TFRC contribution to Internet QoS improvement

Nicolas LARRIEU and Philippe OWEZARSKI

LAAS-CNRS
7, avenue du Colonel ROCHE
31077 TOULOUSE Cedex 4
FRANCE
{nlarrieu**, owe}@laas.fr

Abstract. The Internet is on the way of becoming the universal communication network, and then needs to provide various services and QoS for all kinds of applications. We show in this paper that oscillations that are characteristic of the Internet traffic provokes huge decrease of the QoS that flows can get. After having demonstrated that such oscillations can be characterized by the Hurst (LRD) parameter, we propose an approach for improving Internet flows QoS based on smoothing sending rate of applications. TFRC is a congestion control mechanism that has been issued for this purpose. This paper then proposes an evaluation of TFRC benefits on traffic profile and flows QoS.
Keywords. Internet monitoring, traffic characterization, quality of service, TFRC, congestion control for elephants

1 Introduction

The Internet is on the way of becoming the universal communication network for all kinds of information, from the simple transfer of binary computer data to the transmission of voice, video, or interactive information in real time. It has then to integrate new services suited to new applications. In addition, the Internet is rapidly growing, in size (number of computers connected, number of users, etc.), and in complexity, in particular because of the need of new advanced services, and the necessity to optimize the use of communication resources to improve the QoS[1] provided to users. In fact, the Internet has to evolve from a single best effort service to a multi-services network.

Since at least a decade, Internet QoS is then, one of the major issues in the Internet. Many proposals have appeared as IntServ, DiffServ, etc., but until now, they have not been deployed (or their deployment has been quite limited). Indeed, Internet community contributions to propose differentiated and guaranteed services did not provide the solutions users and operators (Internet service

** Corresponding author: Nicolas LARRIEU – Tel: (33)5-61-33-78-20 – Fax: (33)5-61-33-64-11

[1] QoS: Quality of Service

roviders, carriers, etc.) are expecting. There are always difficulties with the complexity of the Internet and all its network interconnections, with their resource heterogeneity in terms of technologies but also in terms of provisioning, and of course with the traffic characteristics. Indeed, because of the growing complexity of the Internet, all new applications with various and changing requirements, introduce in Internet traffic many characteristics that are very far from common beliefs. In fact, models with simple static metrics such as throughput, delay, or loss rate are really not sufficient to model completely and precisely Internet traffic dynamics that are its essential features. The evolution of the Internet is then strongly related to a good knowledge and understanding of traffic characteristics that will indicate the kind of mechanisms to deploy. Consequently, the development of monitoring-based tools and technologies to collect Internet traffics information, and methodologies to analyze their characteristics is currently an important topic for network engineering and research. In particular, the definition and quantification of Internet QoS is still not completely solved. First monitoring results showed that Internet traffic is very far from Poisson or Markovian models, used in telephony, and also reused as the model for Internet traffic as well. These first results showed that models that better represent Internet traffic are models with self-similarity or LRD[2] characteristics.

Given this previous work on traffic monitoring, our work showed also that Internet traffic has very significant oscillatory behaviors, whose peaks are responsible of some instability issues of the Internet QoS, as well as a serious decrease of Internet performances. This is especially true for big flows transporting a huge quantity of data (called "elephants"). That is why section 2 exposes the analysis results on some Internet links traffic characteristics and shows how oscillating phenomena can have such a bad impact on network QoS and performances. This analysis also indicates that TCP congestion control mechanism is largely responsible of such oscillations, what makes us propose some improvements for the Internet. More precisely, section 3 proposes to use a smoother transport protocol, at least for elephants, to separately smooth the flow behaviors (with a less aggressive congestion control mechanism), and explains how this individual optimization for each flow can bring important improvements for the whole network QoS. Some experiments assessing this approach, are presented in section 4. These experiments have been performed with the NS-2 [1] simulator, and allow the evaluation of the TFRC[3] congestion control mechanism. It is shown in this section that TFRC can optimize Internet QoS by smoothing its traffic. Finaly, section 5 concludes this paper.

