ECN verbose mode: A statistical method for network path congestion estimation

Rémi Diana, Emmanuel Lochin

ECN verbose mode: a statistical method for network path congestion estimation

Rémi Diana *,[a] Emmanuel Lochin [b,c]

[a] *TéSA/CNES/Thales, Toulouse, France*

[b] *CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France*

[c] *Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France*

Abstract

This article introduces a simple and effective methodology to determine the level of congestion in a network with an ECN-like marking scheme. The purpose of the ECN bit is to notify TCP sources of an imminent congestion in order to react before losses occur. However, ECN is a binary indicator which does not reflect the congestion level (i.e. the percentage of queued packets) of the bottleneck, thus preventing any adapted reaction. In this study, we use a counter in place of the traditional ECN marking scheme to assess the number of times a packet has crossed a congested router. Thanks to this simple counter, we drive a statistical analysis to accurately estimate the congestion level of each router on a network path. We detail in this paper an analytical method validated by simulations which demonstrate the feasibility and the accuracy of the concept proposed and illustrate its use in a realistic scenario. We conclude this paper with possible applications and expected future work.

Key words: Congestion estimation, ECN, measurements

* Corresponding author. Address: ISAE, Campus ENSICA, 1 place Emile Blouin, BP 75064, 31033 TOULOUSE Cedex 5

Part of these results has been presented at IEEE Infocom 2010 Work in Progress track.

Email addresses: remi.diana@isae.fr (Rémi Diana), emmanuel.lochin@isae.fr (Emmanuel Lochin).

1 Introduction

While dropping packets to prevent congestion was considered as a paradox, many studies have shown the undeniable assets of the Explicit Congestion Notification flag [13]. The story starts in 1994 when Sally Floyd shows that this notification allows to increase TCP performances [3] and later in [9], where the authors reach similar conclusion concerning the web traffic. At last, Aleksandar Kuzmanovic in "The Power of Explicit Congestion Notification" [8] investigates the pertinence of ECN and demonstrates once again, that ECN's users will obtain better performances even if all the Internet is not fully ECN-capable.

The following study [10] published in 2004 precises that ECN is only used by 2,1% of computers and that this low percentage can be partly explained by firewall, NAT and other *middle-boxes* of the Internet which reset (without any justification) the ECN flag. However, this is definitely not the main reason. Indeed, although this flag is currently implemented both in end-hosts (GNU/Linux, Mac OSX and Windows Vista) and inside the core network (Cisco IOS implements a RED/ECN variant called WRED/ECN), ECN remains surprisingly disabled by default for all these systems. Concerning end-hosts, this might appear paradoxical. While today CUBIC and Compound TCP variants are enabled by default (respectively in GNU/Linux and Windows Vista) and are still under debate concerning their friendliness with the current Newreno TCP version, a proved mechanism as ECN is not.

We believe this trend has two main reasons: firstly, this is partly due to the behaviour of TCP face to ECN marked packets. Indeed, the goal of the ECN bit is to notify TCP sources of an imminent congestion but this binary indicator does not reflect the real network congestion level. Intuitively, CUBIC and Westwood protocols might better perform than TCP Newreno/ECN due to the nature of the information returned by the ECN binary signal which does not provide any quantitative estimation of the congestion level allowing TCP to efficiently adapt its sending rate [1] . In other words, whatever the number of ECN marked, the TCP reaction is to halve the congestion window and this action is not well adapted to all cases. Secondly, CUBIC and Westwood are pure end-to-end solutions and as a result, are much more easier to deploy while TCP/ECN must involve both the core network and the end-hosts. However, several research work demonstrate that the design of a mechanism to optimally manage network congestion and capacity while being fair with other

[1] We remark that there is a lack of performances evaluation study between ECN-compliant protocols and new proposals such as CUBIC for instance. At least, a recent study clearly shows a clear disequilibrium between TCP Newreno and CUBIC [14].

flows cannot be done without network collaboration [12,4,6]. Unfortunately and to the best of our knowledge, the major barrier is that we do not have today a solution, that do not involve complex computation inside the core routers (such as BMCC [12] or XCP [6]), able to assess at the sender side the exact congestion level of the bottleneck of the path allowing a transport protocol such as TCP to correctly react to this congestion. For instance, BMCC introduces complex mechanisms inside the router and is only compliant with IPv4 (due to the use of the 16 bits `IPid` field of the IPv4 header) while XCP involves large architectural changes.

This fact motivates the present study which proposes a statistical algorithm to assess the congestion level at the end-hosts side (i.e. receiver or sender sides) without involving complex computation inside the core network. In particular, we aim at providing a practical solution to return concrete congestion measurements to the sender in order to avoid blind, approximate or excessive reaction from the source. The only modification deals with the marking method which is changed from a binary field to a count field similar to the TTL field from the IP packet. Practically, we do not have to extend the IP headers as the DiffServ Codepoint field is large enough to enable our proposal. We could argue, as in [12], whether such modification involves or not heavy IETF standardization process, however we claim that it would be much more complex and uncertain to convince networking companies to add complex estimation method inside their own routers. Furthermore, this solution is generic enough to consider, as for ECN, this flag either as a simple binary indicator or as a counter. Finally, we point out that a recent IETF group named ConEx (Congestion Exposure) [11], attempts to enable congestion to be exposed within the network layer of the Internet. The main candidate solution is to date re-ECN [1] and propose the use of a second bit inside the IP header in order to differentiate the congestion upstream and downstream from an observation point inside the network. Internet service providers are pushing this idea as this would provide an essential tool (currently missing) to better manage and control their traffic [2]. If this solution is adopted, we could assist to a larger deployment of the ECN field that would facilitates the deployment of our proposal.

Following this new marking scheme, we propose a simple method which permits an accurate estimation of the congestion level experienced inside the routers of a given path. We first present the mathematical basis of our proposition then, we provide simulations and the practical analysis to evaluate the congestion level. Finally, we discuss and conclude about the possibility offered by this solutions and detail the remaining work.

[2] See the IETF `[re-ecn]` mailing-list and [11] for further details.

