Regulation of a Connection Admission Control Algorithm

Thorsten Kurz, Patrick Thiran, Jean-Yves Le Boudec

Institute for Computer Communication and Applications (ICA)
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne
Switzerland
http://icawww.epfl.ch

***Abstract* - Connection Admission Control (CAC) algorithms are used to decide whether an incoming connection should be accepted or rejected in a node of a network offering reservation based services in order to maintain the guaranteed Quality of Service (QoS) in the network. In this paper, we consider the statistical CAC algorithm proposed by *Elwalid et al.* [2]. The traffic model is made of ON-OFF sources and the QoS parameter is the loss probability. Based on the traffic descriptors of existing and incoming connections, the algorithm takes its decision by computing an upper bound of this probability and checking whether it is larger than a given tolerance ε.**

Usually this tolerance is a fixed, given parameter. We propose here to adapt ε to react to the actual losses experienced at the node using a simple regulation mechanism: if the actual loss rate is much smaller than the targeted loss rate, ε is increased to make a more aggressive usage of the available resources, and vice versa if the actual loss rate is too high. We discuss the influence of the regulation parameters and we show that despite its simplicity this regulated CAC improves significantly the performance of its non-tunable counterpart.

I INTRODUCTION

We consider a network offering reservation-based services. Connection Admission Control (CAC) algorithms are used to decide whether an incoming connection should be accepted or rejected in a network node in order to maintain the guaranteed Quality of Service (QoS). In this paper, we consider the statistical CAC algorithm proposed *by Elwalid et al.* [2] and described in Section II. The traffic model is made of periodic ON-OFF sources and the QoS parameter is the loss probability. Based on the traffic descriptors of existing and incoming connections, the algorithm takes its decision by computing an upper bound of this probability and comparing it with a given tolerance ε.

Usually this tolerance is a fixed, given parameter. In Section III we propose here to adapt ε to react to the actual losses experienced at the node using a simple regulation mechanism: if the actual loss rate is much smaller than the targeted loss rate, ε is increased to make a more aggressive usage of the available resources, and vice versa if the actual loss rate is too high. We discuss the influence of the regulation parameters and we show that despite its simplicity this regulated CAC significantly improves the performance of its non-tunable counterpart.

Measuring the actual data loss rate and using this value for a regulation of the CAC parameter ε allows us to focus on the real loss rate instead of the upper bound of the probability of a buffer overflow. A further advantage of the regulation is that we can react to changing traffic patterns and thereby always provide an optimal usage of available resources to the service provider while maintaining the guaranteed QoS for the customer. In Section III we describe this regulation scheme and we show that it improves the performance of the CAC algorithm introduced in Section II.

II CAC ALGORITHM

If a connection is not active continuously, its idle time can be used to transfer data of other connections. A buffer is used to store data temporarily if several connections are sending at the same time. Nevertheless if the buffer is already full, data will be lost. The task of a CAC algorithm is to decide whether an incoming, new connection should be accepted or rejected so that the loss probability is kept below a maximum loss tolerance ε defined by the service provider.

There are various CAC algorithms with different complexity and performance. In this paper we have chosen a simple CAC algorithm developed by *Elwalid et al.* [2].

In this Section the maximum loss tolerance ε is considered as fixed and the CAC algorithm ensures that the actual loss probability does never exceed it, thereby fulfilling the quality of service (QoS) requirements to the user. However we will see below that the actual loss rate can be much below this loss tolerance, leaving space for some "overbooking". In this case ε becomes a parameter, a "gauge", which is tuned so that the actual loss rate remains indeed below (but not too much below) the targeted loss rate.

In this paper, we consider a buffered trunk offering a bandwidth C and a buffering capacity X to the input traffic. The input traffic is made of J connections, described by their traffic descriptors p_j (peak rate), m_j (sustainable rate) and b_j (burst size), for $1 \le j \le J$. The arrival curve [1] of each connection is therefore given by $\alpha(t) = \min\{p_j t,\ m_j t + b_j\}$.