Note however that this work is achieved in the framework of the METROPOLIS project, a French national project granted and funded by the Frenck Network for Research in Telecommunications. METROPOLIS main goal deals with issuing new network monitoring and analysis methodologies.

[2] LRD: Long Range Dependence
[3] TFRC: TCP-Friendly Rate Control

2 Traffic oscillation issues and elephant flows

Current Internet links monitoring results show the presence of very high oscillations in Internet traffic. An example of an Internet link traffic is given on Figure 1. This figure also compares current Internet traffic with a simple model of traffic: the Poisson model that is the model that was supposed to be the one of the Internet several years ago. In fact, traffic curves have to be smoother when the granularity of observation increases. This is what is represented in Figure 1 where for each traffic (actual Internet and simulated Poisson traffic) the amplitude of oscillations is decreasing when the observation granularity is coarser. What also appears on this figure is the difference between the two traffics: with coarse grain analysis, the oscillations amplitude of Internet traffic is much larger than Poisson traffic ones.

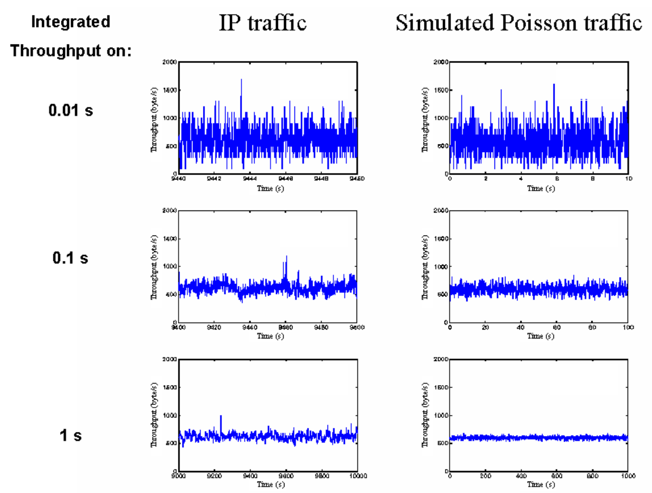

Fig. 1. Comparison between oscillations of Internet and Poisson traffics

Some analysis of Internet traffic performed in recent Internet monitoring projects showed that these oscillations are in fact the results of the presence of LRD and self-similarity in the traffic [2]. These phenomena are due to several causes and in particular to congestion control mechanisms, especially the ones of TCP that is the dominant protocol in the Internet [3]. Among these mechanisms, it is clear that the closed control loop of TCP introduces short scale dependence as the acknowledgment depends on the reception of one packet, and all the following packets of the flow depend on this acknowledgement. In the same way, the two TCP mechanisms – slow start and congestion avoidance – are responsible of introducing dependences between packets of different congestion control windows. And of course, this notion of burstiness in TCP sources plus the LRD explain oscillations in the global traffic. By extending this process, all packets of a flow are dependent from each other. As the increase of capacities in the Internet allows users to transmit larger and larger files (i.e. elephant flows[4]), as music or movies for instance, it is clear that the scale of LRD is increasing, explaining why oscillations of Internet traffic, even with a coarse observation granularity, are so high. Of course, oscillations are very damaging for the global use of network resources as the capacity freed by a flow after a loss for example cannot be immediately used by other flows: this corresponds to some resource waste, and of course a decrease of the global QoS of the traffic and network : the higher the oscillations amplitude, the lower the global network performance [4].

