Towards a Measurement Based Networking approach for Internet QoS improvement

Nicolas LARRIEU and Philippe OWEZARSKI

LAAS-CNRS
7, avenue du Colonel ROCHE
31077 TOULOUSE Cedex 4
FRANCE

Abstract

The Internet is on the way of becoming the universal communication network, and then needs to provide various services and QoS (Quality of Service) for all kinds of applications. We show in this paper that oscillations that are characteristic of the Internet traffic provoke huge decreases of the QoS that flows can get. After having demonstrated that such oscillations can be characterized by the Long Range Dependence (LRD) function (as well as the Hurst parameter), we propose an approach for improving Internet flows QoS based on smoothing sending rate of applications. TFRC (TCP-Friendly Rate Control) 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. Given these first results, this paper deals with, an evolution of the approach based on TFRC, by promoting a new architecture, called MBN (for Measurement Based Networking), aiming at developing new Internet mechanisms for better managing traffic, QoS or network behaviors. The idea of MBN first relies on the fact that traffic is very different from one link to the other, and that even on the same link, traffic is not stationary, exhibiting very frequent ruptures. Issuing a static protocol, or any kind of network mechanisms being optimal on all links all the times, with the presence of huge ruptures in the traffic, is quite impossible. Therefore, MBN proposes to use monitoring and measurement results in real time, to react to any rupture in the traffic, thus adapting to new traffic characteristics and constraints. This paper, then, illustrates the MBN approach on a case study: the development of a new congestion control mechanism called "MBCC" (for Measurement Based Congestion Control) extending TFRC on the basis of traffic monitoring and measurement results. Some preliminary results, based on NS-2 simulations, are presented. They show the perfect suitability of this new networking approach for improving traffic characteristics and QoS in the Internet, given the complexity, variability and versatility of actual traffic.

Key words: Internet monitoring, traffic characterization, quality of service, TFRC, congestion control, Measurement Based Networking

Preprint submitted to Elsevier Science *12 July 2004*

1 Introduction

The Internet is becoming the universal communications network for all varieties of information, from the simple transfer of binary computer data to the transmission of voice, video or interactive information in real time. These new applications require new network services. What has been a network offering a single best effort service now has to evolve into a multi-service network. The resulting question, of how to provide quality of service (QoS), has been a major issue for the Internet for the past decade. Though many proposals have been put forward, such as IntServ, DiffServ, and others, until now none has been deployed (or their deployment has been quite limited). The solutions that the Internet community has offered in the areas of differentiated and guaranteed services have not met the needs of users or operators (Internet service providers, carriers, etc.). Efforts have been stymied by the complexity of the Internet, its myriad systems of interconnections, and by the technological heterogeneity of these systems. They have also run up against poor general knowledge of how to provision networks, based upon traffic characteristics that are largely unknown, and that might deviate significantly from standard suppositions. 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 characteristics.

Given this previous work on traffic monitoring, section 2 also shows 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. The causes of such traffic characteristics have been analyzed and it appears that that TCP congestion control mecha-

⋆ This work is achieved in the framework of the METROPOLIS project, a French national project granted and funded by the French Network for Research in Telecommunications. METROPOLIS main goal deals with issuing new network monitoring and analysis methodologies. This work also got the support of the E-NEXT European network of excellence.

Email address: {nlarrieu, owe}@laas.fr (Nicolas LARRIEU and Philippe OWEZARSKI).

nism is largely responsible of such oscillations. Such result makes us propose some improvements for the Internet. More precisely, section 3 proposes to use a smoother transport protocol 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 also presented. These experiments have shown that TFRC can optimize Internet QoS by smoothing its traffic. And in the same way, this work also illustrates how traffic measurement and monitoring can be used for network and protocol engineering.

However, even if TFRC contributes to enhancing traffic characteristics and then network QoS, the many experiments we performed exhibited that TFRC has variable benefits depending on the links or types of traffic that is transmitted on this link. In fact the type of traffic and its characteristics is changing a lot from one link to the other. In addition, traffic can vary significantly from one time to the other, exhibiting strong ruptures that can be due to daily variations, but also to some popular events as the broadcasting of a cultural or sport event; or due to the publication of a very attractive document on a web server that can raise a lot of attention. Facing such variability, TFRC does not always provide the same level of improvement, and it seems quite impossible to propose a transport protocol which will be optimal in all cases. In fact, this shows the limits of the traditional network and protocol engineering methods given such changing characteristics of Internet traffic. Consequently, section 4 proposes a new approach for managing traffic. It extends the static engineering method based on measurement and monitoring results analysis proposed in section 3, by exploiting the measurements and monitoring results in real time. This will allow the traffic sources and their transport protocol to react in real time to any rupture in the traffic, thus adapting in real time to new traffic characteristics and constraints. This approach, called MBCC for Measurement Based Congestion Control, is in fact an extension of TFRC including the capability of using measurement and monitoring results in real time for deciding what action to trigger depending on the actual network and traffic state[1]. The objectives of MBCC are of course to limit the number of congestions and related losses in the network, but also to improve traffic regularity, optimize network resources utilization, as well as providing fairness between flows. MBCC principles are presented in section 4.2. As well, some simulation results are given in section 5 to compare the results got on the traffic depending whether MBCC or TCP is used for transmitting flows. Finally, section 6 concludes this paper.

[1] MBCC is in fact an example of the generic MBN (Measurement Based Networking) approach (detailed in section 4.1) which can be extended to many other networking areas as traffic management, traffic shaping, traffic engineering, QoS optimization, provisioning, pricing, etc.

2 Traffic characterization

This first section is then presenting the characterization and analysis results we got after analyzing the traffic traces that have been captured on RENATER [2] network [3]. To well understand the new traffic characteristics, it is first required to analyze the evolution of the Internet in terms of usages.

2.1 Evolution of traffic characteristics

The evolution of Internet traffic these last years has been marked by the huge increase of P2P traffic (Kaaza, e-donkey, ...), and now, on some links of the RENATER network, it can represent the same proportion than HTTP traffic (Figure 1). Such a result is quite surprising because, in an academic network as RENATER, students, teachers and researchers are not supposed to download music or movies (and they commit not to use the network for non professional usage by signing the Renater agreement). In fact, even if some P2P traffic exists, its amount in RENATER is pretty low compared to the results observed on the commercial network of France Télécom [4], especially on the ADSL POP were P2P traffic can grow up to 70 % – and sometimes more!

Such an increase of P2P has necessarily an impact on traffic characteristics. In particular, because of the nature of file exchanged – mostly music and movies – that are very long compared to web traffic that was the dominant traffic in the Internet few years ago. In fact, the increase of P2P traffic, in addition of the classical traffic, makes the traffic have the following characteristics:

- There are always thousands of mice [5] in Internet traffic (because of the web, as well as P2P controls). On Renater, mice represents around 80% of the number of flows but less than 20% of the amount of traffic;

[2] RENATER is the French network for education and research that interconnects all universities, public research labs, some schools, as well as some industrial partners (depending on the project in relation with academia they are involved in).
[3] The results presented in this paper have been obtained by analyzing Renater traffic, but the same results have been obtained on all traffics of all other analyzed networks. Results presented here then represent the actual current state of the art in traffic characterization.
[4] France Télécom R&D is part of the METROPOLIS project, but the results got on the France Télécom network are not public and will not be more discussed in this paper.
[5] "Mouse" is a term used to designate a small flow, i.e. a flow that does not last enough to exit from the slow start phase of TCP. At the opposite, very long flows are called "elephants".