2 Marking proposal

The ECN bit, as defined in RFC 3168, is a binary field of the IP header. This field can only contain a boolean value which informs a sender if a packet has crossed at least one congested router. Thus, it is impossible to distinguish a packet marked one time from those marked several times and which would have crossed several congested routers. This prevents any accurate metrology analysis of the link observed for the sake, for instance, of an adapted reaction from the source. In fact, an ECN-capable packet crossing a link composed by two routers and respectively marking at 30% and 40% will have a probability to be marked of 58% (*i.e.* $1-(1-0.4)(1-0.3)$). Obviously, this does not reflect the level of congestion of the network bottleneck (in this example: 40%) and could lead to an excessive reaction from the source. Thus, we propose to enhance the information returned with an incremental field (denoted ECN*) to count how many times a packet is marked. The marking scheme, as for RED/ECN, strictly follows the RED algorithm [5]. We will use this new metric (*i.e.* how many times a packet is marked) to determine the level of congestion of the bottleneck. A RED/ECN* router will increment this counter instead of simply setting the ECN field to one. Through the analysis of the data received, a source can build the distribution of the marked packets.Obviously, we cannot use this metric as it stands, in the following, we present the analytical method to interpret the data collected.

3 Analytical Study

We present in this part the statistical analysis allowing us to process the data collected with our marking proposal. The results obtained allow to establish a relationship between the frequency of ECN* marked packets and the queue size of routers of the path.

3.1 Hypothesis and notations

We consider a topology of n core routers in a row. For $1 \leqslant i \leqslant n$, we note R_i the router number i. All these n routers adopt the previously exposed ECN* marking scheme. Each router drops packets only if its queue is full and probabilistically marks a packet following the average queue size. We call "marking rate" this probability and we adopt the following notations:

- n: number of congested routers;
- p_i: marking rate of the i^{th} router from a path of n routers;

- M_k^n: a packet is marked k times;
- $p(M_k^n)$: the probability of the event M_k^n;
- σ_k^n: the k^{th} elementary symmetric polynomial with n variables. We remind: $\sigma_k^n = \sum_{1 \leqslant j_1 < j_2 < \cdots < j_k \leqslant n} x_{j_1} \cdots x_{j_k}$

3.2 A first simple example : case of two routers

Let's assume a topology of two congested core routers R_1 and R_2 ($n = 2$). In this example, we want to determine the marking rate of both routers with data collected by the sender positioned before R_1. In the same way as standard ECN which uses an ECN echo, the value of the counter ECN* is sent back to the sender with the TCP acknowledgement. Following the previous notations, we call p_1 and p_2 the marking rate of respectively R_1 and R_2. A simple calculation shows that a connection will observe a packet marked with a probability of $1 - (1 - p_1)(1 - p_2)$. Thus, with a standard ECN field, the sender cannot differentiate the two marking rates and so interprets a global congestion which is higher and not representative of the real congestion state. With our proposition ECN*, we refine this information sent back to the sender thanks to the determination of the marking rate of each crossed router. Thus, the sender can determine the level of the bottleneck queue and so could react in a more adapted way to the congestion state. In this example, we can estimate the ratio not only of the marked packets but also of packets marked one and two times. The sender can now estimate $p(M_1^2)$ and $p(M_2^2)$. These values become the new entries of the problem. If we develop these probabilities we have:

$$\begin{aligned} p(M_1^2) &= p_1(1 - p_2) + p_2(1 - p_1) = \sigma_1^2 - 2\sigma_2^2 \\ p(M_2^2) &= p_1 p_2 = \sigma_2^2 \end{aligned}$$

which is equivalent to:

$$\begin{aligned} p(M_1^2) &= \binom{2}{0} \sigma_1^2 - \binom{2}{1} \sigma_2^2 \\ p(M_2^2) &= \binom{2}{0} \sigma_2^2 \end{aligned}$$

Thanks to these equations, the sender can easily determine σ_1^2 and σ_2^2. Thus, using the existing relationship between the polynomial coefficients and the elementary symmetric function of its roots, the sender can evaluate p_1 and

p_2 (here, p_1 and p_2 are the roots of the polynomial $P(x) = x^2$ - σ_1^2 x + σ_2^2). We detail in the following part how to compute in a more general way the polynomial to find the different p_k. Of course, the sender cannot associate each marking rate with the corresponding router but it gets a correct estimation of the congestion level of the bottleneck.

We develop this case as it constitutes the basis of the proof by mathematical induction for the general formula of $p(M_k^n)$. Indeed, when the distribution of the marked packets is done, the crucial step is the deduction of the σ_k^n. To do this, we use the formula of $p(M_k^n)$ and a basic system resolution. Then, as shown in the following part, the determination of the polynomial roots give us the different p_i.

The general formula has the following form:

$$\forall k,\ 1 \leqslant k \leqslant n, \quad p(M_k^n) = \sum_{i=0}^{n-k} (-1)^i \binom{i+k}{i} \sigma_{i+k}^n \tag{1}$$

Proof : To demonstrate (1), we use a proof by mathematical induction. The induction is done on the number of congested routers: n.

Basis: the formula is demonstrated in part 3.2.

Inductive step: $p(M_k^{n+1})$ is the probability for a packet to be marked k times over a path of $n+1$ routers. The event M_k^{n+1} can be decomposed. Indeed, be marked k times over a path of $n+1$ routers is similar to be marked k times by the n first routers and not be marked by the router $n+1$; or to be marked $k-1$ times by the n first routers and be marked by the router $n+1$. In terms of probability, this decomposition can be written as follows:

$$\forall k,\ 1 \leqslant k \leqslant n,\ p(M_k^{n+1}) = p(M_k^n)(1 - p_{n+1}) + p(M_{k-1}^n)p_{n+1} \tag{2}$$

Moreover, we have the following relations:

$$\begin{aligned} \forall k,\ 1 \leqslant k \leqslant n, \sigma_k^{n+1} &= \sigma_k^n + x_{n+1}.\sigma_{k-1}^n \\ \sigma_{n+1}^{n+1} &= x_{n+1}\sigma_n^n \end{aligned} \tag{3}$$

Developing (2) and using (3) we have :

$$\begin{aligned}
\forall k,\ 1 \leqslant k \leqslant n, p(M_k^{n+1}) &= [\sum_{i=0}^{n-k}(-1)^i \binom{i+k}{i} \sigma_{i+k}^n](1-p_{n+1}) \\
&\quad + [\sum_{i=0}^{n-k+1}(-1)^i \binom{i+k-1}{i} \sigma_{i+k-1}^n]p_{n+1} \\
&= \sum_{i=0}^{n-k}(-1)^i C_{i+k}^i \sigma_{i+k}^n - \sum_{i=0}^{n-k}(-1)^i C_{i+k}^i p_{n+1}\sigma_{i+k}^n \\
&\quad + \sum_{i=0}^{n-k+1}(-1)^i C_{i+k-1}^i p_{n+1}\sigma_{i+k-1}^n \\
&= \sum_{i=1}^{n-k}(-1)^i C_{i+k}^i \sigma_{i+k}^n + p_{n+1}C_{i+k}^i\sigma_{i+k}^n \\
&\quad + p_{n+1}\sigma_{k-1}^n + \sigma_k^n + (-1)^{n-k+1}p_{n+1}\sigma_n^n \\
&= \sum_{i=1}^{n-k}(-1)^i C_{i+k}^i(\sigma_{i+k}^n + p_{n+1}\sigma_{i+k}^n) + \sigma_k^{n+1} \\
&\quad + (-1)^{n-k+1}\sigma_{n+1}^{n+1} \\
&= \sum_{i=1}^{n-k}(-1)^i C_{i+k}^i \sigma_{i+k}^{n+1} + \sigma_k^{n+1} + (-1)^{n-k+1}\sigma_{n+1}^{n+1} \\
&= \sum_{i=0}^{n-k+1}(-1)^i \binom{i+k}{i} \sigma_{i+k}^{n+1}
\end{aligned}$$