It is also clear that elephants introduce oscillations with higher amplitudes than mice (short flows). Indeed, elephants, because of their long life in the network, have time to reach large values of the congestion control window, and thus, any loss event can provoke a huge reduction, followed by a huge increase of the sending rate. This phenomenon is even more important in current Internet compared to what happened few years ago. Few years ago, Internet traffic consisted almost exclusively of web traffic with very short flows. Nowadays, because of the arrival of peer-to-peer applications used most of the time for huge files exchanges (as audio tracks or movies), Internet traffic consists of both web and Peer-to-peer traffic, meaning that there are more and more elephants and that elephants are getting larger and larger (essentially thanks to new high capacity Internet access technologies: ADSL, cable modem, etc.). Our past and current network monitoring results shows that elephants are now reaching more than 5 % of the number of flows in the Internet (it was 2 or 3 % few years ago), and that this 5 % of elephants represent around 60 % of the full Internet traffic.

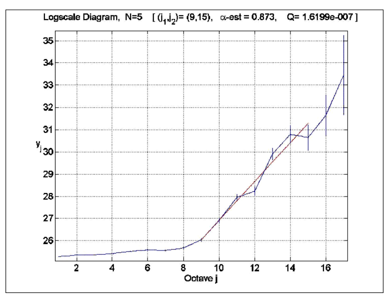

Fig. 2. LRD evaluation for edge network traffic

It is then clear that elephants and the huge oscillations they induce, directly impact traffic profile and also global network performances. Figure 2 represents the LRD evaluation of the network traffic depicted in Figure 1. This figure has been produced using the LDestimate tool designed by Abry and Veitch [5] [6] that

[4] In this paper, we define an elephant as a flow that contains more than 100 packets exchanged in the same mono-directional connection.

estimates the LRD that appears in Internet traffic at all scales. The principle of this tool relies on a fractal decomposition of traffic time series, what then allows users to have a graphical representation of the dependence laws at all time scales. Then small value octaves represent short range dependence, while large value ones represent long range dependence (LRD). In figure 2, we can note a "bi-scaling" phenomenon (cf. the elbow in Figure 2 around octave 8) which shows a difference in the LRD level between short and long time scales for the traffic exchanged. For short scale (octave < 8), representing the dependence between close packets (i.e. packets whose sending time are not very far from each other), the dependence is quite limited. Such dependence is the one that can exist for packets belonging to the same congestion window and that are then very close from each other. On the other side, for long time scales (octave > 8) LRD can be very high. For octaves 8 to 12, that correspond for instance to the dependence between packets of consecutive congestion windows, the dependence is higher. This can be explained by the closed loop structure of TCP congestion control mechanism in which the sending of one packet of a congestion control window depends on the receiving of the acknowledgement of one packet of the previous congestion control window. Of course, this phenomenon exists for consecutive congestion window, but also for all congestion windows of the same flow. This means, that the presence in the traffic of very long flows introduces very long scale dependence phenomenon, as depicted on figure 2 for very large octaves. The consequence of such LRD is one major issue as every oscillation at time t will be repeated at any other time t' that is dependent from t (because of the long range dependence between packets due to protocols – here TCP on long flows). That is why, if we want to both improve traffic profile and QoS, it is mandatory to decrease both LRD and oscillation levels for elephants. A solution for this is proposed in next section.

3 A new approach for improving Internet QoS

3.1 Increasing QoS by smoothing flow behaviors

It is clear now that network traffic has complex and high oscillating features. Indeed, it clearly appears the presence of scale laws in the traffic that induce the repetition of an oscillating phenomenon. This is especially visible on Figure 1. From this observation, it appears that the most urgent problem to address deals with reducing oscillations and more precisely with regulating the long term oscillations having such a damaging effect on traffic QoS and performance. Therefore, the main objective is then to bring more stability to elephants flows.

