Can We Multiplex ACKs without Harming the Performance of TCP?

Jose Saldana, Julián Fernández-Navajas, José Ruiz-Mas
Communication Technologies Group (GTC), Aragon Institute of Engineering Research (I3A)
EINA, University of Zaragoza
Zaragoza, Spain
{jsaldana, navajas, jruiz}@unizar.es

***Abstract*— This paper explores the application of traffic optimization techniques, based on header compression and multiplexing, to the flow of ACKs sent by a TCP client when downloading a file. Significant compression rates can be obtained, but a multiplexing delay has to be added as a counter-part. The influence of this delay in TCP dynamics is explored by means of simulations including FTP flows sharing a bottleneck. The results show that the impairments can be maintained in reasonable limits, by selecting proper values of the multiplexing period. Thus, the multiplexing of these ACK flows can be further studied and considered as a possibility for traffic optimization.**

***Index Terms*—traffic optimization; multiplexing; TCP**

I. INTRODUCTION

The increase of emerging real-time services using small packets is modifying the traffic mix present on the Internet [1]. These services make an inefficient use of the network, especially when the header and the payload length are in the same order of magnitude. This problem is stressed for IPv6, which header is twice as big as that of IPv4.

Different mechanisms have been proposed in order to improve network efficiency. Specifically, Tunneling Multiplexed Compressed RTP (TCRTP) [2] applies header compression and multiplexing to a number of VoIP flows sharing a common network segment. Packets from different flows can be included in the same bundle, which is transmitted using a tunnel between the multiplexer and the de-multiplexer. Thus, the amount of packets can be significantly reduced, and bandwidth can be saved at the same time. This can be important in wireless or satellite links, where the limit is given by a fixed amount of packets per second which can be transmitted. In addition, a lower amount of packets per second will reduce energy consumption in network devices since packet processing and switching fabric require 60% and 18% of the router power consumption respectively [3].

A proposal for widening the scope of TCRTP [4] is currently being discussed within the IETF [5], with the aim of considering not only VoIP, but also other flows consisting of small packets. As a counterpart of the savings, a small multiplexing delay is required in order to get a number of packets to be sent together. This delay will vary between 0 and the multiplexing period *PE*, with an average value of *PE*/2 [6].

TCP ACKs are a very particular case of small packets: they have no payload, but they are necessary to control the dynamics of TCP. The amount of ACKs present on the Internet is huge, taking into account that e.g. video traffic mostly uses TCP, and this service is expected to account for 69% of all consumer Internet traffic in 2017 [7]. When a user is downloading a video using a DSL of 5 to 10 Mbps, his/her computer may generate some hundreds of ACKs per second, during some tens of seconds. Thus, in some scenarios (e.g. the aggregation network of an operator) certain points where high numbers of long-term flows of ACKs share a common path can be found. In addition, the header compression ratio of ACKs without payload can be really high: according to [8], an ACK can be compressed from 40 to 7 or 8 bytes (savings of 80%).

All in all, in this paper we want to give a first answer to this question, which has been discussed within the IETF [5]: is it a good idea to compress and multiplex these flows? Would the multiplexing delay degrade the performance of TCP, as it may happen with “delayed ACK” [9]?

II. TESTS AND RESULTS

We have created an ns2 dumbbell scenario (Fig. 1) where two FTP downloads (from A’ to A, and from B’ to B) share a bottleneck N-M. In this scenario, the ACKs travelling from B to B’ experience a multiplexing delay of period *PE* in the link P-N. This simulates the existence of a network segment in which this flow of ACKs would be multiplexed and compressed in combination with other similar flows. The bandwidth is 10 Mbps in the bottleneck, and 100 Mbps in the rest of the links; the OWD in A-A’, and B-B’ is 40 ms.

In order to show the influence of multiplexing on the temporal evolution of the throughput, we have run two different simulations using TCP *Tahoe* (we have used the most basic TCP variant, to observe the effects more clearly). First, Fig. 2.a shows the throughput obtained by a single FTP download from A’ to A. The normal evolution of TCP can be observed: it keeps on increasing the sending window until congestion is reached, and then the window is reset to 1. As a

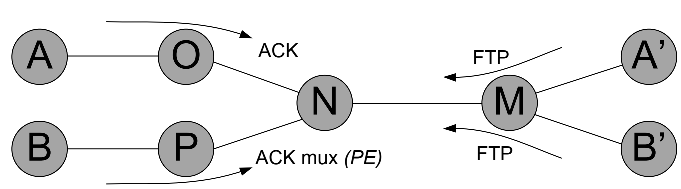


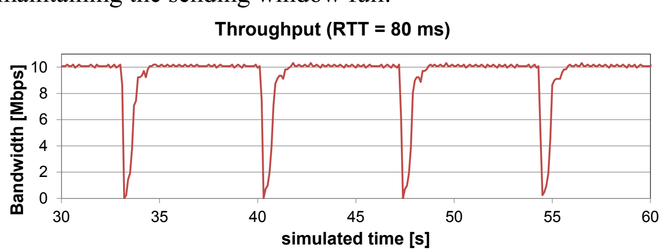
Fig. 1. Simulations scheme

result, the average throughput is 9.24 Mbps. In contrast, Fig. 2.b shows the behavior of a B'-B download when a high value of the multiplexing delay (*PE*=50ms) is added to the ACKs. It can be observed that the interval between windows resets gets increased from 7 to 9.5 sec, and we can also see that the throughput presents some oscillations during its rise. The cause of this is that the arrival of ACKs to the sender becomes bursty, thus triggering a number of packet departures. Overall, the average throughput of the isolated flow falls to 8.039 Mbps, i.e. a difference of 1.2 Mbps (12 %) with respect to the case where no multiplexing is applied in the network.