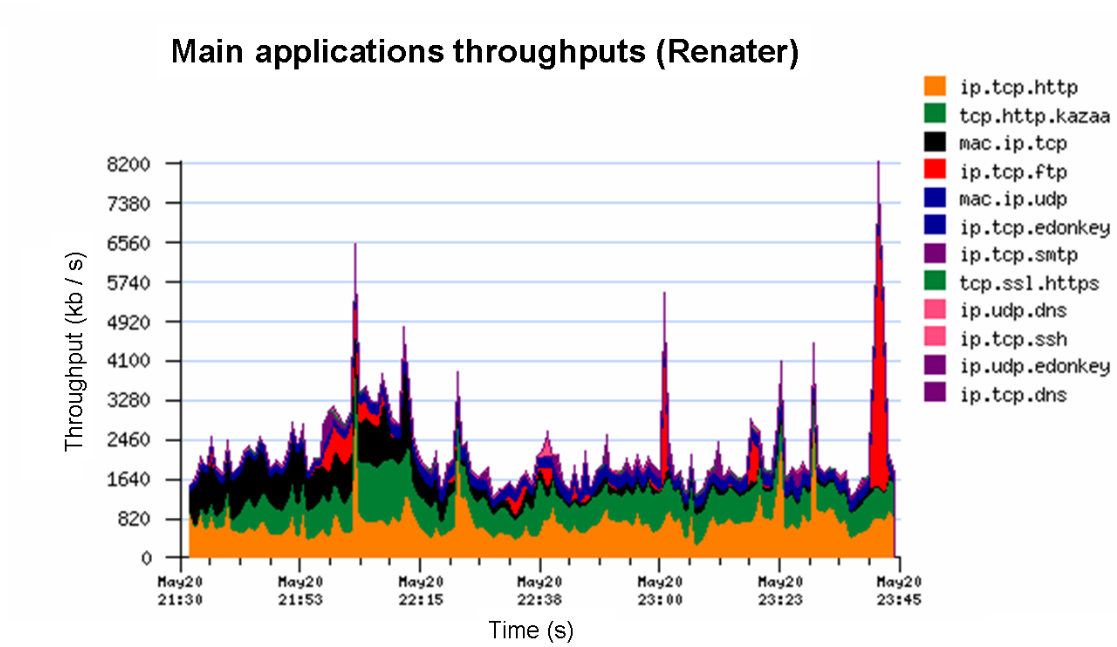


Fig. 1. Traffic distribution on the RENATER network in May 2003. *On the figure, the traffic of each application is represented by order of importance: then HTTP traffic is at the bottom of the curve, kazaa is just over, and so on. The applications are indicated on the list on the right part of the figure, and are ordered according to their amount of traffic. HTTP is then the application creating the more traffic, DNS is the one creating less traffic (among the ones which are printed).*

- But there are also a larger and larger number of elephants. Elephants represents around 5% of the number of flows on Renater, but more than 60% of the amount of traffic.

So, one of the main consequence of the evolution in terms of applications and usage is related to the flow size distribution changes. Figure 2 represents the flow size distribution between 2000 and nowadays. The exponential function is taken as a reference because the exponential distribution is closely related to the Poisson model that is most of the time used as the reference model for Internet traffic (for instance for call arrivals) in simulations or for performance evaluation. We can see on this figure that between year 2000 and nowadays the proportion of very long flows has increased in an important way. If in 2000, flow size distribution was almost exponential, this is completely wrong nowadays. Current distribution is very heavy tailed, and this distribution is then very far from the exponential distribution traditionally considered.

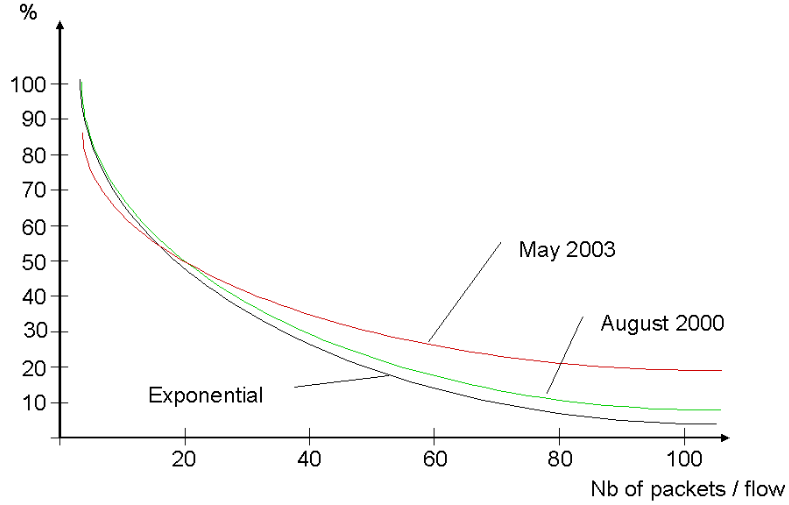


Fig. 2. Flow size distribution evolution between 2000 and 2003. *For readers that cannot get a color printing of this figure, exponential is the lower curve, the curve for August 2000 is just over, and the one with the heavier tail is the one of May 2003.*

2.2 Traffic Long Range Dependence and related issues

This increase of the proportion of P2P elephants hugely impacts traffic profile. Figure 3 illustrates it in the current traffic. It shows the difference between the actual Internet traffic and Poisson traffic. These two traffics are observed with different granularities (0.01 s, 0.1 s and 1s), and it appears that Internet traffic does not smooth as fast as Poisson traffic when increasing observation granularity. The analysis demonstrated that this result is completely due to elephants. In fact, the transmission of elephants creates in the traffic the arrival of a large wave of data that has the particularity of lasting for a long time – more than 1 second – while web flows are generally transmitted in less than one second on the current Internet. That is why we have this difference between Poisson and real traffic: the nature of oscillations between the two traffics changes, with oscillations in actual current traffic that are more persistent.

In addition, as TCP connections used for transmitting larger flows are longer, the dependence that exists between packets of the same connection propagates on longer ranges. This phenomenon is usually called Long Range Dependence (LRD) or long memory. It has several causes, and in particular congestion control mechanisms deployed in the Internet, especially the ones of TCP, this protocol being the dominant protocol in the Internet [14]. Among all the TCP mechanisms, it is obvious that its closed control loop introduces dependence, as acknowledgements depend on the arrival of a packet, and the sending of all the following packets of the connection depends on this acknowledgement. In the same way, the two TCP mechanisms – slow-start and congestion avoidance – introduce some dependence between packets of different congestion control