Such an approach is quite different from what can be proposed by classical service differentiation techniques. In general, classification of application flows depends on the QoS level they require, meaning that a web browser and a video streaming application are not in the same class: in this case the web browser is generally assigned a best effort service, while the streaming application get the best existing service (EF, gold or whatever name). This is an application oriented service selection, but that has the disadvantage of not taking into account network requirements. Our approach is basically based on a network centric point of view, and the classification proposed is based on the disturbance that flows induce on the traffic. Based on monitoring results, it appeared that elephants are the ones that introduce the more disturbances. Applications, as videoconferences, video on demand, telephony on IP, etc., that are typical applications requiring high quality services with classical service differentiation approaches, are also typical application generating long flows. In addition, such applications also require smooth services for smooth traffic. Our approach then perfectly fits the requirements of such applications. Of course, with our approach we are also going to smooth FTP or peer-to-peer long flows, that do not have the same requirements. Nevertheless, our approach introduces a big difference with application classes oriented approaches, as here, long and smooth flows that introduce few disturbances in the network are considered as low quality. This means that stream oriented applications are the applications introducing the less disturbances, and are the ones that should pay less. On the other side, applications that have sending rates oscillating a lot and that cannot be shaped or smoothed (as interactive video applications using MPEG[5]), are the ones that are considered as the most constraining and they will be charged more as the most disturbing. Note however that FTP or web traffic that is nowadays sent using the usual best effort service can easily use a smooth service thanks to its elastic nature. In both approaches, elastic traffic is the one that is the more flexible and then the one that is the easier to handle and then the cheaper.

To increase elephant flows regularity (i.e. to suppress observable oscillating behaviors at all scales), the new TFRC congestion control mechanism seems to be able to provide a great contribution. TFRC has been designed to provide a service suited for stream oriented applications requiring smooth throughputs. TFRC, then, tries as much as possible to avoid brutal throughput variations that occur with TCP because of loss recovery. Note however that for both TFRC and TCP, we will estimate the evolution of the oscillating behavior of the traffic by evaluating LRD features (also called the Hurst factor: H) on packet arrival series.

3.2 TFRC principles

TFRC aims to propose to applications a smooth sending rate with very soft increases and decreases; at least much softer than the ones of TCP. By associating such a congestion control mechanism to elephants, i.e. to the main part of the traffic, we expect to be able to control traffic oscillations, and then to increase global QoS and performance of the network. The sending rate of each TFRC source is made thanks to a receiver oriented computation, that calculates, once by RTT[6], the sending rate according to the loss event rate measured by the receiver [8] [9] according to equation 1:

[5] Because of the dependence induced between frames with this coding (P frames depends on previous frames and B frames on previous and next frames [7]).

[6] RTT: Round Trip Time

$$X = \frac{s}{R * \sqrt{2 * b * \frac{p}{3}} + (t_{RTO} * (3 * \sqrt{3 * b * \frac{p}{8}}) * p * (1 + 32 * p^2)))} \quad (1)$$

where:

- X is the transmit rate in byte/second,
- s is the packet size in byte,
- R is the round trip time in second,
- p is the loss event rate (between 0 and 1.0), of the number of loss events as a fraction of the number of packets transmitted,
- t_{RTO} is the TCP retransmission timeout value in second,
- b is the number of packets acknowledged by a single TCP acknowledgement.

In TFRC, a loss event is considered if at least one loss appears in a RTT. This means that several losses appearing in the same RTT are considered as a single loss event. Doing so, the loss dependence model of the Internet is broken since most dependent losses are grouped in a same loss event. Thus, the recovery will be eased and more efficient compare to what TCP can do: it is well known that TCP is not very efficient to recover from several losses in sequence. This approach follows the results of [10] that proposes an analysis and a model for the Internet loss process.

4 Evaluation of TFRC impact on QoS

4.1 Experiment description

Our experiment aims to provide a comparative evaluation of the global traffic characteristics if elephants use TCP or TFRC as the transmission protocol. This experiment aims to provide values in a realistic environment. For that, of course, the experiment relies on the use of traffic traces grabbed thanks to passive monitoring tools as the DAG [11] equipments. Therefore, traffic flows identified in the original traffic trace are replayed in NS-2 with the same relative starting date and the same others characteristics. Elephant flows are transmitted in the simulator using TFRC while others flows use TCP New Reno[7]. Then in the remainder, the comparative study will focus on the original trace and the simulated one where elephants are generated using TFRC.

