Influence of the Distribution of TCRTP Multiplexed Flows 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-TCRTP is a multiplexing scheme for RTP, defined by IETF, which reduces the overhead of real-time flows. This work presents a preliminary study of how the number of multiplexed flows in TCRTP can affect the Quality of Service (QoS) of VoIP in terms of R-factor. Two different buffer policies are considered. The conclusion is that multiplexing all the calls in a single TCRTP flow is not always the best solution for certain buffer policies. The reason is that higher packet sizes can increase loss probability.

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

I. INTRODUCTION

The use of RTP for interactive real-time purposes usually has very poor bandwidth efficiency, as the number of samples that can be included in a RTP packet is small, due to time constraints. IP, UDP and RTP headers have to be included in every packet, and they add 40 bytes to the packet size. In November 2005, IETF approved TCRTP in RFC 4170 [1] as "Best current practice". Its main objective was to save bandwidth in scenarios where a number of RTP flows have the same origin and destination. The inclusion of samples from different flows avoids the delays derived from sending a big number of samples of one call in the same packet.

IETF has also defined some header compression protocols, like CRTP, ECRTP and ROHC, which try to reduce header fields that are similar or have constant variations, using delta compression and other techniques. They have to maintain a "context" at the origin and the destination, and they only work hop-by-hop. TCRTP uses ECRTP to reduce headers, and then it creates a tunnel by means of PPP Mux, PPP and L2TP, allowing the use of end-to-end header compression.

When there is a real-time communication between two local networks which have commercial Internet accesses, the behaviour of the routers' buffers may have a big influence in the delays and losses experienced by RTP traffic. Many routers have a high capacity buffer which, in case of saturation, can add unacceptable delays. In [2] a time-limited buffer was suggested, which drops packets spending more than a certain time in it. This policy is adequate to limit end-to-end delays, but it has also the effect of increasing losses for big packets, as they have more probability of being discarded.

TCRTP packets are bigger than RTP ones, as many flows are multiplexed. So we will have two effects: on one hand, TCRTP saves bandwidth, thus it is good for avoiding losses and delays. However, on the other hand, the increase in packet size may harm Quality of Service (QoS).

II. MEASUREMENTS, TESTS AND RESULTS

We have used VoIP to make a preliminary study of how grouping a different number of TCRTP flows modifies QoS. In the measurement diagram (Fig. 1) 40 calls using G729a codec with 2 samples per packet, are sharing the same link between two extremes. We will study the variation on ITU's R-factor [3] with different values of k, which is the number of multiplexed flows of a TCRTP tunnel, and l, the number of tunnels, satisfying the equation $l \times k = 40$ calls.

First, a testbed is used to send both desired and background traffic through the same 2 Mbps link, using two different buffer policies: high capacity buffer and time-limited buffer. The background traffic distribution is: 50% of the packets are of 40 bytes, 10% of them are of 572 bytes, and 40% of 1500 bytes [4]. Table I shows the required bandwidth for different values of k, and also the average packet size at IP level.

The percentage of compressed headers has been obtained from [5]. A binomial distribution is used to calculate the number of compressed headers on each TCRTP packet.

The traffic is captured, an then a fixed network delay of 20 ms and a lognormal distributed delay of 20 ms average and a variance of 5 have been added offline [6]. Playout buffer delays (40 ms) and losses are finally added.

Fig. 2 shows the R-factor obtained for different values of k, using a high capacity buffer. Its behaviour is good until the bandwidth limit is reached, and then it falls dramatically. In this case, the best behaviour is obtained using $k= 40$ and $k= 20$.
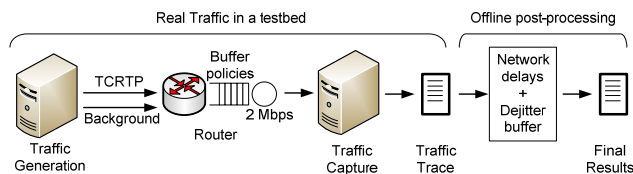

Figure 1. Measurement diagram.

TABLE I

AVERAGE PACKET SIZE AT IP LEVEL (IN BYTES), AND BANDWIDTH (IN KBPS)

$l \times k$	1×40	2×20	4×10	5×8	8×5	No mux
avg. packet size (bytes)	1073	549	287	234	156	60
Bandwidth (kbps)	429	439	459	469	499	960

Fig. 3 is similar to Fig. 2, but we have used a time-limited buffer, which discards packets that spend more than 80 ms on it. The advantage of this buffer policy with respect to the high capacity one is that the slope of the curves is lower, so an acceptable conversation quality (R > 70) can be achieved with more background traffic. The behaviour of simple RTP is very bad for these values of background traffic, as offered traffic is over the bandwidth of the link. The graph of 40 RTP flows (no multiplexing) has also been included.

To clarify, we have included the most interesting values of Fig. 3 in Table II, i.e. the ones for 1500 and 1600 kbps of background traffic: the best behaviour is obtained with $l = 2$ flows of $k = 20$ multiplexed calls, which is very similar to $l = 4$ flows of $k = 10$ calls.