The formula is so demonstrated for $n+1$. Then, we have :

$$\forall k,\ 1 \leqslant k \leqslant n+1,\ p(M_k^{n+1}) = \sum_{i=0}^{n+1-k}(-1)^i \binom{i+k}{i} \sigma_{i+k}^{n+1}$$

QED

■

3.3 Resolution

Since the formula is now established, we now have to detail the operations a sender has to realize in order to deduce all the marking rates of the congested routers of its path. We detail and recall in this part the different steps mandatory to obtain the result. First of all, thanks to the distribution of the marked

packets, the sender can estimate all the $p(M_k^n)$. Indeed, the $p(M_k^n)$ value is only the ratio between the number of packets marked k times and the total number of received packets by the sender. Moreover, using (1), the sender can compute the σ_k^n. Indeed, if we develop these relations we obtain :

$$\begin{aligned}
p(M_1^n) &= \sigma_1^n - 2\sigma_2^n + \cdots + (-1)^{n-2}\binom{n-1}{n-2}\sigma_{n-1}^n \\
&\quad + (-1)^{n-1}\binom{n}{n-1}\sigma_n^n \\
p(M_2^n) &= \sigma_2^n - 3\sigma_3^n + \cdots + (-1)^{n-2}\binom{n}{n-2}\sigma_n^n \\
\vdots &= \vdots \\
p(M_{n-2}^n) &= \sigma_{n-2}^n - (n-1)\sigma_{n-1}^n + \binom{n}{2}\sigma_n^n \\
p(M_{n-1}^n) &= \sigma_{n-1}^n - (n)\sigma_n^n \\
p(M_n^n) &= \sigma_n^n
\end{aligned}$$

The unknowns are the σ_k^n, we obtain a diagonal system with n equations and n unknowns. The resolution is trivial.

3.3.1 Solving polynomial

As the previous system is solved, all the σ_k^n are known. We now have to deduce the p_i. As said previously, the σ_k^n are elementary symmetric functions. Thus, using the relationship between a polynomial and the elementary symmetric functions of its roots we can deduce the p_i. We detail this step in the following. Let be $P(x)$ a polynomial of degree n, we write $P(x)$ as follows:

$$P(x) = \sum_{m=0}^{n} a_m x^m \tag{4}$$

Let be p_i, $1 \leqslant i \leqslant n$ the n roots of P. Thus:

$$\forall k,\ 1 \leqslant k \leqslant n,\ \sigma_k^n = \sum_{1 \leqslant j_1 < j_2 < \cdots < j_k \leqslant n} p_{j_1} \cdots p_{j_k}$$

Moreover, we have the following relationships:

$$\forall k,\ 1 \leqslant k \leqslant n,\ \sigma_k^n = (-1)^k \frac{a_{n-k}}{a_n} \tag{5}$$

We set $a_n = 1$ in (5). Then (4) becomes:

$$P(x) = \sum_{k=0}^{n} (-1)^{n-k} \sigma_{n-k}^n x^k$$

So, we have a n degree polynomial where the roots correspond to the n marking rates of the n crossed congested routers of the path. We just need now to estimate these roots.

4 Simulation study

In this section, we evaluate our algorithm with data obtained with an ns-2 simulation. This section is divided in three points. First, we present the topology used in the ns-2 simulation and the results. Then, we present the establishment of the solving polynomial and a subtlety for its resolution. Finally, we present our results, a comparison with expected results and a brief discussion about these two last points.

4.1 Tests Topology and gathering of data

The topology used for the tests is given Figure 1. We use TCP/Newreno flows and the reaction of the senders to ECN is disabled. As a result, they do not react with a decrease of their congestion window when they receive an ECN marked acknowledgement. We have implemented our ECN* field and all the RED/ECN* routers use the same parameters: $min_{th} = 50$, $max_{th} = 100$, $max_p = 1$ with a queue length of 100. Concerning the disturbing flows aggregate, an accurate tuning of the senders' emission window has been necessary to simulate a distributed congestion.The analysis of the data is done after 10 minutes when we consider the network stable (this corresponds to a generation of 50000 packets). We analyze the two following TCP flows: the flow #1 from SRC1 to RCV1 and the flow #2 from SRC2 to RCV2. The topology voluntary presents two routers in common to estimate the impact of crossed traffics on our algorithm. To ease the analysis, we first consider that the congestion inside the network is stable in order to obtain a constant marking probability.

In other words, this relative network stability induces a constant congestion level and as a result, a constant average queue size for each router.