As in [2], we consider here periodic ON-OFF traffic sources, sending at rate p_j during $T_{ON\,j}$ and then idle during period $T_{OFF\,j}$, for $1 \le j \le J$.

0-7803-5417-6/99/$10.00 ©1999 IEEE.

For such sources, we have the relations

$$m_j = \frac{T_{ON_j}}{T_{ON_j} + T_{OFF_j}} \cdot p_j$$

$$b_j = (p_j - m_j) \cdot T_{ON_j} = \frac{T_{ON_j} \cdot T_{OFF_j}}{T_{ON_j} + T_{OFF_j}} \cdot p_j$$

for $1 \le j \le J$. The phases of each connection are statistically independent from each other.

The CAC proposed by *Elwalid et al.* [2] consists in allocating to each connection a fraction c_{0j} of the total bandwidth C and a fraction x_{0j} of the total buffer X as follows:

$$c_{0j} = \begin{cases} \dfrac{p_j}{1 + \dfrac{X}{C} \cdot \dfrac{p_j - m_j}{b_j}} & \text{if } m_j < b_j \cdot \dfrac{C}{X} \\ m_j & \text{if } m_j \ge b_j \cdot \dfrac{C}{X} \end{cases} \qquad (1a)$$

$$x_{0j} = \frac{m_j}{c_{0j}} \qquad (1b)$$

With this allocation, an upper bound (*Chernoff's* bound) for the probability of buffer overflow is then computed [2]

$$P_{congestion} \le \exp\{-\mathrm{F_k}(s^*)\} \qquad (2)$$

where

$$\mathrm{F_k}(s^*) = \sup_{s \ge 0} \left[sC - \sum_{j=1}^{J} \ln\{1 - x_{0j} + x_{0j} \exp(sc_{0j})\} \right]. \qquad (3)$$

For all new incoming connections, the CAC computes the upper bound (2) for the existing connections and the new connection and checks whether this bound is below a given tolerance ε. In this case the incoming connection is accepted, otherwise it is refused.

A less conservative decision can be taken by using the *Bahadur-Rao* approximation instead of *Chernoff's* bound, which reads

$$P_{congestion} \approx \frac{e^{-\mathrm{Fk}(s^*)}}{s^* \sigma(s^*) \sqrt{2\pi}} \qquad (4)$$

where

$$\sigma(s^*) = \sqrt{\sum_{j=1}^{J} \frac{(1 - x_{0j}) x_{0j} c_{0j}^2 \exp(s^* c_{0j})}{(1 - x_{0j} + x_{0j} \exp(s^* c_{0j}))^2}} \qquad (5)$$

while $\mathrm{F_k}(s^*)$ is given by (3), as before.

However both *Chernoff's* bound (*2*) and *Bahadur-Rao's* approximation (*4*) apply to *the probability of congestion* (that is the probability that the buffer is full), whereas the quantity of interest to the user is the *loss rate* (that is, the fraction of traffic that has been lost per time unit). We therefore need a probability that a given data unit gets lost. For a bufferless node, the probabilistic bounds obtained by large deviation techniques turn out to be significantly lower than (4), of 2 orders of magnitude (See *[4]* p. 154). For the buffered node, we have adopted a scaling factor γ larger than $\sqrt{2\pi}$ in (*4*):

$$P_{loss} \approx \frac{e^{-\mathrm{Fk}(s^*)}}{s^* \sigma(s^*) \gamma} \qquad (6)$$

Experimentally we found that $\gamma = 200$ was a good choice. The CAC regulation described in the following Section will anyway adapt this acceptation policy to the current loss rate, so that the computation of an accurate estimate of the loss probability is not imperatively needed.

Consequently, the CAC algorithm that has been finally implemented in our regulation system enforces the following policy: A new incoming connection is accepted if, based on (1) and (3) computed for all existing connections and the new one, one has that $\frac{e^{-\mathrm{Fk}(s^*)}}{s^* \sigma(s^*) \gamma} \le \varepsilon$ and is rejected otherwise.