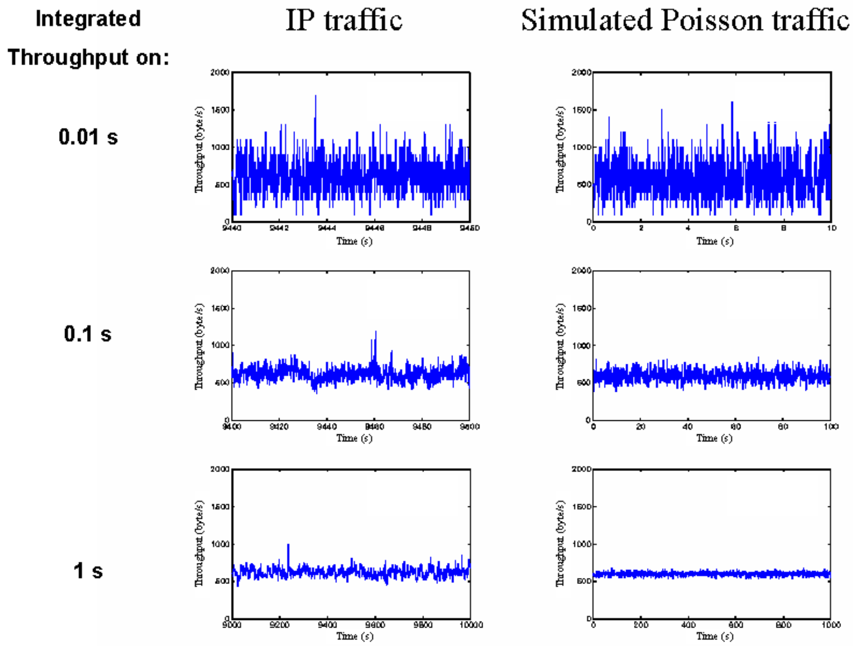


Fig. 3. Comparison between oscillations of Internet and Poisson traffics

windows. By generalizing these observations, it is obvious that all packets of a TCP connection are dependent the ones from the others. In addition, with the increase of the Internet link capacities that allows the transmission of longer and longer flows, it is obvious that the range of the LRD phenomenon increases. That is why the persistence of the Internet traffic oscillations measured, even with a coarse granularity, is so high. Indeed, because of the TCP dependence phenomenon propagating in the traffic thanks to flows (connections), the increase of flow size also makes the dependence range increase and propagates on very long ranges. An oscillation at time t then provokes other oscillations at other time being potentially very far from t. A (short term) congestion due to a huge oscillation of a connection can then continue to have some repeats several hours later (in the case of a movie download for instance), i.e. this flow will continue to propose to the network some traffic peaks directly dependent from this first oscillation, and can create some new short term congestions. Moreover, it is clear that elephants, because of their long life in the network, and because of the large capacities of networks – most of the time over-provisioned – have time to reach high values of the congestion control window (CWND). Thus, a loss induces a huge decrease, followed by a huge increase of the throughput of the flow. The increase of flow size then favors high amplitude oscillations, dependent on very long ranges. Of course, oscillations are very damaging for the global utilization of network resources as the capacity released by a flow that experiences a loss (for example) cannot be immediately used by another flow (because of slow start for instance): this corresponds to resource waste, and introduces a decrease of the global QoS of the network. In fact, the more the traffic oscillates, the lower the perfor-

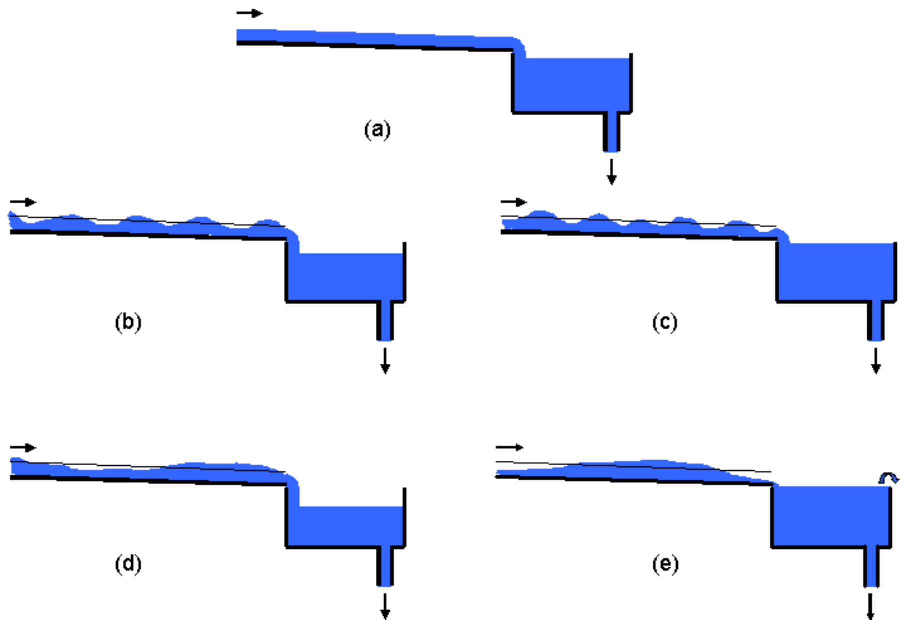


Fig. 4. Illustration of LRD issues on losses

mances [15].

To give a concrete view of LRD issues on traffic, Figure 4 aims at illustrating it on a simple traffic parameter: losses. Figure 4.a depicts a leacky bucket as an analogy with a router, for instance, its buffer, ingress and egress links. So, when there are waves in the arriving traffic (Figure 4.b), and if the goal is to provide a good service with no extra losses and no extra delays, it is first required to over-provision the link (otherwise the traffic will be smoothed, and at least, delays will be introduced for some packets). The second characteristic appears in the buffer when a wave arrives: it makes the level of the buffer increase (Figure 4.c). This is a well known issue of networking addressed many time before, especially by [9]. But when the range of oscillations increases (Figure 4.d), and this is the case with current Internet traffic, the arrival of a persistent wave provokes a buffer overflow, thus leading to losses (Figure 4.e). As a conclusion of this practical illustration, it is important to point out that LRD in traffic induces bad performances and QoS for networks, as it is the source of congestion and losses.

But what has been shown in what precedes just presents some qualitative results on the nature of Internet traffic. It is however also needed to quantify its oscillating nature and its LRD characteristics. For that, it is most of the time accepted to use the auto-correlation [6] function, that can show on a graph

[6] The auto-correlation function is equivalent to the auto-covariance function. The two names designate the same function. This is a very basic function in statistics.

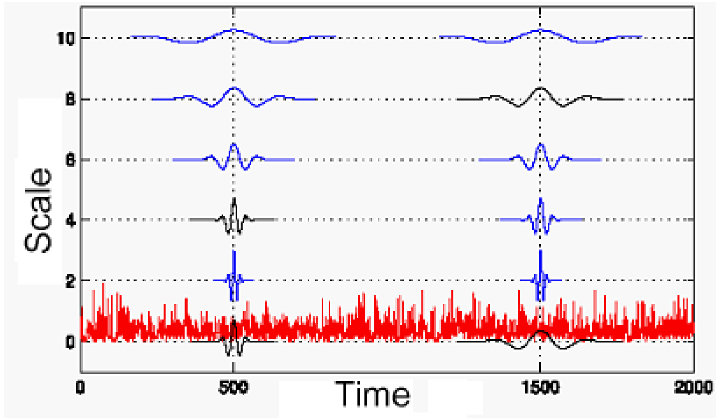


Fig. 5. Principle of wavelet decomposition

the correlation (and then the dependence) that exists between the transmission of two packets separated by k packets. Then the auto-correlation function can show for large values of k the trends of LRD in the traffic. But, for accuracy reasons, and to limit bias in the computing of LRD, we recommend to use a wavelet based analysis of the traffic [1]. The wavelet function has been selected as it is the best way to represent an oscillation (proof is visual – see figure 5). The principle deals with extracting from the traffic all wavelets. For that we use several wavelet functions with different frequencies to get the different ranges of variability of the analyzed traffic, as depicted on figure 5. The different wavelets represent different ranges of oscillations. The waves with the largest periods represent the very long waves, i.e. the ones generated by elephant flows. Interested readers can refer to [2].

The curve on Figure 6 has been obtained using the LD Estimate tool [17] that estimates the LRD that appears in Internet traffic at all scales. The output of this tool is a graphical representation of the dependence laws at all time scales. It represents the result of a wavelet based analysis of Internet traffic (and it is important to note that the same qualitative result has been exhibited for all links that have been monitored all over the world by researchers working on Internet links monitoring). It represents the variability of the oscillations depending on the observation range. Also note that the Hurst factor H that is the factor fully characterizing a self-similar process – and Internet traffic is often said to be self-similar [11] [16] – can be obtained directly depending on the slope of the LRD curve. The curve on figure 6 shows a bi-scaling phenomenon (two lines in a log-log scale), with an elbow around octave 8, which shows a difference in the LRD level between short and long time scales for the traffic exchanged, and meaning that there are two different invariant power laws. For short scales (octave < 8, left part of the curve), 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

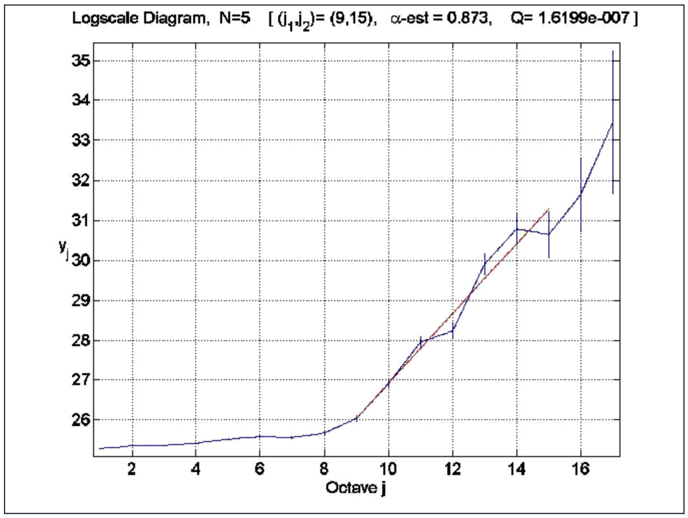


Fig. 6. LRD evaluation for edge network traffic

time scales (octave $>$ 8, right part of the curve), 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 control loop of TCP congestion control mechanisms 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 6 for very large octaves. Note however that some additional experiments also showed that the elbow in the curve corresponds to the mean size of flows, meaning that the right part of the curve corresponds to the impact of elephant flows.