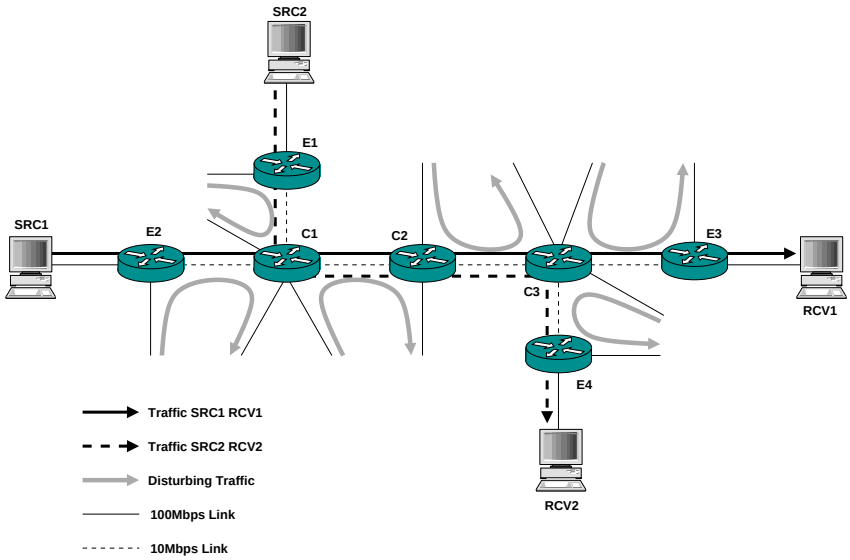
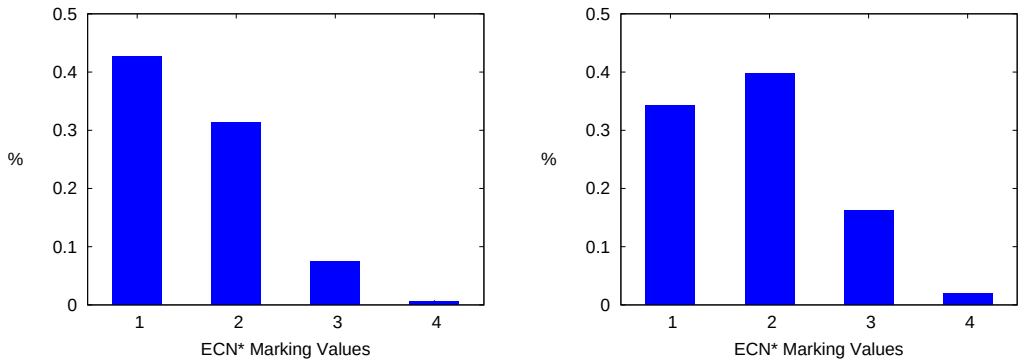


Fig. 1. Topology used for the simulation

The statistic study consists in building the histogram of the distribution of the values of the ECN* marking field for the flows #1 and #2. These results are presented in Figures 2(a) and 2(b).



(a) Results for flow #1 (b) Results for flow #2

Fig. 2. Distribution of ECN* marked packets

4.2 Determination of solving polynomial for flow #1

Figure 2(a) gives the following results (here $n = 4$):

$$\begin{cases} p(M_1^4) = 0.4264 & = \sigma_1^4 - 2\sigma_2^4 + 3\sigma_3^4 - 4\sigma_4^4 \\ p(M_2^4) = 0.3134 & = \sigma_2^4 - 3\sigma_3^4 + 6\sigma_4^4 \\ p(M_3^4) = 0.0738 & = \sigma_3^4 - 4\sigma_4^4 \\ p(M_4^4) = 0.00548 & = \sigma_4^4 \end{cases}$$

We then deduce the following σ_k^4:

$$\begin{cases} \sigma_1^4 = 1.297 \\ \sigma_2^4 = 0.5676 \\ \sigma_3^4 = 0.0957 \\ \sigma_4^4 = 0.00548 \end{cases}$$

By applying the method previously described we have:

$$P(x) = x^4 - 1.297x^3 + 0.5676x^2 - 0.0957x + 0.00548$$

4.3 Practical Resolution

As the solving polynomial is built, we now have to solve $P(x) = 0$. The four roots of $P(x)$ correspond to the four marking rates of the four congested routers crossed by packets arriving to RCV1. As this problem is a stochastic one, we have to consider an uncertainty on the measurements obtained with the simulations. Indeed, unless having an infinite number of packets, we have to consider a drift. We take this possible drift in consideration in the determination of roots of $P(x)$. Basically, we resolve $P(x) = \epsilon$ for $-10^{-3} \leqslant \epsilon \leqslant 10^{-3}$. Thus, we obtain four "areas of roots" instead of "solving roots". We consider that the good value as the middle one. We note ϵ_{min} and ϵ_{max} the extreme values of ϵ from which $P(x) = \epsilon$ have four solutions. Indeed, if we have packets marked four times, we have to determine four solutions of the equation $P(x) = \epsilon$. This condition allows us to determine these four areas of roots.

In our example, we obtain the four following areas of roots: $[0.075, 0.14]$ $[0.14, 0.28]$ $[0.34, 0.50]$ $[0.52, 0.57]$. This allows us to deduce the four following marking rates : 11%, 21%, 42% and 55%. With the same reasoning, we obtain for the flow #2 the four following root areas : $[0.17, 0.23]$ $[0.24, 0.38]$ $[0.40, 0.49]$ $[0.73, 0.74]$ and so the four following marking rates : 20%, 31%, 44% and 74%. These results are presented in the Tab 1.

4.4 Results interpretation

We now compare the results computed with the average queue length of each RED/ECN* routers measured during the simulation. Thus, we can deduce the real marking rate of each RED queue. These results are grouped and presented in the Tab 1. They correspond to the roots computed for the flow #1 and #2 in the previous section 4.3. We note that the observed average queue values have a low standard deviation. These values are almost constant for all the simulation.

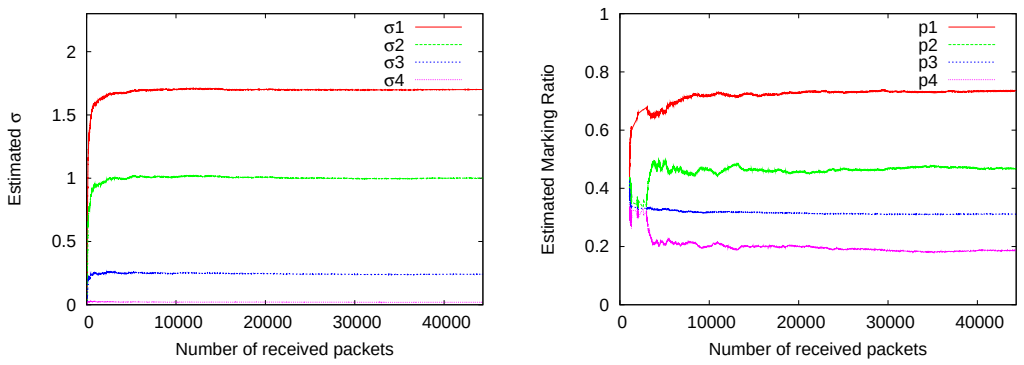
Queue	Average Size (# pkts)	Theoretical Marking Rate	Estimated Marking Rate	
			flow #1	flow #2
Queue1 (E2–C1)	55.5	11%	11%	⊘
Queue2 (C1–C2)	60.5	21%	21%	20%
Queue3 (C2–C3)	72	44%	42%	44%
Queue4 (C3–E3)	77.5	55%	55%	⊘
Queue5 (E1–C1)	65.5	32%	⊘	31%
Queue6 (C3–E4)	87	74%	⊘	74 %

Table 1
Average queue length and corresponding theoretical marking rate

These results globally correspond to the estimations with a slight difference explained by the size of the sample. Moreover, if we do a correlation between the results analytically obtained and those obtained by simulation in table 1, we can notice that flows #1 and #2 estimate two marking rates in common corresponding to the two common routers crossed by both flows. Thus, not only these results correspond to the expected ones but they also underline an important aspect: it seems these measurements are not disturbed with each other and are perfectly independents (several other measurements, not presented here tend to confirm this fact). In other words, this allows to drive several measurements in parallel on a same network. We also verify, thanks to this simulation, that the hypothesis of network stability is sufficient. Thus, if we assume that the path used is relatively constant and the congestion level remains stable, this method allows a good estimation of the congestion level of the different routers of a given path.