III CAC–REGULATION

A. Regulation Algorithm

The CAC algorithm takes its decision of acceptance or rejection of an incoming connection so that the loss rate is kept below a given value, the target loss rate.

The actual loss rate may be well below this target loss rate, because the formulae in the previous Section are upper bounds or approximations. In this case the resources (bandwidth, buffer) are not optimally used.

This drawback can be eliminated by adapting the loss tolerance ε to the actual loss rate. Such a regulation scheme is proposed in this paper. The goal is to deliver the expected QoS to the customers, i.e. to keep the actual loss rate below the target loss rate, while ensuring a better usage of the resources for the service provider. The regulation system is shown in Fig. 1.

The loss rate is measured over a time window of T_l units of time, and the controller provides a new value of the tolerance ε every T_l units of time. Therefore T_l is the sampling time of the continuous-time CAC/buffered node system. The controller will be developed in discrete time.

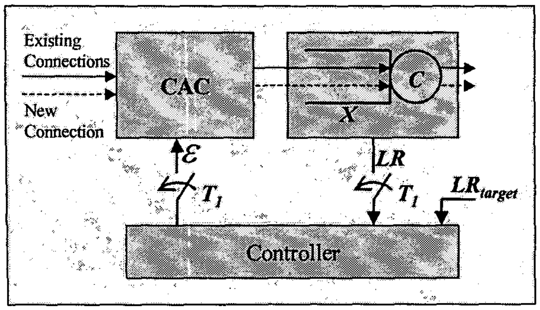


Fig. 1: The regulation system

Let $LR(t)$ denote the actual loss rate at time t, which is the input to the controller, and let LR_{target} denote the targeted loss rate, i.e. the loss rate which is agreed upon between the service provider and its customers. If the loss rate $LR(t)$ is higher than LR_{target}, the QoS begins to deteriorate and the contract between the service provider and its customer is no longer respected, whereas if $LR(t)$ becomes much smaller than LR_{target}, the potential increase in QoS is considered as being insignificant and is not perceived by the user.

The goal of the controller is therefore to adapt ε so that the magnitude of the error signal

$$\Delta L(t) = LR(t) - LR_{target}$$

is kept as small as possible.

As we will discuss in the next Section this signal is highly fluctuating, so that some low pass filtering is needed. Moreover, a simple regulation scheme is preferred over a complex one. These reasons motivated us to adapt a regulator be-longing to the PI (Proportional-Integral) family. The fast varying signal $LR(t)$ prevented us from taking a PID (Proportional-Integral-Derivative) controller for stability reasons.

The first part of the regulator is a discrete-time first-order low-pass filter, with one pole $0 < \beta < 1$. The filtered signal $\Delta L'(t)$ is given by

$$\Delta L'((k+1)T_l) = \beta\, \Delta L'(k\,T_l) + \Delta L(kT_l)\ . \tag{7}$$

Note that this filtered signal appears as an exponentially weighted average of the error signal, as the solution of (7) is

$$\Delta L'(k\,T_l) = \sum_{l=0}^{k} \beta^{(k-l)Tl}\, \Delta L(l\,T_l)\ .$$

The halfrate time of the exponential weight is denoted by

$$T_h = -\frac{\ln 2}{\ln \beta}\quad \text{, i.e.}\quad \beta = \sqrt[T_h]{\tfrac{1}{2}}\ .$$

The second part of the regulator is then the incremental adaptation of the tolerance ε proportionally to the filtered error signal

$$\varepsilon((k+1)T_l) = \varepsilon(k\,T_l) + k(1-\beta)\Delta L'((k+1)T_l)\ . \tag{8}$$

There are therefore two control parameters to adjust for each regulation equation (7) and (8): The *pole* β or equivalently the half rate time T_h of the weighting exponential function used in the filtering step (7), and the *gain* k with which a modification must be brought to the old value of ε in (8).