What comes out from this LRD analysis, given the fact that traffic LRD has a very bad impact on network QoS, is that the use of network resources is far from being optimal (especially because, as it has been demonstrated before, TCP is not suited for the transmission of long flows on high speed links). This implies that LRD forces Internet carriers to hugely over-provision link capacities compared to the amount of effective traffic to transmit. It then comes out that LRD (that is a parameter helping to characterize the oscillating nature of Internet traffic) is a good parameter for quantifying what QoS a network can provide in the transmission of the considered traffic: the higher the LRD, the worse the QoS.

3 A new approach for improving Internet QoS

3.1 *Increasing QoS by smoothing flow behaviors*

Given these results on traffic characterization and analysis, 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.

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 [7]. It is actually under discussion at the IETF (Internet Engineering Task Force) in the DCCP (Datagram Congestion Control Protocol) working group to be integrated in the new transport protocol they propose. Note however that for both TFRC and TCP, we will estimate the evolution of the oscillating behavior of the traffic by evaluating LRD features (as well as the Hurst factor H that is closely related) on packet arrival series.

3.2 *TFRC principles*

TFRC aims at proposing 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 (Round Trip Time), the sending rate according to the Loss Event Rate (LER) measured by the receiver [5] [6] [8] according to

[7] TCP Vegas and Westwood also contribute to smoothing traffic rate. However, we do not consider them as they are performing really less than TCP in term of bandwidth usage. TFRC is up to our knowledge and experiment results, the only protocol able to reach the same performance level as TCP with a smooth sending rate for each flow.

equation 1:

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

where:

- X is the transmit rate in bytes/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 and is normally equal to $4 * R$,
- b is the number of packets acknowledged by a single TCP acknowledgement.

In TFRC, the RTT is computed thanks to control packets sent between the two entities. A simple difference between arrival times of consecutive packets belonging to two different sending periods allows the sender to estimate the connection RTT.

The loss ratio (LER in TFRC definition [7]) is computed by the receiver by dividing the total number of received packets by the total number of packets exchanged in a RTT (computed thanks to both sending throughput and packet size and also sequence numbers included in data packets). Therefore, 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 [18] that proposes an analysis and a model for the Internet loss process.

3.3 Evaluation of TFRC impact on QoS

Our experiment aims at providing a comparative evaluation of the global traffic characteristics if elephants use TCP or TFRC as the transmission protocol. This experiment aims at providing values in a realistic environment. For that, every simulation is based on real traffic traces collected on the RENATER network. These traces are replayed in NS-2 with a special module detailed in [13] whose goal is to make simulation scenarios as realistic as possible, in particular with replayed traffic having characteristics as close as possible from real traffic. This module then replays in simulation environments traffic samples reproducing all characteristics of real traffic, with all its variability and LRD characteristics for example (interested readers can refer to the evaluation of this monitoring-based replay module in [13]). In this experiment aiming at smoothing traffic oscillations and reducing LRD, elephant flows are then transmitted in the NS-2 simulator using TFRC while others flows use TCP New Reno [8]. Then, the following study provides a comparative study between 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 parameters related to variability. For that, we used the Stability Coefficient (SC), that is define as the following ratio:

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

Figure 7 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.

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 kB/s $\ggg$ σ(simulated traffic) = 102.176 kB/s). 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 kB/s $\approxeq$ Throughput(simulated traffic) = 77.707 kB/s). 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 almost the same performance level as TCP. This confirms the importance of stability for good performances [15].

Speaking about LRD in the simulated case, figure 8 shows that the bi-scaling

[8] 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.

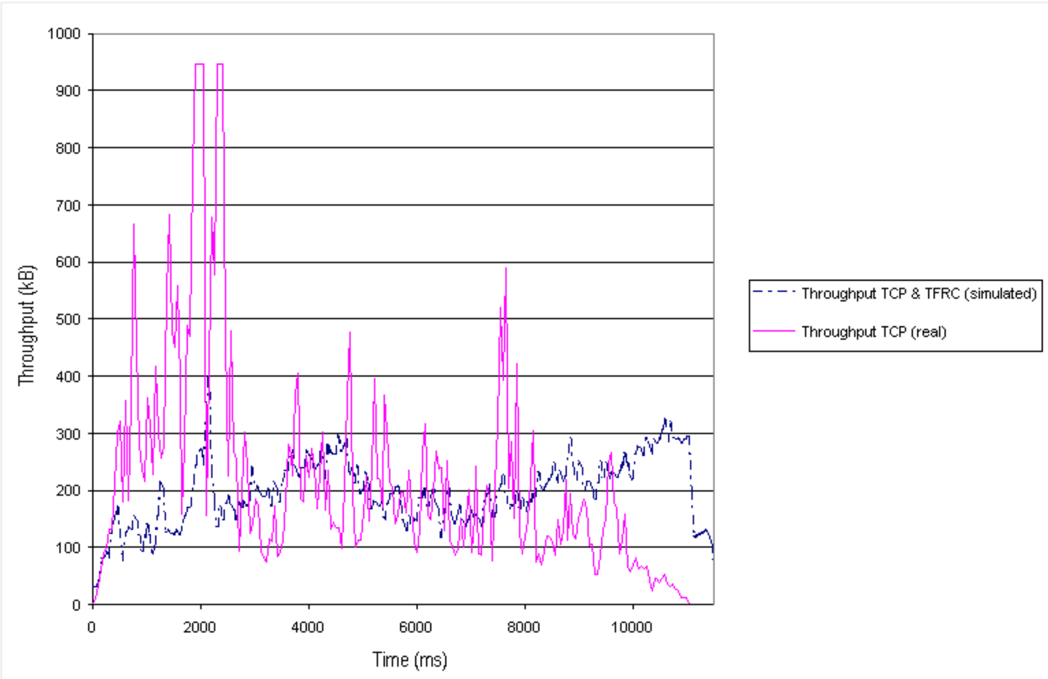


Fig. 7. Throughput evolution during time

Table 1
Throughput evolution during time for TCP and TFRC protocols

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

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 have been drastically reduced. The values for the 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 related to oscillations.

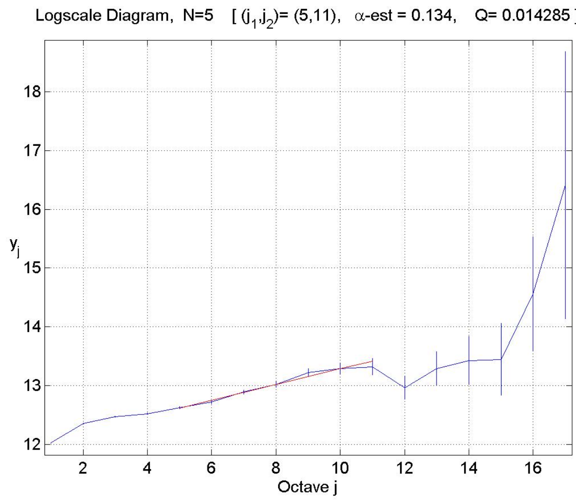


Fig. 8. LRD evaluation for simulated traffic including TFRC elephants

3.4 TFRC evaluation conclusion and discussion

The conclusion of this first partial study is quite good as we demonstrated that using a congestion control mechanism taking into account the oscillating nature of the traffic (and then the related LRD), by smoothing it as much as possible, as well as being not aggressive, can significantly improve the network characteristics, bringing more stability in the traffic, and thus giving a better environment for providing guaranteed services. Of course, the global throughput got using TFRC is not greater than the one using TCP, because TFRC aims at being TCP friendly, and then limits its sending rate according to the statistical TCP rate equations [3]. But, TFRC, and this is its second big advantage compared to TCP, induces less congestions and losses in the network, because of its reduced aggressiveness. It then optimizes the use of resources in the network, and in particular the use of bandwidth. In fact, using TFRC instead of TCP, because of the reduced number of losses, and then retransmissions, helps in saving some resources. The goal of next sections is to propose a solution for being able to consume this spare bandwidth, and then to provide a new congestion control mechanism – MBCC – extending TFRC and which will be more efficient than the one of TCP, and that will rely – of course – on the concept of providing smooth sending rates.