4.5 Convergence of this method

As detailed previously, we adopt a probabilistic approach to solve this problem. We admittedly take in consideration the measurement uncertainty by solving $P(x) = \epsilon$. Nevertheless, it is necessary to focus on the convergence time of this solution. It means to assess when the size of the sample is big enough to correctly determine the different marking rates. To do so, we evaluate the different σ, directly linked to the coefficients of the solving polynomial every 50 received packets. The evolution of these coefficients as a function of the number of received packets allow us to determine a threshold from which the value of these coefficients does not evolve anymore. A second threshold can also be set: the one which corresponds to the number of packets from which we can find the solutions to the equation $P(x) = \epsilon$. This approach is presented in figure 3(a).



(a) Evaluation of sigmas as a function of the number of received packets

(b) Computed marking rates as a function of the number of received packets

Fig. 3. Evolution of sigmas and marking rates as a function of received packets

As we can see in the Figure 3(b), we only need 4000 packets to have a correct estimation of the coefficients and about 8000 packets to reach a perfect estimation (equivalent to a 90 seconds transfer in our simulation) with an $\epsilon \pm 10^{-3}$. If we focus on Figure 3(a), we can note that between 3000 and 4000 received packets, the coefficients of the polynomial do not evolve much more. This underlines the accuracy necessary to establish the good solving polynomial. Indeed, we have to accurately estimate the $p(M_k^n)$ to have good results. Other simulations, not presented here, done over a similar topology but with routers less congested, have shown that these thresholds are slightly higher. In fact, the lower is the event corresponding to the marking of a packet, the higher the size of the sample has to be in order to observe this event and so to accurately estimate it. Respectively, the higher is the marking rates (equivalent to an important congestion) the smaller can be the size of the sample.

5 Practical analysis of ECN* to monitor network queues

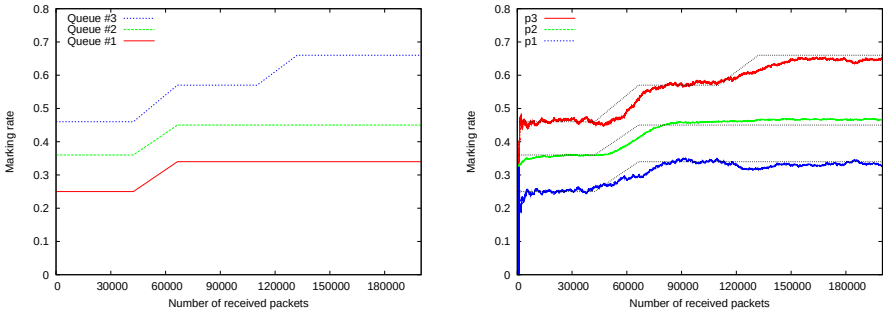
We propose in this part to assess the performances of our algorithm in a more realistic scenario where the load of the network is changing over the time. We first apply the method previously presented and identify the limit of its use. As the success of this algorithm is linked to the size of the statistical sample, we propose to improve our algorithm in order to limit the estimation window length. We have driven several measurements and propose in this section only a selection in order to illustrate the behaviour of our algorithm. Finally, we present the impact of the choice of the estimation window on the pace of convergence to the solution.

5.1 Limit of use of the estimation algorithm

In this first experiment, the analyzis is done every time a packet is received on a sliding window of 20000 packets. This value represents only the maximum statistical sample used to compute the estimation of the marking rates as the algorithm attempts an estimation each time a packet is received. We use the same kind of topology previously presented in Figure 1 except that the number of congested serialized routers between the source (SRC1) and the destination (RCV1) becomes variable. We differentiate *congested routers* from *mute routers* and simulate only the congested routers of the link. Indeed, it is useless to serialize several non-congested routers (i.e. mute routers) as they only increase the end-to-end delay of the path and do not impact on the statistical sample. As a result, by only simulating congested routers, we simulate the case where a flow crosses n routers over a path where only $n-k$ are "severely" congested. We choose to experiment the case where $n-k = (3,4,5,6)$.

We remind that all simulation parameters and particularly the RED parameters remains the same than those defined Section 4.1. We only analyze the traffic between SRC1 and RCV1. The first simulation presented is done with three congested routers between RCV1 and SRC1. The results are presented in Figure 4. Figure 4(a) shows the evolution of the queues occupancy while Figure 4(b) gives the marking rate estimated by the algorithm.

These figures show that after approximatively 1000 packets received (the first correct estimation is exactly given after $x = 1084$ packets in Figure 4(b) as we do not suppress the TCP slowstart phases from the statistical sample), the algorithm is able to follow the evolution of the three marking rates and as a result, the level of congestion of each routers with a good accuracy. In this scenario, the level of congestion is changing over the time with the introduction of



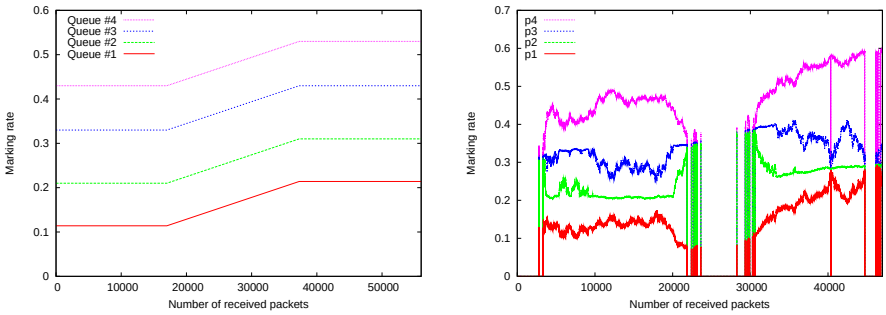
(a) Queues occupation as a function of time

(b) Marking rate estimated

Fig. 4. Results obtained with 3 routers

supplementary disturbing flows. At the beginning, the three congested routers marking rates are 25%, 35% and 46%. Then, the marking rates of all routers increase linearly to 33%, 46% and 56% and finally the marking rate of the most congested router increases up to 67%. If we focus on the marking rates estimated by our algorithm, we can observe the evolution of the estimations while the network conditions evolves. Moreover, the computed marking rates are very closed to the theoretical ones symbolized by the set lines in Figure 4(b). We also remark that the last increase does not impact on the estimation returned for the two other routers.