The larger T_h and the larger β, the smoother is the ε-curve. The larger k, the stronger is the reactions of ε to a variation in ΔL. Table I shows the 4 parameters that can affect the control performance. T_l and LR_{target} are parameters given a priori, that have however a strong influence on the regulation scheme, as briefly discussed in the next Section. T_h and k are control parameters that need to be set adequately. This is also discussed in the next Section.

B. *Optimization of Regulation Parameters*

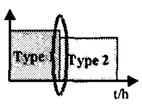
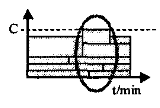
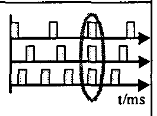
Three reasons for variations in loss rates can be qualitatively distinguished, as shown in Table II.

1. *Global changes of traffic types on the link*, as shown in the left column of Table II. These changes occur at low frequency and represent different persistent aggregated rate and/or a different burstiness of the traffic patterns, i.e. different peak rates, sustainable rates and burst sizes of the connections.

TABLE I
PARAMETERS AFFECTING THE REGULATION

Para-meter:	Description:	Effects:
T_l	Time interval for taking measure-ments. Smallest time step over which data are sampled.	Sets the sampling rate for network regulator.
T_h	Size of the weighted sliding window of regulation.	Changes the reaction speed of the regu-lation.
k	Gain.	Changes the reaction strength of the regu-lation
LR_{target}	Targeted loss rate.	ε is regulated to maintain the actual loss rate close to LR_{target}.

TABLE II
THREE REASONS FOR CHANGES OF THE LOSS RATE

	Change of traffic type	Critical connection constellations	Random fluctuations
			
Time scale:	Minutes to hours	Seconds to minutes	Milliseconds to seconds
Fre-quency:	low	medium	high
Reac-tion:	yes	yes	no

2. *Transient critical configurations of admitted connections*, as shown in the middle column of Table II. Usually the amount of buffer and bandwidth, which has not been allocated to the existing connections, can absorb bursty incoming data arriving simultaneously from some existing connections. Sometimes all buffer and bandwidth resources have been allocated to connections, so that no resource is any more available for these bursty data, which results in a temporary increase of the loss rate. Such fluctuations occur in the medium range frequency.

3. *Transient rapid and random fluctuation of losses*. These changes are due to an overleap of ON-states of several connections for a short period of time. They occur at a very high frequency and in a random fashion so that they are assimilated to noise.

The regulation should compensate for the first two fluctuation patterns, and ignore as much as possible the third one. The influence of the different parameters is as follows:

T_l: Since the loss rate is computed by averaging the losses over a time interval of length T_l, a small interval will result in a very bursty loss rate, close to the instantaneous rate, and will be quite noisy. Conversely a too large interval will smooth out important variations of the signal.

T_h: The *pole* β determines the cut-off frequency of the filter of the loss signal. Equivalently, T_h has to be chosen large enough to eliminate the noisy fluctuations, and small enough to discover the changes of traffic type and critical configurations of admitted connections.

k: The *gain* must be large enough to react promptly to a change in $\Delta L'$, but not too large to avoid the emergence of undesired oscillations.

LR_{target}: The range of values for LR_{target} can be of order of magnitudes of difference, and this affects considerably the loss curves. With a value of LR_{target} ten times smaller, a smooth curve *LR(t)* becomes a sequence of impulses. This changes completely the regulation design, and will be discussed at the end of Section III.

C. *Performance of the proposed regulation scheme*

To determine how well the proposed adaptation works, and how parameters k and T_h need to be chosen, a series of tests have been performed for one link of bandwidth C=155MBps and buffer X=1Mbit.