Given the results of previous experiment, and the results of traffic characterization analysis, it appears two main issues:

- First, TCP, and by transition TFRC also, are looking at only one parameter to evaluate the congestion level on the worst link of a path: they interpret out of sequence or delayed packets as losses, and losses as congestions [9].

[9] All TCP versions except TCP Vegas that uses both losses and RTT.

And in that case, they drastically reduce their sending rate. Of course, one single parameter is not enough to characterize what happens on a path in a network as the Internet. Other parameters as the number of competing flows, the delays variations, traffic matrices, etc. can give information on the actual state on the path which appeared congested. In particular, because of the increase of network capacities, that has lead to very high speed links, congestion issue on a link can be recovered much faster than the TCP reaction time, which depends on the RTT. MBCC design principle relies on the analysis of much more traffic and network parameters than the single out of sequence packets.

- Second, as it has been mentioned just before, the traffic and network states in the case of the new high speed links can change pretty fast. Indeed, the analysis of traffic traces shows that traffic is not stationary. It can be stationary on few hours, but it is easy to understand that the traffic is different during nights and days, and this on both the amount of data that are transmitted, and in terms of applications, then impacting the traffic characteristics as flow size distribution, LRD, correlation, etc. Large differences can be seen also on traffic during the week–end and working days, at lunch time compared to working hours, etc. But it also appears some important and unpredictable ruptures at any time. For instance, if a very popular event is broadcasted on the Internet, it will raise a large number of connections and provoke a huge increase of the traffic. It is the same if a very attractive document is published on the web. But these ruptures can also be due to some denial of service attacks (DoS), or network components failures. In fact, such ruptures appear to be very frequent on the Internet traffic. Because of this huge variability of the traffic, its inconstancy, its non stationarity, it is impossible to issue static[10] mechanisms or protocols that can be perfectly suited in all these conditions. Thus, MBCC will be designed as a dynamic mechanism that will adapt upon the network and traffic conditions.

Our proposal for solving these two issues deals with using traffic monitoring and measurement equipments, which have been used for characterizing and analyzing traffic in section 2. These equipments are able to provide many parameters to analyze network state, and in particular are able to detect ruptures. By signaling them to MBCC sources, MBCC sources can in real time adapt to new traffic and network conditions.

[10] By static, we mean a protocol running always the same mechanisms, as it is the case for TCP or TFRC.

4 Measurement Based Congestion Control

MBCC is in fact an illustration of the much generic MBN concept – based on measurements – that could be used in several other areas of networking. This generic architecture is presented in the following section.

4.1 The generic "Measurement Based Networking" principles

Figure 9 illustrates the topology and administrative structure of the current Internet. The Internet is generally defined as an interconnection of networks. That is of course true, but such definition does not include everything. In fact, the Internet can be more and more considered as a global worldwide network but splitted in several domains (also called AS for Autonomous Systems), administratively independent and independently designed and managed. Each network of each AS then does provide users with different services and QoS levels, especially with new kinds of networks that rely on new technologies as wireless or satellite networks, and that are providing very different quality of services. In such a context, ensuring end-to-end QoS is a pain as the final QoS got by users will be the one of the worst network on the path from source to destination. In particular, the peering links interconnecting the different AS are generally under-provisioned, and the source of huge QoS and performance decrease in end-to-end communications [11].

Given such topological structure of the Internet, in addition of all the issues (quoted in the introduction of this paper) related to current traffic as unstability, non-stationarity, huge oscillating nature, correlation, dependence, and a huge versatility of traffic types during time, it is easy to understand that it is impossible to find an optimal solution suited for all connections in the Internet. This statement leads us proposing MBN in order to react in real time and locally to some events on the network. Indeed, TCP end-to-end control loop can not react fast enough to some changes arising in the middle of the path from the connection source to its destination.

The first requirement of MBN is then to be aware of the network and traffic changes. It is then necessary to measure traffic and QoS parameters locally, as well as on long distances when the connection crosses several domains. Figure 9 also depicts how measurement tools can be deployed in the Internet for this purpose. Note that we are completely convinced that measurement and monitoring tools that are more and more popular and then more and

[11] Since few years, this issue has raised some efforts from carriers and Internet Service Providers (ISP), and the capacity of peering links has been increased, thus improving their QoS.

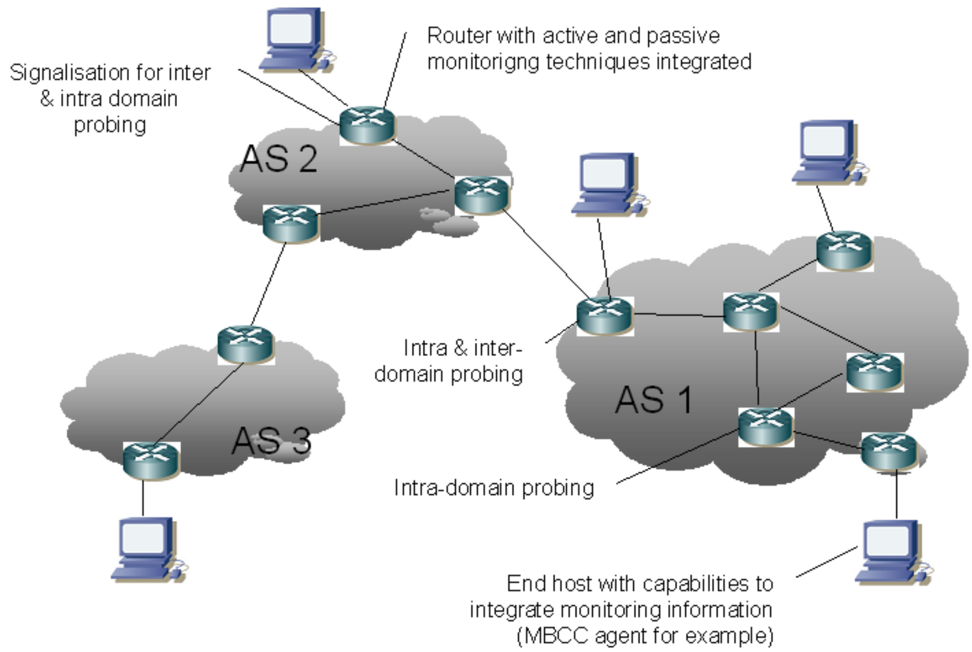


Fig. 9. Network entities needed in the network to deploy MBN

more deployed in the real networks, should be generalized in a short future. In addition, even if it is impossible to say that all links and all routers of the Internet will be monitored one day, we argue that taking into account the results of the measurement and monitoring tools effectively deployed in the Internet will be of great interest for improving Internet networking. MBN is designed that way: even if on some points measurement information is not present, the network continue to work with good performances and QoS. But performance and QoS can be much improved, and even become optimal, if measurement information is available.

Thus, given the administrative topology of the Internet, we propose to use different measurement techniques. Then, intra-domain measurements as loss ratio, used and available bandwidth, number of active flows, etc., can be made using passive equipments (as Netflow, SNMP based tools, etc.). This is possible and certainly very easy as the domain is administrated and managed by a single entity, and such measurement or at least management tools might be already present. In addition, passive measurement tools provide information on the traffic with a carrier point of view [12]. In fact, only delays measurement will be done in an active way, for easiness reasons (measuring delays with passive techniques is more difficult as it would require tracking packets from hop to hop).