We now drive another simulation with four congested routers with a different network scenario. As shown in Figure 5(a), the congestion level increases linearly at $t = 200$ seconds. The experiment starts with four stable marking rates of 11%, 22%, 33%, and 44%; then these rates increase linearly up to 21%, 32%, 42% and 53%.



(a) Queues occupation as a function of time

(b) Marking rate computed by RCV1

Fig. 5. Results obtained with 4 routers

The estimations of the marking rates presented in 5(b) are not satisfying. Despite of an estimation of the marking rates returned later (after approximatively 3500 packets) compared to the previous experiment (the explanation comes from the slowstart that is not suppressed from the statistical sample), the main problem shown in this figure is that our algorithm is not able to

provide a solution all the time. Indeed, there are several gaps of thousand packets in which the algorithm provides no estimation of the marking rates (for instance, the largest gap without solution is approximatively between $[24000; 30000]$). Moreover, when the algorithm is able to provide some estimations, they are sometimes inaccurate. Obviously, the estimated marking rates do not correspond to the theoretical ones. This problem persists when we increase the number of router over the path even with a larger estimation window. For instance, we drive an experiment with six routers and obtained no concrete resolution. To solve this problem and to enforce the performances of our algorithm, we detail in the following section an improvement allowing our algorithm to remain resistant whatever the network conditions are.

5.2 Proposed improvement of the solving algorithm

As shown is the previous part, the method explained in the analysis is not satisfying in practice. The reason is that the solving polynomial requires a level of accuracy extremely high. In fact, the roots can be very different if the polynomial coefficients change only slightly. In an obvious manner, estimating accurately the $p(M_k^n)$ is not trivial. In the case of a stable network, we know that the larger is the length of the estimation window, the more accurate are the estimation of the $p(M_k^n)$ probabilities. But in the case of changing network conditions, we cannot use a too large estimation window. So we have to develop a different algorithm to still use the solving polynomial method. This algorithm have to be more robust than the research of roots. It also needs to be fast and to have a low computational complexity. The analysis of the shape of the solving polynomial allowed us to propose a novel method to estimate the marking rates. This algorithm is more robust and requires less accuracy in the estimation of the polynomial coefficients. The method is based on the fact that the marking rates tend to be characteristics points of the polynomial. These representative points are the points where the second derivative of the polynomial is zero. We illustrate this method in the Figure 6.

Of course, these points do not give all the marking rates. We now have to compute the coordinates of the intersection points. Mathematically, this resolution is like considering a linear error in the determination of the polynomial roots. Indeed, instead of solving $P(x) = \epsilon$, we solve $P(x) = \epsilon_1 x + \epsilon_2$. We now have to find the correct ϵ_1 and ϵ_2 values. If we just choose random values and analyze the result, the algorithm would be obviously too complex and would converge slowly. Since the points, where the second derivative is zero, correspond to a solution, this allows us to define ϵ_1 and ϵ_2. Either we can do a linear regression on this representative points to find the ϵ values or we can decide to adopt

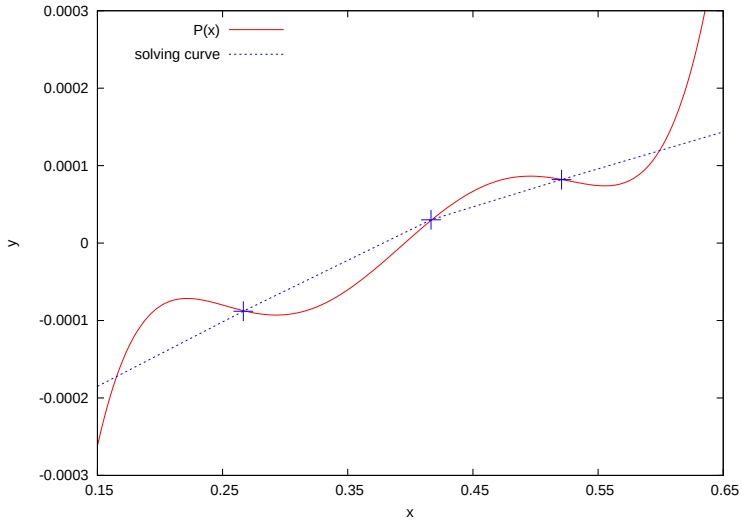


Fig. 6. Second method to solve the $p(M_k^n)$

the broken line scheme[3] presented in Figure 6. We choose the broken line method following several experiments that returned more accurate results. At last but not least, this solution has a lower complexity compared to the linear regression.

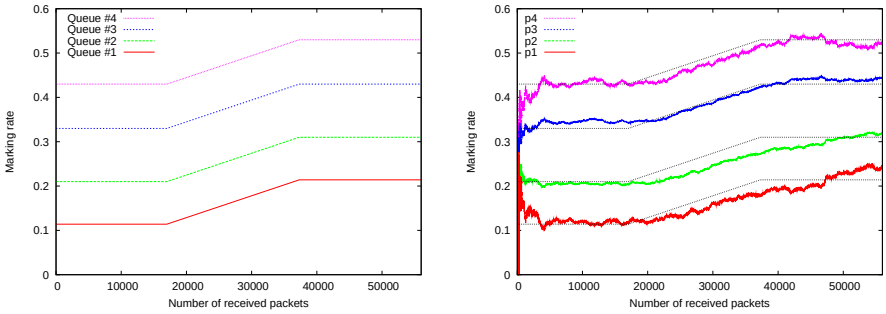
The improved method works on the second derivative of the solving polynomial. As a consequence, the root finding analysis is done on a polynomial where the order corresponds to the number of non mute core routers minus two. The complexity of roots finding analysis has been of interest for several centuries and several studies ahve attempted to propose new algorithms more accurate or faster than previous exiting ones. In particular, the authors in [7] have developed an algorithm of complexity $O(d(logd)^2|log\theta| + d^2(logd)^2)$, where d is the degree of the polynomial and θ the desired precision. At the present time and to the best of our knowledge, this complexity is the best known in terms of degree. Thus, for reasonable values of d and θ, the computation time on today computers is scalable and might be considered as negligible.

