background traffic is above 650 kbps, RTP gets worse. This is mainly caused by the bigger bandwidth used by RTP, that makes the total bandwidth get near 90% of the limit, while TCRP and *Sze* are using only 114 kbps and 99 kbps respectively, as can be seen in Table I.

When background traffic is 95% of the limit, it can be seen that simple RTP again achieves a better result than multiplexing. The cause is that multiplexed packets are dropped in a bigger percentage due to their size. But in Fig. 7 we can see that in that case background traffic has a worse loss ratio with simple RTP than with multiplexing schemes.

The behavior of *Sze* is a little better than TCRTP, because it does not use a tunnel. But *Sze* is not a standard protocol and requires the addition of some elements to work properly. The use of a tunnel in TCRTP avoids that elements, and uses the protocols according to the standards.

Finally, Fig. 8 shows the R-factor improvement when using different number of multiplexed flows with respect to simple RTP. It can be seen that above 5 multiplexed flows, the use of multiplexing is a good improvement, gaining up to 21%. For small traffics, the impairment of multiplexing is below 1 %. If background traffic is above 90%, the behaviour gets worse.

V. Conclusions

This work studies different multiplexing schemes have been tested and compared with simple RTP in terms of ITU R-factor, using three different buffer policies. On one hand, the use of RTP multiplexing requires less bandwidth but, on the other hand, it introduces new delays, i.e. retention time and also small processing times in both sides of the communication, and also packet size: Multiplexed packets are bigger than RTP ones, and this could increase their probability of being discarded, depending on buffer policies.

It has been found that in certain conditions multiplexed RTP can obtain better results than simple RTP. The use of a tunnel in case of TCRTP does not suppose an important drawback, although its use implies some bandwidth cost.

The obtained results show that multiplexing and buffer policies are a good way to search algorithms and solutions able to improve customer experience of VoIP.

TABLE I

BANDWIDTH OF RTP, TCRTP AND SZE AT IP LEVEL IN KBPS

Number of calls	5	10	15	20
RTP	120	240	360	480
TCRTP	62	114	166	216
Sze	55	99	143	185

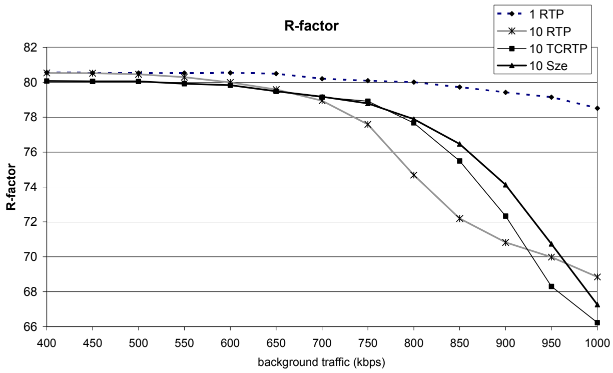

Figure 6. R-factor with k = 10 multiplexed calls

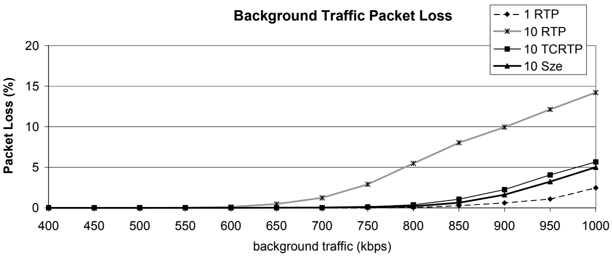

Figure 7. Packet loss for background traffic with k = 10

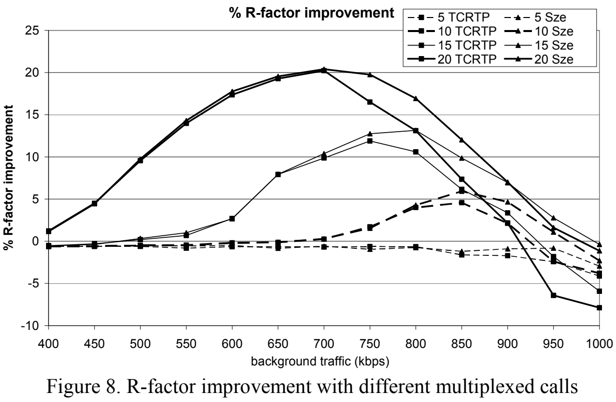

Figure 8. R-factor improvement with different multiplexed calls

ACKNOWLEDGMENT

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.

REFERENCES

[1] S. Casner et al. *RFC 2508: "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links",* February 1999.

[2] T. Koren et al. *RFC 3545: "Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering",* July 2003.

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

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

[5] *"The E-model, a computational model for use in transmission planning",* ITU-T Recommendation G.107. Mar. 2003.

[6] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. *RFC 3550: "RTP: A transport protocol for real-time applications",* July 2003.

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

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

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

[11] J. Yu, I. Al-Ajarmeh. *"Call Admission Control and Traffic Engineering of VoIP",* In Proc. Second Intenational Conference on Digital Telecommunications, IEEE ICDT 2007.

[12] 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.

[13] K. Pentikousis, E. Piri, J. Pinola, F. Fitzek, T. Nissilä, I. Harjula. *"Empirical evaluation of VoIP aggregation over a fixed WiMAX testbed"*. In Proc. 4th international Conference on Testbeds and Research infrastructures For the Development of Networks & Communities. Innsbruck, Austria, Mar. 2008.

[14] C. Villamizar, C. Song. *"High performance TCP in ANSNET".* ACM Computer Communication Review, Oct. 1994.

[15] G. Appenzeller, I. Keslassy, and N. McKeown. *"Sizing router buffers"*, In SIGCOMM '04, pages 281–292, New York, USA, 2004. ACM Press.

[16] A. Dhamdhere and C. Dovrolis, *"Open issues in router buffer sizing",* Comput. Commun. Rev., vol. 36, no. 1, pp. 87–92, January 2006.

[17] *"One-way transmission time".* ITU-T recommendation G.114. Feb. 1996.

[18] A. Botta, A. Dainotti, A. Pescapè, *"Multi-protocol and multi-platform traffic generation and measurement"*, INFOCOM 2007 DEMO Session, Anchorage, Alaska, USA, May 2007.

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

[20] G. Dimitriadis, S. Karapantazis, F.-N. Pavlidou, *"Comparison of Header Compression Schemes over Satellite Links"*, In Proc. International Workshop on IP Networking over Next-generation Satellite Systems (INNSS'07), Budapest, Hungary, Jul 2007.

[21] 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.

[22] CAIDA. Macroscopic Topology Project, http://www.caida.org/.

[23] AT&T Global IP Network, http://ipnetwork.bgtmo.ip.att.net/pws/

[24] R.G. Cole, J.H. Rosenbluth. *"Voice over IP performance monitoring".* SIGCOMM Comput. Commun. Rev. 31, 2 (Apr. 2001), pp. 9-24.

[25] J. Saldaña, E. Viruete, J. Fernández-Navajas, J. Ruiz-Mas, J. I. Aznar. *"Hybrid Testbed for Network Scenarios".* SIMUTools 2010, the Third International Conference on Simulation Tools and Techniques. Torremolinos, Spain. Mar. 2010.