Connections are made of traffic of two different connection types: connections of Type 1, characterized by an ON period chosen randomly with a uniform distribution between 100 and 150ms, an OFF period chosen randomly between 150 and 220ms and a Peak rate chosen randomly between 3.2 and 4.6Mbps. Connection of Type 2, characterized by an ON period chosen randomly with a uniform distribution between 46 and 68ms, an OFF period chosen randomly between 320 and 460ms and a Peak rate chosen randomly between 6.8 and 10Mbps. Two typical traffic sources of each type are shown in Fig. 2.

The aggregate traffic pattern is made alternatively of each kind of sources, changing in a two-hour period for 10 hours of simulation time. The total sustainable rate is also changing on a 2 hours scale, varying between an average of 160Mbps for those periods where connections type 1 are active to 140Mbs for periods where connections of type 2 are active (C.f. Fig. 3). This is done to take the different degrees of overbooking into account.

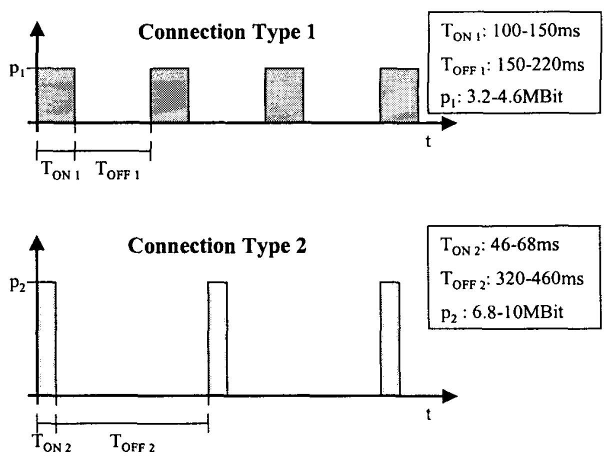


Fig. 2: The two different connection types for the test traffic pattern

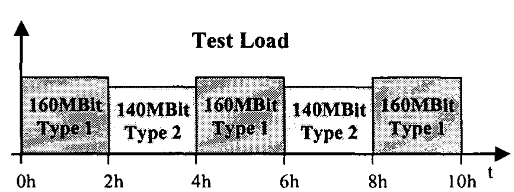


Fig. 3: The test traffic pattern

The test traffic pattern is by no means limited to two alternating connection types. The existence of many very different connection types makes in fact the task of the controller a bit easier, as critical constellations appear less likely. The traffic pattern of Fig. 3 makes the job of the controller more difficult, which is the reason why it has been adopted.

To compare the performance of the regulated CAC algorithm, different quantities of interest are

Loss rate (LR): This quantity is of primary importance. The average value over 10 hours of simulated time is reported but is not very indicative. Indeed, the figures for the non regulated CAC are very similar to those of the regulated CAC, as in the first case the loss rate is sometimes much too high, sometimes much too low, although giving the same average value as seen in Fig. 4. Therefore an estimation of the variance of the loss rate has been also computed for the whole time series, and is an essential parameter to assess the quality of the CAC regulation.

Rejected connection rate (RCR): This quantity is defined as the rate between rejected connections and requested connections. It depends not only on the performance of the CAC, but also on the actual traffic.

Bandwidth usage (SBR): This quantity is defined as the sum of all declared sustainable rates (m_i) divided by the available bandwidth C:

$$SBR = \frac{\sum_{\substack{i=admitted \\ connections}} m_i}{C}$$

Again this value depends also on other factors than the performance of the CAC regulation. The traffic parameters have an considerable influence on the maximum *SBR* the CAC regulation can achieve.

While the average figures of the *LR* and the *SBR* are inappropriate for a performance evaluation, the comparison of the time series for the *RCR* and the *SBR* in Fig. 4 and Fig. 5 gives a good illustration of the gain achieved by a CAC with feedback over an open loop CAC.

For the first series a targeted loss rate of $5.5*10^{-4}$ was set. Table III shows the results found for this range, T_l being equal to 1 second.