Fig. 4 shows the percentage of improvement in R-factor that can be obtained by multiplexing, respect to the values obtained with simple RTP. This figure illustrates the fact that multiplexing all the calls in a single tunnel is not always the best solution. It depends on buffer policies.

On one hand, multiplexing all the calls in one tunnel ($k = 40$), will save more bandwidth than any other solution. But, on the other hand, packets will be bigger, so the probability of being discarded by the buffer increases.

It may also be noticed that the use of big values of k helps to save background traffic from being lost. As it can be seen in Fig. 5, the bigger the number of multiplexed calls, the smaller the packet loss percentage for background traffic. So there is a tradeoff: if we want to prioritize voice traffic, we will have to use the value of k that maximizes R-factor. But if we want to save bandwidth, we will multiplex all the calls, achieving the best performance for background traffic.
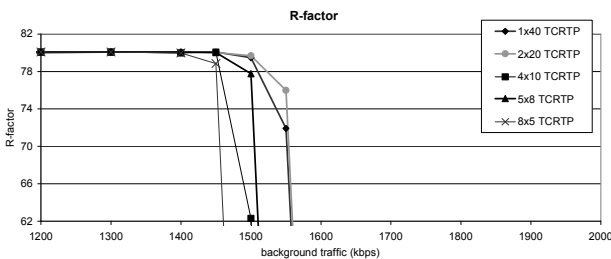

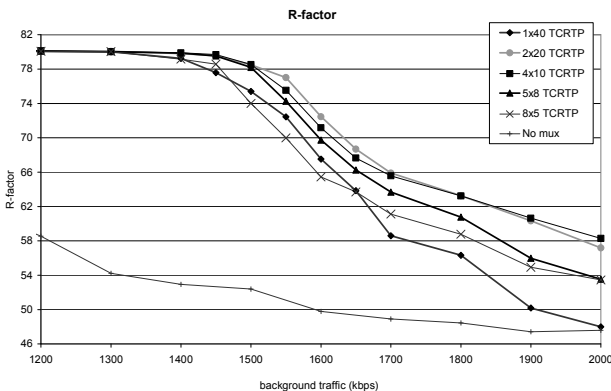Figure 2. R-factor for different values of k with high capacity buffer.

Figure 3. R-factor for different values of k with time-limited buffer of 80 ms.

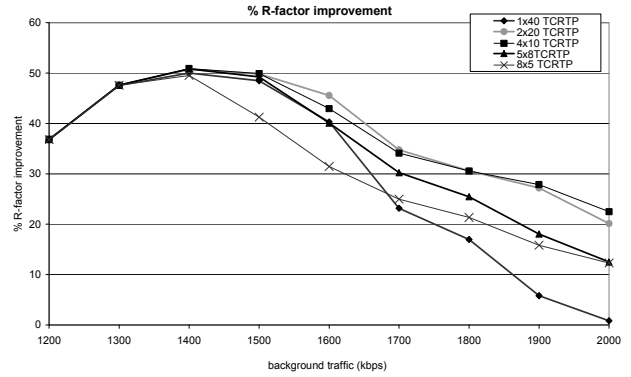| l x k	1 x 40	2 x 20	4 x 10	5 x 8	8 x 5	No mux
BG. 1500 kbps	72,44	77,01	75,54	74,26	69,99	52,39
BG. 1600 kbps	67,51	72,46	71,16	69,74	65,44	49,78

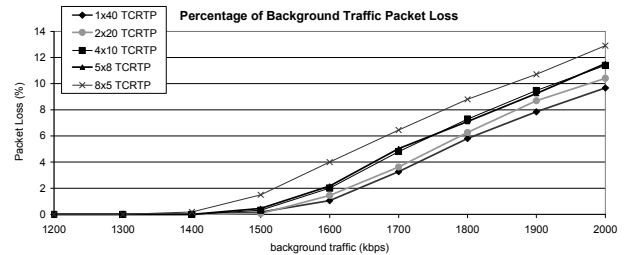Figure 4. Percentage of R-factor improvement with respect to 40 RTP flows

Figure 5. Percentage of background packet loss with different values of k

Our group is currently working in a dynamic algorithm, capable of calculating the value of k in order to obtain the best performance in each case, taking into account the kind of traffic, the buffer policies and time constraints.

REFERENCES

[1] B. Thompson, T. Koren, D. Wing. *RFC 4170: "Tunneling Muliplexed Compressed RTP (TCRTP)"*, November 2005.
[2] A. Dhamdhere and C. Dovrolis, *"Open issues in router buffer sizing"*, Comput. Commun. Rev., vol. 36, no. 1, pp. 87–92, January 2006.
[3] *"The E-model, a computational model for use in transmission planning"*, ITU-T Recommendation G.107. Mar. 2003.
[4] Cooperative Association for Internet Data Analysis *"NASA Ames Internet Exchange Packet Length Distributions"*.
[5] 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.
[6] 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.