5.3 Results with the broken line resolution algorithm

We test our improved algorithm with the previous failed scenario presented in Figure 5. The new results obtained are given in Figure 7

The results obtained with this improved resolution method are unequivocal.

[3] We have intuitively designed this method to accelerate the computation. The core algorithm is similar to a linear regression. Basically, instead of using a straight line to approximate a cloud of coordinates, we use a "broken line" to find the roots. In our case this increases the pace of computation.



(a) Queues occupation as a function of time

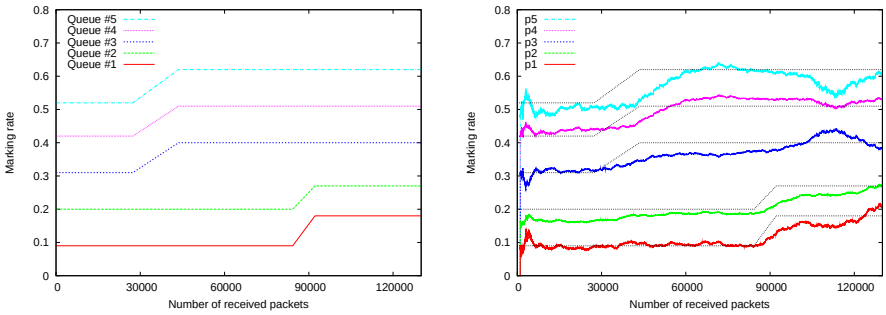
(b) Marking rate computed by RCV1

Fig. 7. Results with 4 routers (new method)

This algorithm greatly enhances the estimation provided. Once again, the maximum errors are about 3% or 4%.

5.4 Simulations with several routers

We now increase the number of congested routers over the path with two scenarios of five and six routers in a row with a moving congestion level scenario. As shown in Figure 8(a), the level of congestion evolves as a function of time and an increase of the level is done during the simulation at $t = 400$ concerning queues #1, #2 and #3, then at $t = 1200$ for both remaning queues. The goal of this experiment is to assess the performance of our algorithm to follow the moving marking rates. The results are presented in 8 where the theoretical marking rates are represented with dotted lines.



(a) Queues occupation as a function of time

(b) Marking rate computed by RCV1

Fig. 8. Results with 5 routers (new method)

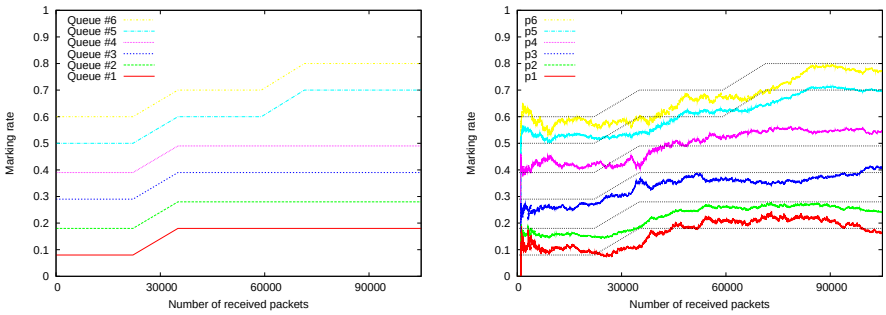
This figure shows that our algorithm provides a good estimation of the marking rates trend. Both most congested queues correctly follows the theoretical marking rates when a change occurs in the congestion.

To conclude these simulations, in the following section we test a topology with six congested routers and investigate the impact of the estimation window

length.

5.5 Impact of the length of the estimation window

The first impact that we want to underline is the delay (time to reach the solution) introduced by the increase of the estimation window. In fact, as the algorithm treats a large set of received packets, it observes the previous state of the network. Thus, in an obvious manner, the larger is the estimation window, the longer is this delay. In the three routers scenario, the estimation of $P(M_k^n)$ is possible with a relatively "small" sample of packets. This explains why the convergence to the solution observed is about 1000 packets. In the five routers experiment, the delay is ranging from 5000 to 10 000 packets with an estimation window of 20 000 packets. To illustrate this convergence pace, we present the simulation with six routers. Once again, the theoretical marking rates are represented with dotted lines in Figure 9.



(a) Queues occupation as a function of time

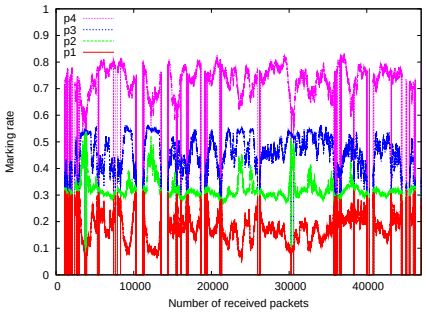
(b) Marking rate computed by RCV1

Fig. 9. Results with 6 routers (new method)

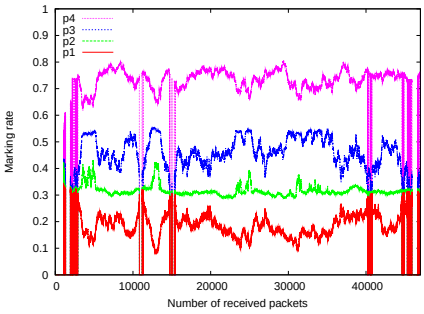
Figure 9(b) shows that if we do not consider the pace of convergence, the marking rates are well followed with a maximum error of 5% and a mean error of 2% or 3%. On the first hand, we can conclude that even with six routers, our analyze allows the receiver to follow accurately the marking rates of each congested routers. On the other hand, the delay is substantial and goes up to 10 000 packets. Indeed, the probability $p(M_k^n)$ is lower when the number of routers increase so their estimation requires a bigger sample set of packets and as a result, a larger estimation window.

The second aspect, already emphasized, is the length needed by the estimation window. In fact, the more the $p(M_k^n)$ are small the more the estimation window needed is large. The value of the $p(M_k^n)$ are both linked to the number of congested routers crossed by the studied traffic and their congestion level. Thus, the $p(M_k^n)$ values decrease with the increase of the number of routers and increase with the increase of the level of congestion of crossed routers. Figure 10 illustrates the impact of choosing a too small estimation window. In the

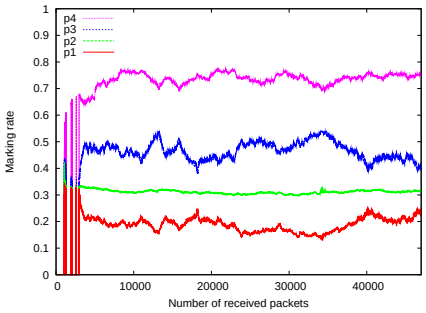
context of realtime monitoring, a compromise is obviously necessary between the accuracy of the computed marking rates and the convergence delay to the solution. In the context of using ECN* to manage a TCP congestion window, this lack of information will not disturb the standard ECN behaviour as TCP will interpret any value of ECN* marked packet as an ECN binary mark. The interaction between ECN* and TCP is reserved for a future work.