The usage, the rejected connection rate and the loss rate do not differ significantly for different parameters k and T_h, because, as discussed before, they are average values, which produce the same results by using a constant ε whose value is equal to the average value of the varying ε obtained with the regulation feedback (cf. Table IV).

TABLE III
REGULATION RESULTS FOR DIFFERENT T_h AND k AND $LR_{target} = 5.5*10^{-4}$

Results after 10h:		$k = 0.2$	$k = 0.4$	$k = 0.5$	$k = 0.6$	$k = .7$	$k = 0.8$
$T_h =$ 0.25	Mean(*SBR*):	77.1%	77.2%	77.2%	77.2%	77.1%	76.9%
	Mean(*RCR*):	19.9%	19.9%	20.0%	20.0%	20.3%	20.2%
	Mean(*LR*):	5.2e-4	5.4e-4	5.5e-4	5.5e-4	5.5e-4	5.6e-4
	Var(*LR*):	2.2e-7	2.1e-7	2.1e-7	2.1e-7	2.2e-7	2.4e-7
$T_h = 0.5$	Mean(*SBR*):	77.0%	77.1%	77.2%	77.2%	77.1%	77.0%
	Mean(*RCR*):	19.9%	20.1%	20.1%	19.8%	19.9%	20.4%
	Mean(*LR*):	5.2e-4	5.4e-4	5.5e-4	5.5e-4	5.5e-4	5.6e-4
	Var(*LR*):	2.2e-7	2.1e-7	20e-7	2.0e-7	2.1e-7	2.3e-7
$T_h = 1.0$	Mean(*SBR*):	77.0%	77.2%	77.2%	77.2%	77.1%	76.9%
	Mean(*RCR*):	20.1%	19.9%	19.9%	19.7%	20.1%	20.5%
	Mean(*LR*):	5.2e-4	5.4e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4
	Var(*LR*):	2.4e-7	2.1e-7	2.1e-7	1.9e-7	2.2e-7	2.3e-7
$T_h = 2.0$	Mean(*SBR*):	77.0%	77.1%	77.0%	76.9%	77.0%	77.1%
	Mean(*RCR*):	20.2%	20.2%	20.2%	20.5%	20.3%	20.1%
	Mean(*LR*):	5.2e-4	5.4e-4	5.5e-4	5.5e-4	5.6e-4	5.6e-4
	Var(*LR*):	2.6e-7	2.3e-7	2.2e-7	2.1e-7	2.2e-7	2.3e-7
$T_h = 4.0$	Mean(*SBR*):	77.0%	76.6%	76.5%	76.3%	76.2%	75.9%
	Mean(*RCR*):	20.3%	21.1%	20.9%	21.1%	21.4%	21.6%
	Mean(*LR*):	5.2e-4	5.6e-4	5.6e-4	5.6e-4	5.6e-4	5.7e-4
	Var(*LR*):	2.5e-7	2.5e-7	2.6e-7	2.6e-7	2.7e-7	2.7e-7

But without regulation the values of the loss rate are either too low or too high as clearly shown by comparing Fig. 4 and Fig. 5 and reflected quantitatively by the fact that the variance is much lower in the second case with regulation. The bandwidth is also much better exploited as can be seen by comparing the bandwidth usage (*SBR*) at the bottom of Fig. 4 and Fig. 5.

The performance of different regulations can only be measured by comparing variance of the loss rate, which tells how well the loss rate conforms to the targeted QoS. In this case we see that the variance was reduced from $3.7*10^{-7}$ in the case of using no regulation to $1.9*10^{-7}$ in case of the best-found regulation, with T_h=1 and k=0.6 (cf. Table IV).