In addition of the classical traffic throughput parameter, this study focuses on QoS statistical parameters as the LRD (as justified in section 2) and some

[7] TCP New Reno has been selected as it is currently the most used version of TCP in the Internet. To increase again the realism of simulations, it would be interesting to replay short flows with the same TCP version than the one that was used in the original trace, but finding out such information is impossible for most of short flows: only the ones that experiment a huge number of losses can provide enough information to find out the TCP version that was used.

parameters related to variability. For that, we used the Stability Coefficient (SC), that is define as the following ratio:

$$\text{Stability Coefficient (SC)} = \frac{\text{exchanged average traffic}}{\text{exchanged traffic standard-deviation } (\sigma)} \qquad (2)$$

4.2 TFRC impact on flow QoS

Figure 3 presents the traffic in both cases, i.e. in the real and simulated cases. It visually clearly appears that using TFRC for sending elephants, instead of TCP, makes global traffic much smoother, avoiding all the huge peaks that can be seen on the real traffic.

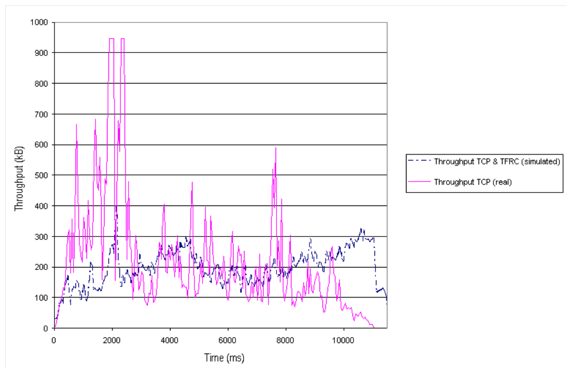

Fig. 3. Throughput evolution during time

Quantitatively speaking, results are indicated in table 1. This confirms that the traffic variability in the case of real traffic (using TCP for transmitting elephants) is much more important compared to the simulated case in which elephants are transmitted using TFRC (for the standard deviation σ it has been calculated that σ(real traffic) = 157.959 ko $\ggg$ σ(simulated traffic) = 102.176 ko). In the same way the stability coefficient is less important in the real case (SC = 0.521) than in the simulated one (SC = 0.761).

Dealing with the global throughput we got for both real and simulated traffic rather equal values (Throughput(real traffic) = 82.335 ko $\approxeq$ Throughput(simulated traffic) = 77.707 ko). This result is quite good as TFRC is not able to consume as many resources as TCP [12], and even if TFRC is less aggressive than TCP, it is able to reach the same performance level as TCP. This confirms the importance of stability for good performances [4].

Speaking about LRD in the simulated case, figure 4 shows that the bi-scaling property of the curve is strongly decreased, and that the curve has a very small slope. This means that all kinds of dependences, especially the long term ones

Protocol	Average throughput (kB)	Throughput σ (kB)	SC
TCP New Reno (NR): real case	82.335	157.959	0.521
TCP NR & TFRC: simulated case	77.707	102.176	0.761

Table 1. Throughput evolution during time for TCP and TFRC protocols

have been drastically reduced. The values for the LRD (Hurst factor are: (H(real traffic) = 0.641 and H(Simulated traffic) = 0.194). Such result confirms two aspects of our proposal:

- TFRC helps to smooth individual flow traffic (thus providing a smoother QoS better suited for stream oriented applications) as well as the global traffic of the link;
- LRD is the right parameter to qualify and quantify all scaling laws and dependencies between oscillations.