On the other side, for inter-domain measurements it is impossible to use passive techniques as the other domains are not managed the same way, and their

[12] It is generally said that passive measurement are carriers oriented measurements.

administrators may not use measurement techniques, or not necessarily the same techniques. In addition, even if they are performing measurements on their domain, in an open market where ISP have to compete to each other, they may not be willing exchanging such measurement information. In addition, you never know if the measurement information they send you is trustable? So, in that case, it is required to address a measurement technique with a user point of view. Therefore, if you want to get information on other domain, the best solution consists in measuring what you need with active techniques, i.e. sending packet through the other domains, and measure what happens to these probe packets.

Then, all these measurements performed in real time and signaled to traffic sources (i.e. service users), can give an accurate knowledge of network and traffic state, and allow them to perfectly adapt their sending rate (for instance) to available resources. Note however that one important aspect of MBN deals with the design of a protocol for signaling measurement information. Such protocol has necessarily to work in intra-domain, but can also be extended for inter-domain signaling. However, as you do not know if information signaled by a "partner" ISP is correct, we recommend to keep the active measurements for checking it. Recall also that measurement information can be missing, for example if the measurement tool crashes, if some links are congested forbidding measurement signaling to reach you, or if your ISP "partner" suddenly decides not to cooperate anymore, etc. We are designing MBN in order to be able to continue to work efficiently, even if measurements are missing.

We do believe that MBN can be a universal solution for managing the Internet and its traffic. In particular, MBN has been designed in order to be able to provide a suited solution that can adapt to any kinds of networks, any traffic nature and conditions, etc. The remainder of this paper illustrates its benefits on the case of congestion control.

4.2 MBCC principles

In fact, as it as been said in section 3.4, the objectives of MBCC are:

- to smooth its sending rate;
- to optimize resource, and in particular bandwidth utilization;
- to provide fairness between competing flows.

Given the good results obtained with TFRC for smoothing traffic, MBCC is in fact an extension of TFRC to which it adds some capabilities for using measurement results coming from passive and active monitoring equipments in the network. Making this choice also helps to fulfill one of the MBN requirements that is to continue providing good results even if monitoring information are missing[13], as it has been demonstrated in section 3.

The extension of TFRC is related to the use of monitoring tools that can compute the available and consumed bandwidth on the monitored links. Thus, and this is the main principle of MBCC, the TFRC algorithm for computing the sending rate can be corrected thanks to the knowledge of the available and consumed bandwidth. If bandwidth is available, then sources should generate more traffic than indicated in equation 1 (corresponding to a TCP flow throughput) without creating any loss or congestion in the network. On the other hand, information about available and consumed bandwidth per flow can help sources to adapt their throughput during the whole duration of the connection. Thus, the network congestion level should be significantly decreased by having "proactive" sources able to adapt in real time their sending rate according to available resources. As well, such mechanism will help to improve fairness between flows, as the correction on the sending rate will not depend on the RTT connection value, but on the real bandwidth consumed as well for the additional bandwidth allocated.

Thus, a new algorithm for computing source sending rate (X) has been defined[14]. It is important to note that at the beginning of a connection transmitting an elephant, throughput starting value is computed thanks to the TFRC throughput equation decribed in equation 1. This feature is important because it allows MBCC agents to continue working in an autonomous way even if they do not receive any monitoring information, for instance in a strong congestion case, or when monitoring probes cannot work anymore.

Moreover, for a normal period (when monitoring information is correctly received and when there is no congestion), each elephant flow can get an additional fraction of available resources. This fraction is computed by dividing the total available bandwidth by the average number of elephant flows in the network at this time. It makes sense to divide the available bandwidth by the average number of active flows (N) crossing this link, as it has been demonstrated in the Metropolis project that elephant flow arrivals follow a Poisson process [4]. Indeed, in a Poisson process, as mean equals variance, the average number is significant because the process values will never be far from this average. At the opposite, for a self-similar process whose variance largely overruns its mean, it would have been better to consider the sum of mean and variance in the bandwidth function of MBCC. In fact, even with a Poisson process, the number of active flows can grow up to two times the average value (i.e. the mean plus variance) and dividing only by the average N, can create

[13] Note that monitoring information can be missing if they have been lost during the transmission by the network, but also if the monitoring equipment fails, or if monitoring equipments are not deployed on this network.

[14] Note that this algorithm is only defined for elephant flow senders.

cases in which all active flows, all together, will be allocated more bandwidth than available. Therefore, some congestions and losses can arise but as the number of flows is never far from its average on long periods, and because the traffic considered has some elastic properties, the number of losses will be quite reduced. Indeed, we checked that the bandwidth usage optimization overruns the wastes due to retransmissions. However on a different kind of network providing a very limited amount of resources as wireless networks for instance, the optimal trade off could be to avoid as much as possible losses and retransmissions. In that case, in the bandwidth function of MBCC, it would be better to allocate to each flow an extra bandwidth equal to the total available bandwidth amount divided by $2*N$ (i.e. mean plus variance). As the number of active flows will never be greater than $2*N$ in a Poisson process, there will not be any congestion or losses, but all the extra bandwidth will not be used most of the time.

At last, for a congested period, MBCC senders have to reduce their sending rate for resolving congestion, and this trying to be as fair as possible. Thus, MBCC sources send the minimum value between the possible TFRC throughput and the effective throughput got by the flow at this time in the bottleneck of the network (this information being given by monitoring tools met all along the path).

So, the equations of this algorithm can be summed up as follows:

- For a period without congestion ($LER = 0$):
  $X_{MBCC} = X_{TFRC} + aBW_{flow}$
- For a congested period ($LER \neq 0$):
  $X_{MBCC} = min(X_{TFRC}; cBW_{flow})$

where:

- X_{TFRC} is the TFRC throughput computed thanks to loss and RTT estimation as described in equation 1;
- aBW_{flow} is the Available BandWidth per flow in the bottleneck link of the path (for example, figure 10 depicts one bottleneck link: the one where the passive probe is deployed). It is computed with the ratio $\frac{\text{total available bandwidth}}{N}$;
- cBW_{flow} is the Consumed BandWidth per flow in the bottleneck link on the path.

The consumed bandwidth per flow parameter has been integrated in the throughput update equation in order to add more fairness, i.e. providing equal bandwidth for all MBCC flows.

5 MBCC evaluation

This new congestion control mechanism has been implemented and evaluated using NS-2. It has also been needed to develop a set of tools for monitoring available and consumed bandwidth in the simulated network and to exchange the measurement results between routers and traffic sources.

5.1 Simulations details

The topology used is described on figure 10. It represents two different edge networks with high bandwidth and an access link with a capacity really less important than the one of edge links. This link represents the most "congested" link on the considered path, i.e. the one that will mainly influence the MBCC sending rate. This difference should induce important congestion periods where adaptability skills of MBCC can be analyzed, and its performance level compared with others congestion control mechanisms especially the ones of two TCP versions: TCP New Reno and TCP SACK. Every simulation is based on real traffic traces collected on the RENATER network. These traces are replayed in NS-2 with the special module also used for the experiment described in section 3.3.

In this experiment, short and long flows are also differentiated, for the same reasons than in section 3.3. Then, as mice do not induce any transfer problem in the network (in fact, it has be demonstrated that mice traffic represents a white gaussian noise [4]), it is not necessary to modify the transport protocol they use. They will be transmitted using TCP and more precisely TCP New Reno that is the most frequent version of TCP in the Internet. At the opposite, elephant flows create in the network long range oscillations which induce congestions. This is the reason why MBCC, a new congestion control mechanism, has been designed for suitably transmitting such elephants flows. Thus, simulations compare the case in which elephants are transmitted using our new MBCC congestion control mechanism, and the one in which they are transmitted using TCP SACK[15].