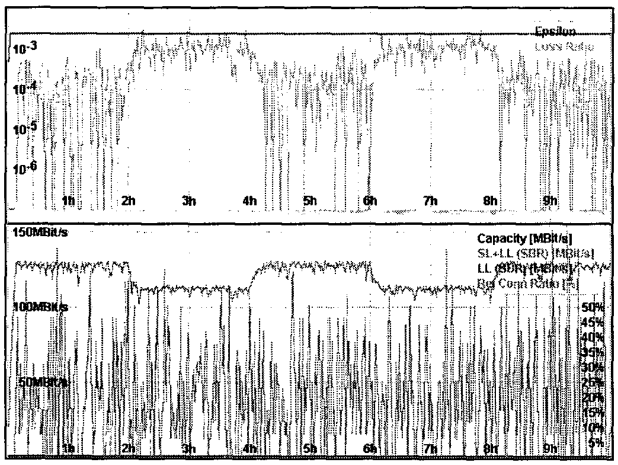


Fig. 4: Time series for a constant ε = 0.0025 without regulation: (top:) ε and *LR(t)*, (bottom:) SBR(t) and RCR(t).

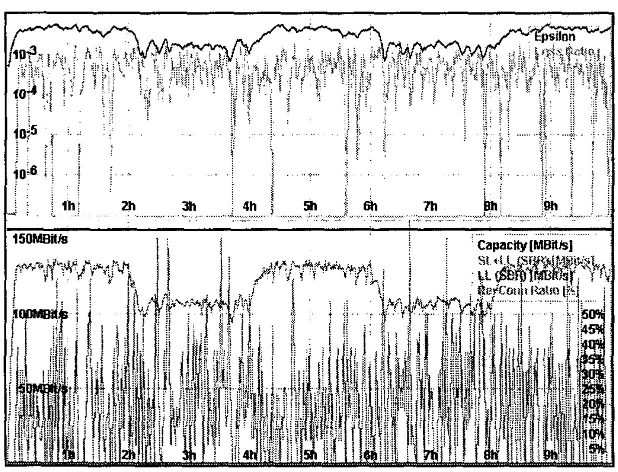


Fig. 5: Time series of the best-found regulation with T_h=1.0 and k=0.6: (top:) ε *(t)* and *LR(t)*, (bottom:) SBR(t) and RCR(t).
The reaction is just optimal for fluctuations with low and middle frequency. Comparing the loss rate and bandwidth to Fig.4 shows clearly the advantages of the CAC regulation

The effects of changing the parameters T_h and k are shown in Fig. 6 to Fig. 9. The results show that some reactivity to short-term changes is useful to prevent more connections to be established in an already critical situation. Too much reactivity however results in nervous reactions to all random fluctuations. (Cf. Fig. 6 and Fig. 7). Also certain strength of reaction is required, but too much strength results in oscillations. (Cf. Fig. 8 and Fig. 9.)

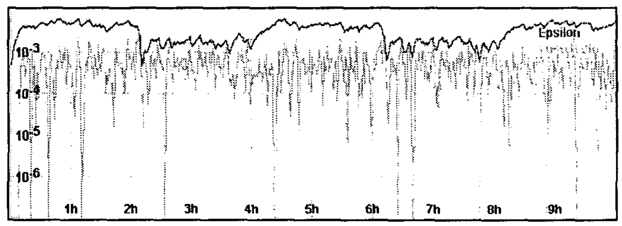


Fig. 6: Time series of ε *(t)* and *LR(t)* for T_h=0.25 and k=0.6: The smaller sliding window results in more nervous reactions.

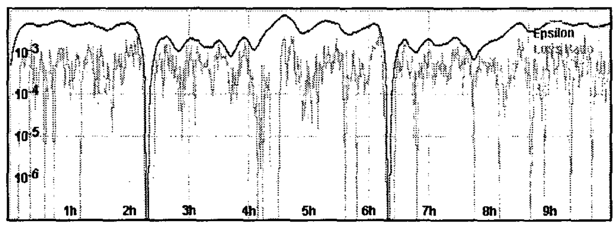


Fig. 7: Time series of ε *(t)* and *LR(t)* for T_h=4.0 and k=0.6: The larger sliding window introduces too much inertia to the reactions.