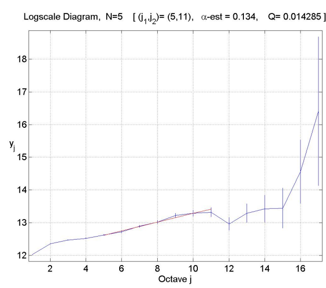

Fig. 4. LRD evaluation for simulated traffic including TFRC elephants

5 Conclusion

In this paper, we proposed a new approach for improving flow QoS. This approach relies on a preliminary study of Internet traffic characteristics that has been made possible thanks to some passive monitoring tools. This traffic characterization showed that Internet traffic suffers from the number and the amplitude of oscillations, especially important in the case of long flows, called elephants. The first contribution of this paper was then to explain why such oscillations arise, and proposes to use the LRD metric to characterize such feature in addition to the stability coefficient and other well known statistic moments as standard deviation. Therefore, the solution proposed in this paper consists in smoothing the traffic generated by each flow, especially elephants. The main protocol designed for this purpose (and under discussion at the IETF) is TFRC.

This paper then proposed a comparative evaluation of real traffic, and the same traffic but this time with elephants running TFRC instead of TCP. The results we got confirmed all our starting hypothesis in relation with oscillations, the LRD metric to characterize them, and the impact of TFRC for their reduction and for getting a smoother traffic, much more easy to handle.

However, it also appears that the global throughput that can be transmitted using TFRC instead of TCP is not higher. This is due, in fact, because TFRC is a less aggressive congestion control mechanism than the one used in TCP. The problem with congestion control is really tricky: on one side, transport protocols and their congestion control mechanisms have to be very aggressive to be able to rapidly consume network resources and being able to exploit the capacity of new networks, and on the other side, protocols have to be not aggressive to limit the oscillation phenomena that are very damaging for flow QoS. These two requirements are contradictory, but this is the challenge to enforce for next generation transport protocols, i.e. being able to rapidly consume resources without provoking damaging oscillations.

References

1. NS-2 web site: *http://www.isi.edu/nsnam/ns/*.
2. K. Park and W. Willinger, *"Self-similar network traffic: an overview"*, In Self-similar network traffic and performance evaluation, J.Wiley & Sons, 2000
3. K. Park, G. Kim and M. Crovella, *"On the relationship between file sizes, transport protocols, and self-similar network traffic"*, IEEE ICNP, 1996.
4. K. Park, G. Kim and M. Crovella, *"On the Effect of Traffic Self-similarity on Network Performance"*, SPIE International Conference on Performance and Control of Network Systems, November, 1997.
5. Abry P. and Veitch D., *"Wavelet Analysis of Long Range Dependent Traffic"*, Trans. Info. Theory, Vol.44, No.1 pp.2-15, Jan 1998.
6. Abry P., Veitch V. and Flandrin P., *"Long-Range Dependence: Revisiting Aggregation with Wavelets"*, Journal of Time Series Anal., Vol.19, No.3 pp.253- 266 May 1998.
7. LeGall D., Mitchell J.L., Pennbaker W.B. and Fogg C.E., *"MPEG video compression standard"*, Chapman & Hall, New York, USA, 1996.
8. Floyd S. and Fall K., *"Promoting the use of end-to-end congestion control in the Internet"*, In Proc. IEEE ACM Transactions on Networking, 14 pages, February 1998.
9. Floyd S., Handley Mr., Padhye J. and Widmer J., *"Equation-based congestion control for unicast applications"*, In Proc. ACM SIGCOMM, 14 pages, 2000.
10. Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, *"On the Constancy of Internet Path Properties"*, Proc. ACM SIGCOMM Internet Measurement Workshop (IMW'2001), San Francisco, California, USA, November 2001.
11. S. Donnelly, I. Graham, R. Wilhelm, *"Passive calibration of an active measurement system"*, in Passive and active measurements workshop, Amsterdam, April 2001.
12. P. Owezarski and N. Larrieu, *"Coherent Charging of Differentiated Services in the Internet Depending on Congestion Control Aggressiveness"*, to be published in Computer Communications, special issue on "Internet Pricing and Charging: Algorithms, Technology and Applications", 2003.