5.2 Analysis parameters

The main goal of this study is to compare MBCC adaptation capabilities regarding:

[15] TCP SACK serves as the TCP reference as it is the best performing version of TCP.

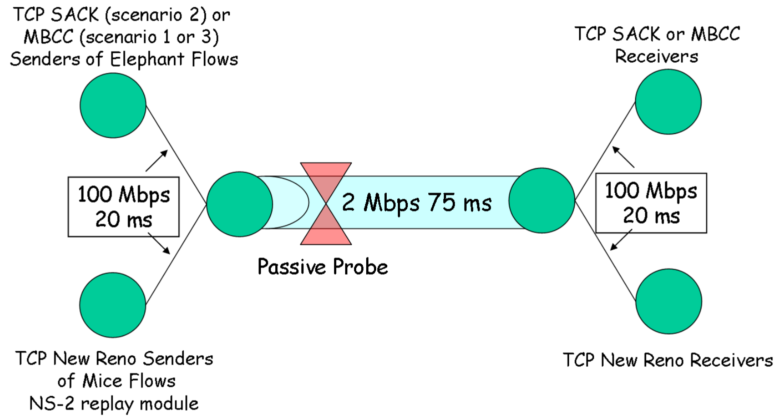


Fig. 10. Network topology used in NS-2 simulations

- the network load increases (and decreases);
- others congestion control mechanisms in order to evaluate fairness between different flows.

For evaluating MBCC and its contribution on the different objectives, several parameters have been evaluated in simulations:

- throughput evolution by traffic type (TCP or MBCC) to study the traffic variability [16]: this study is based on the computing of the average throughput (A), standard deviation (σ) and stability coefficient (SC $= \frac{A}{\sigma}$);
- loss process evolution in order to evaluate MBCC adaptation to resources compared to TCP;
- traffic oscillation range by computing the LRD and the related Hurst factor [17].

5.3 Experiment results

Several simulations have been realized with different real traces and they all exhibited similar results. Recall however that there are three different scenarios: in scenario 1, elephant flows are transmitted using MBCC while in scenario 2, elephants are transmitted using TCP SACK. In these two scenarios, mice flows are sent using TCP New Reno [18]. In a third scenario, all flows (mice and

[16] These parameters are the same as the ones defined in the TFRC evaluation in section 3.3.

[17] The Hurst factor, noted H, is a quantification of traffic LRD and represents also a good evaluation of traffic oscillation on long ranges. To compute it we use a wavelet based analysis of the traffic detailed in [1]. Note that LRD usage to quantify traffic oscillation have already been validated in [10].

[18] In fact, for mice, using TCP New Reno or TCP SACK, or any other TCP versions makes no difference. Mice being very short flows, there cannot be significant

elephants) are transmitted using MBCC. This experiment which represents an ideal case, is used as a reference to compare TCP and MBCC contributions to the congestion level in the network (cf. the loss process analysis in figure 12).

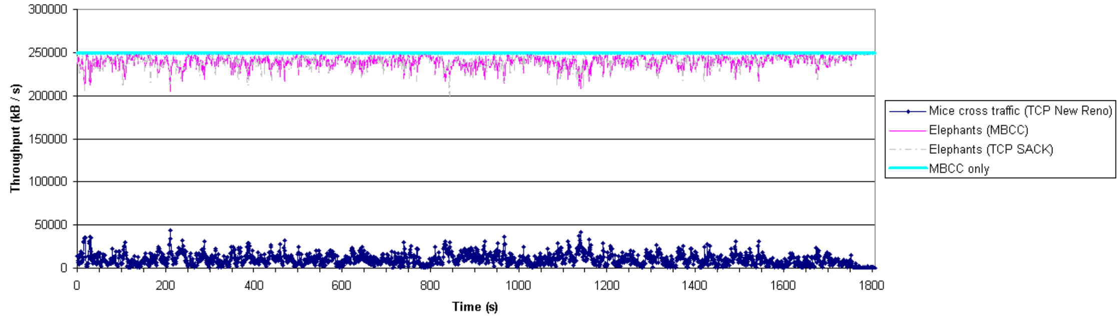


Fig. 11. Throughput evolution

First, the throughput evolution during time was computed. Figure 11 and table 2 describe these results. When there is a cross traffic (i.e. mice flows are present in the network, cf. standard and dash curves of figure 11), global performances are better for MBCC. Indeed, the stability coefficient for MBCC is higher than TCP SACK (cf. in the left part of table 2: SC(MBCC) > SC(TCP SACK)). The average throughput is higher for MBCC compared to TCP SACK, thus showing that MBCC is able to consume the available bandwidth that cannot be taken by TCP. Indeed, this result shows that all resources have been consumed efficiently. In addition, TCP SACK standard deviation is much more important than MBCC (cf. table 2 for scenarios 1 and 2: σ(TCP SACK) > σ(MBCC)). Finally, this shows the higher adaptability capability of MBCC compared to TCP, that helps to optimize available bandwidth utilization, keeping the level of oscillations very low, and thus providing a very stable service.

Table 2
Traffic variabily analysis

	SCENARIOS 1 & 2			SCENARIO 3
	Mice Flows TCP New Reno	Elephants Flows		Mice & Elephants Flows
		MBCC	TCP SACK	MBCC only
Average Throughput (B/s)	10348,7	243738,2	242014,2	249943,8
Throughput Standard Deviation (σ) (B/s)	6898,3	16350,5	19585,5	1453,6
Stability Coefficient (SC)	1,500	14,907	12,357	171,946

In addition, when there is no more cross traffic, figure 11 (thick curve) illustrates that MBCC makes a perfect optimization of the available bandwidth utilization. Indeed, there is no more oscillation for MBCC throughput and

differences while using different TCP version. We then chose TCP New Reno as its implementation is more optimized than the one of TCP SACK in NS-2, such limiting the time required for running the simulations.

global traffic is really less variable than TCP SACK (cf. the right part of table 2: SC(MBCC) $\gg$ SC(TCP SACK)).

This better optimization of global available resources is possible because MBCC creates less congestions than TCP in the network. Figure 12 illustrates this result. In this figure, you can see the cumulative loss number for both MBCC and TCP SACK traffic. It is then clear that MBCC inducing less losses in the network compared to TCP SACK (cf. curves of scenarios 1 and 2 in figure 12), there will be a much reduced number of retransmissions, then less bandwidth consumption that can be used for usable traffic. It is also important to mention that MBCC makes global losses in the network decrease even if TCP New Reno competing traffic makes global loss process increase in the first scenario compared to the ideal scenario 3 (cf. curves of scenarios 1 and 3 in figure 12).