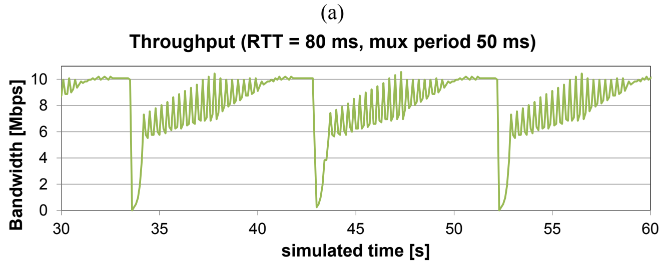
Another battery of tests has been deployed, with the aim of studying the competence of multiplexed and non-multiplexed flows. In this case, two FTP flows share the bottleneck: A'-A is not multiplexed, and B'-B uses different values for the period *PE*. Table I summarizes the throughput difference between the two flows, computed this way: for example, if A'-A obtains 5.6 Mbps, and B'-B obtains 4.2 Mbps, the difference is reported as 14 % (of the total bottleneck bandwidth). Four TCP variants are used. Each test lasts 1000 seconds (the average is computed over the last 500).

It can be observed that the throughput difference is small when *PE* is 5 ms, and it can be maintained in acceptable levels (between 10 and 20 %) using 10 ms of period. These period values are feasible for certain scenarios, e.g. in an aggregation network it is possible to find 100 flows of ACKs (100 pps each) sharing a common segment. In this case 50 packets could be multiplexed using a period of 5 ms. However, when higher values of the period are used, then the difference becomes more significant (over 25 %, and up to 60 % in some cases).

The differences observed for TCP *Reno* are higher than those of *Tahoe*, since *Reno* uses *fast recovery*, thus dividing by 2 the sending window in A'-A in many cases, whereas B'-B frequently resets the window size to 1. This difference is reduced in the most popular TCP variants nowadays, i.e. *New Reno* and *SACK,* because of the mechanisms they use for maintaining the sending window full.



(a)



(b)

Fig. 2. Instantaneous bandwidth generated by a single FTP flow a) RTT=80ms; b) RTT=80ms, plus a multiplexing delay with a period of 50 ms

TABLE I. THROUGHPUT DIFFERENCE (PERCENTAGE) BETWEEN A MULTIPLEXED AND A NON-MULTIPLEXED FLOW SHARING THE BOTTLENECK

	Multiplexing Period *PE* [ms]				
TCP	***5***	***10***	***15***	***20***	***25***
Tahoe	4.91 %	10.05 %	31.67 %	7.88 %	49.74 %
Reno	5.95 %	17.78 %	48.62 %	24.29 %	61.92 %
New Reno	4.82 %	12.95 %	30.52 %	16.70 %	52.87 %
SACK	2.27 %	12.70 %	20.62 %	14.75 %	50.90 %

III. CONCLUSIONS

This paper has studied the suitability of the application of traffic optimization techniques, based on header compression and multiplexing, to the flow of ACKs sent by a TCP client when downloading a file. The expected bandwidth savings are huge because of the absence of payload in these packets. As a counterpart, we have observed a throughput reduction and a disadvantage when an optimized flow shares a bottleneck with a non-optimized one. This disadvantage can become unacceptable if high values of the multiplexing period are used. However, these impairments can be maintained in tolerable limits, by setting an upper bound on the multiplexing period. Thus, future work will further study the terms of this trade-off and its potential benefits for reducing bandwidth and the amount of packets per second. Other effects of multiplexing should also be studied, e.g. the influence of packet size increase on packet loss.

IV. 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; Ibercaja Obra Social; Project of Catedra Telefonica.

REFERENCES

[1] Huawei, White Paper "Smartphone Solutions" 2.0, 2012-07-17. Available online: www.huawei.com/ilink/en/download/HW_193034

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

[3] R. Bolla, R. Bruschi, F. Davoli, F .Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," Communications Surveys & Tutorials, IEEE, vol.13, no.2, pp.223, 244, 2011.

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

[5] TCMTF BoF, IETF 87 minutes (Berlin, Germany, Jul 2013). Available online: http://www.ietf.org/proceedings/87/minutes/minutes-87-tcmtf

[6] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, E. Viruete, L. Casadesus, "Influence of online games traffic multiplexing and router buffer on subjective quality," IEEE Consumer Communications and Networking Conference (CCNC), pp.462,466, Jan. 2012.

[7] Cisco Visual Networking Index: Forecast and Methodology, 2012–2017. Available online: http://www.cisco.com/en/US/solutions/collateral /ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf

[8] 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.

[9] O. González, I. de Miguel, N. Merayo, P. Fernández, R. M. Lorenzo, E. J. Abril, "The impact of delayed ACK in TCP flows in OBS networks", Proc. of the 10th European Conference on Networks & Optical Communications (NOC'05), pp. 367-374, 2005.