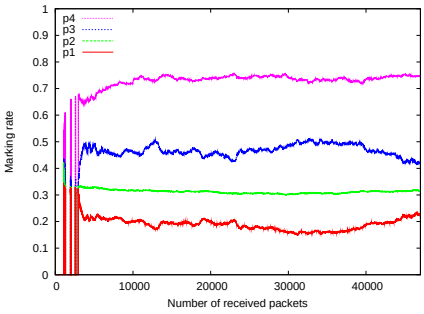
(a) Estimation window of 1000 packets



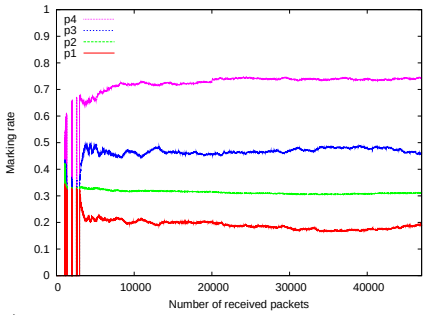
(b) Estimation window of 2000 packets



(c) Estimation window of 5000 packets



(d) Estimation window of 10000 packets



(e) Estimation window of 20000 packets

Fig. 10. Results for different estimation window sizes

6 Discussion on ECN*

In this section, we propose to discuss on the application of the proposed method and in particular, concerning two aspects that might impact on the accuracy of ECN*: the stability of the routing path and the size of the statistical sample.

6.1 On the dynamic nature of Internet route

One of the main assumption to perform a correct estimation with ECN* is the relative stability of the networking path. Dynamic routing is an observed behavior inside the Internet. Although recent measurements studies have shown that Internet routes are relatively stable, the most regular changes are due to load-balancing where ISP spread their traffic to avoid congested links [2]. This re-routing should not have an impact on our algorithm as the resulting new route should normally be congestion free (i.e. we should only cross mute routers, see Section 5.1).

Finally we believe that today, the congested links are mostly at the edge of the network. Thus, ECN* would allow end-host to better react to this direct congestion and would improve the overall performance of the end-user traffic.

6.2 On the size of the statistical sample

As already mentioned before, our method needs a certain statistical sample size to perform a correct estimation of the congestion level. However, when the algorithm cannot perform an estimation, our solution remains ECN-compliant, meaning that an end-to-end protocol is still able to use the binary indication of the ECN field. Thus, we believe that ECN* can be used conjointly with standard ECN congestion signal for long-lived traffic. In other words, ECN* must be seen as a qualitative mechanism that can be used to complement and enhance the accuracy of the standard ECN mechanism when possible.

In this paper, we did not tackle the expected reaction of the E2E protocol as we believe this issue is out of scope of the current paper. Nevertheless, we plan in a new contribution to tackle this aspect and study the interaction between default transport protocol reaction to ECN flag conjointly with ECN*.

7 Conclusion

In this article, we have proposed to increase the level of congestion information returned by TCP feedback messages with an ECN* marking scheme. ECN* enables the ECN field to count how many times a packet has crossed a congested router. We define an algorithm able to estimate the congestion level of each queue of a given path through the analysis of the data collected and demonstrate the existing relationship between this ECN* marking rate and the filling level of each routers' queue. Simulation results suggest that this method is reliable and robust to cross traffics. In order to illustrate the use of this method, we have also investigated its performance in the context of a network with a varying congestion level. We show that we can still assess the level of congestion of each routers witn a slight improvement of our algorithm allowing to faster converge to the solution. Furthermore ECN* remains ECN compliant. It means that an ECN* marked packet is always interpreted by a TCP ECN-compliant flow as a binary mark. We are convinced by the benefits that we would get by implementing such simple marking scheme inside IP, thus we propose in a future work to use this congestion signal conjointly with TCP following the method proposed in [12] and to present this scheme to the Conex IETF working group in order to allow the implementation of such simple counter.

References

[1] B. Briscoe, A. Jacquet, T. Moncaster, and A. Smith. Re-ECN: Adding accountability for causing congestion to TCP/IP. Internet draft, Internet Engineering Task Force, July 2008.

[2] Fabien Viger et Al. Detection, understanding, and prevention of traceroute measurement artifacts. *Elsevier Computer Networks Journal*, 2008.

[3] S. Floyd. TCP and explicit congestion notification. *ACM Comp. Comm. Review*, 24(5):10–23, 1994.

[4] S. Floyd. HighSpeed TCP for Large Congestion Windows, December 2003. Request for Comments 3649.

[5] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transaction on Networking*, 1(4):397–413, August 1993.

[6] D. Katabi, M. Handley, and C. Rohrs. Congestion control for high bandwidth-delay product networks. *SIGCOMM Comput. Commun. Rev.*, 32(4):89–102, 2002.

[7] Myong-Hi Kim and Scott Sutherland. Polynomial root-finding algorithms and branched covers. *Society for Industrial and Applied Mathematics (SIAM) Journal*, 1991.

[8] A. Kuzmanovic. The power of explicit congestion notification. *SIGCOMM Comput. Commun. Rev.*, 35(4):61–72, 2005.

[9] L. Le, J. Aikat, K. Jeffay, and F. Smith. The effects of active queue management on web performance. In *In Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[10] A. Medina, M. Allman, and S. Floyd. Measuring the evolution of transport protocols in the internet. *Computer Communication Review*, 35(2), April 2005.

[11] T. Moncaster, L. Krug, M. Menth, S. Blake, and R. Woundy. The need for congestion exposure in the internet. Internet draft, Internet Engineering Task Force, October 2009.

[12] I. Qazi, L. Andrew, and T. Znati. Congestion control using efficient explicit feedback. In *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, 20-25 Apr 2009.

[13] K. Ramakrishnan and S. Floyd. A Proposal to add Explicit Congestion Notification (ECN) to IP, January 1999. Request for Comments 2481.

[14] L. Stewart, G. Armitage, and A. Huebner. Collateral damage: The impact of optimised TCP variants on real-time traffic latency in consumer broadband environments. In *IFIP Networking 2009*. Springer, 2009.