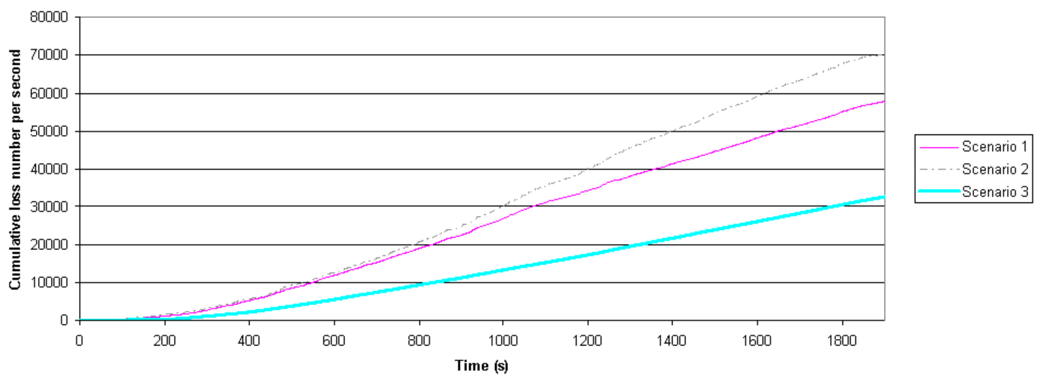


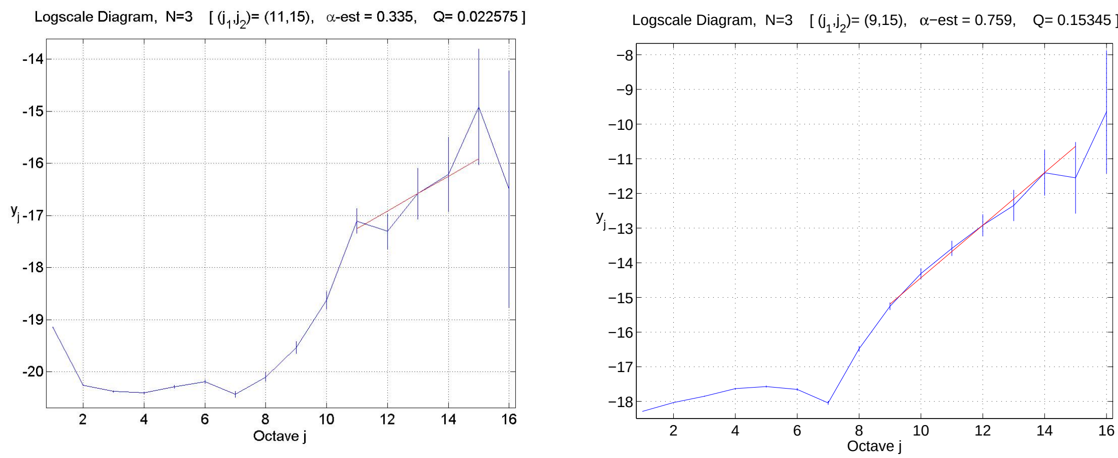
Fig. 12. Loss process

Finally, as it is depicted on figure 13, MBCC impacts in a very important way traffic LRD. In fact, thanks to MBCC, the LRD is much reduced in the global traffic (cf. figure 13(a) where H = 0.66) compared to the reference TCP traffic where LRD is very high (H = 0.88 in figure 13(b)). Consequently, there are less oscillations (cf. related stability coefficient values of table 2).

As a conclusion, the previous experiments put forward several good points of the use of MBCC for exchanging Internet traffic. These benefits in terms of traffic regularity, loss level and throughput, are directly felt by users as a QoS improvement.

6 Conclusion

In this paper, we proposed and evaluated a new approach for improving Internet 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



(a) MBCC traffic (H = 0.66) (b) TCP SACK traffic (H = 0.88)

Fig. 13. Traffic LRD estimation

the number and the amplitude of oscillations, especially important in the case of long flows, called elephants. The first contribution, introduced in section 2, was then to explain why such oscillations arise, and proposes to use the LRD metric to characterize such feature. Therefore, the solution proposed consists in smoothing the traffic generated by each flow, especially elephants. The main protocol designed for this purpose is TFRC and 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. This first method that makes us propose the use of TFRC for solving the issues on traffic characteristics illustrates a new approach for network and protocol engineering that is based on measurement, and that we call MBNE for Measurement Based Network Engineering.

But the results with TFRC also showed that the use of resources (in particular bandwidth) is not optimal as TFRC is not able to adapt to traffic variability and its very frequent ruptures. It is in fact impossible to issue a static solution working in an optimal way in all cases. As a consequence, this paper proposes an extension of TFRC called MBCC whose principle is to adapt to the huge variations of traffic when ruptures arise. MBCC relies on the use of monitoring and measurement tools which start to be widely deployed and that should be generalized in a short future. The principle of MBCC consists in using in real time the information on traffic characteristic evolutions that monitoring tools can provide, in order to perfectly adapt the reaction to the actual network and traffic conditions. The difference between MBNE and MBCC - or more generally to the generic MBN concept - is that MBNE relies on the use of static and invariant characteristics of traffic for issuing protocols or mechanisms that have to work all the time, while MBN is in fact based on specific suited real time reactions depending on the results of measurement, monitoring and analysis of traffic characteristics or network state, made in real time. The experiment results proved that MBCC reaches its objectives: it provides an optimal and smooth traffic throughput (due to a very limited number of losses), uses all available resources, and also, by construction, provides more fairness between flows.

As a final conclusion, it is clear that the results got with MBCC demonstrate the benefits that MBN can provide. Of course, a lot remains to do, especially on all the other aspects of networking than congestion control and traffic characteristics improvements which were the two aspects under consideration in this paper. However, we do believe that measurement results have to impact network engineering and must be at the origin of new networking approaches. That is why we do believe that MBN is the right new architecture to develop and deploy in the Internet for solving all traffic issues as oscillations, unstability and variability, improving traffic control, etc. In addition, it is also the right architecture for addressing the issue of providing multi-domains guaranteed end-to-end QoS given the administrative splitting of the Internet into domains and AS, which are providing different services and QoS; differences greatly increased by the arrival of new network technologies as wireless networks.

References

[1] P. Abry and D. Veitch, *Wavelet Analysis of Long Range Dependent Traffic*, Trans. Info. Theory, Vol.44, No.1 pp.2-15, Jan 1998.

[2] P. Abry, V. Veitch and P. Flandrin, *Long-Range Dependence: Revisiting Aggregation with Wavelets*, Journal of Time Series Anal., Vol.19, No.3 pp.253-266 May 1998.

[3] E. Altman, K. Avrachenkova and C. Barakat, *A Stochastic Model of TCP/IP with Stationary Random Losses*, in proceedings of ACM SIGCOMM, Stockholm, Sweden, August 2000.

[4] N. Ben Azzouna and F. Guillemin, *Analysis of ADSL traffic on an IP backbone link*, In Proc. Globecom 2003, San Francisco, December 2003.

[5] S. Floyd and K. Fall, *Promoting the use of end-to-end congestion control in the Internet*, In Proc. IEEE ACM Transactions on Networking, 14 pages, February 1998.

[6] S. Floyd S., M. Handley, J. Padhye and J. Widmer, *Equation-based congestion control for unicast applications*, In Proc. ACM SIGCOMM, 14 pages, 2000.

[7] S. Floyd, E. Kohler, J. Padhye, *Profile for DCCP Congestion Control ID 3: TFRC Congestion Control*, IETF Internet draft,February 2004.

[8] M. Handley, S. Floyd, J. Pahdye and J. Widmer, *TCP Friendly Rate Control (TFRC): Protocol Specification*, RFC 3448, Proposed Standard, January 2003.

[9] L. Kleinrock, *Queing Systems Theory*, Wiley, 1975.

[10] N. Larrieu, P. Owezarski, *TFRC contribution to internet QoS improvement*, 4th COST 263 International Workshop on Quality of Future Internet Services (QoFIS'2003), Stockholm (Sweden), October 2003, Lecture Notes in Computer Science 2811, Quality for all, Eds. G.Karlsson, MI.Smirnov, 2003, ISBN 3-540-20192-0, Springer, pp.73-82.

[11] W. Leland, M. Taqqu, W. Willinger, D. Wilson, *On the self-similar nature of Ethernet traffic*, ACM SIGCOM, September 1993.

[12] P. Owezarski and N. Larrieu, *Coherent Charging of Differentiated Services in the Internet Depending on Congestion Control Aggressiveness*, Computer Communication Journal, special issue on "Internet Pricing and Charging: Algorithms, Technology and Applications", Vol. 26, issue 13, August 2003.

[13] P. Owezarski, N. Larrieu, *A trace based method for realistic simulations*, IEEE International Conference on Communications (ICC'2004), Paris, France, 20-24 June 2004.

[14] K. Park, G. Kim and M. Crovella, *On the relationship between file sizes, transport protocols, and self-similar network traffic*, IEEE ICNP, 1996.

[15] 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.

[16] K. Park and W. Willinger, *Self-similar network traffic: an overview*, In Self-similar network traffic and performance evaluation, J.Wiley & Sons, 2000.

[17] D. Veitch, P. Abry, *A wavelet based joint estimator for the parameters of LRD, Special issue on Multiscale Statistical Signal Analysis and its Applications* IEEE Trans. Info. Th. April 1999, Volume 45, No.3, 1999.

[18] 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.