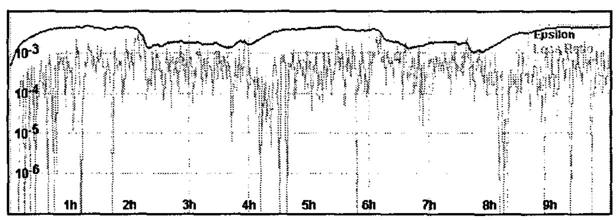


Fig. 8: Time series of ε *(t)* and *LR(t)* for T_h=1.0 and k=0.2: The smaller integral constant makes reactions smooth, but weak.

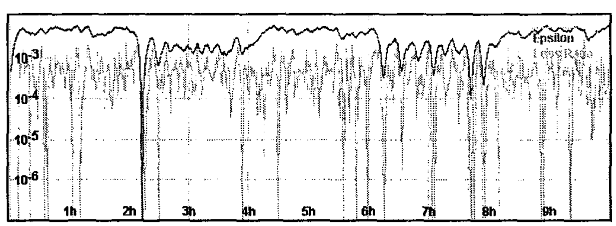


Fig. 9: Time series of ε *(t)* and *LR(t)* for T_h=1.0 and k=0.8: The larger integral constant provokes oscillations in the regulation.

TABLE IV

NO REGULATION VS. BEST REGULATION FOR $LR_{target} = 5.5*10^{-4}$.

	Constant ε = 0.0025	T_h = 1.0 k = 0.6
Mean(*SBR*):	77.2%	77.2%
Mean(*RCR*):	19.8%	19.7%
Mean(*LR*):	5.5e-4	5.5e-4
Var(*LR*):	3.7e-7	1.9e-7

In a second series a targeted loss rate of $5.5*10^{-5}$ was set. Because of the small targeted loss rate, the losses appear not anymore as a continuous function of time, but purely in the form of punctual shots. Now the difference of the variances between the case of no regulation and the best-found regulation with T_h=1 and k=0.5 is very small ($1.2*10^{-8}$ to $1.0*10^{-8}$, cf. Table V), while the graphs (Fig. 10 and Fig. 11) still indicate a slight improvement achieved by the regulation.

IV CONCLUSION AND FUTURE WORK

We have proposed and developed a regulation scheme of the loss tolerance of a CAC algorithm. It enables the CAC to keep the actual loss rate close to its targeted value defined by the user. By preventing the loss rate to take too large values, a service provider ensures that the required QoS is provided

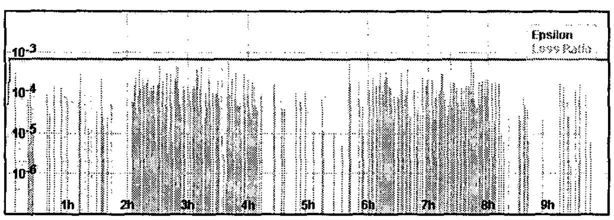


Fig. 10: Time series of $LR(t)$ for a constant ε = 0.00065 without regulation

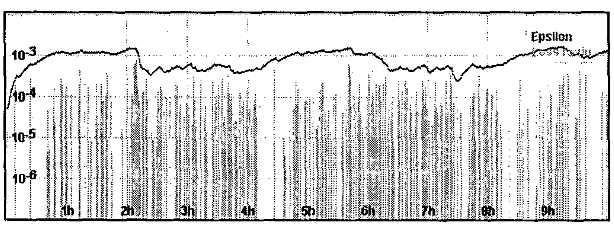


Fig. 11: Time series of ε (t) and $LR(t)$ for T_h=1.0 and k=0.5: Good parameters decrease the variance of losses.

TABLE V

NO REGULATION VS. BEST REGULATION FOR $LR_{target} = 5.5*10^{-5}$.

	Constant ε = 0.00065	T_h = 1.0 k = 0.5
Mean(*SBR*):	69.9%	70.1%
Mean(*RCR*):	22.5%	22.7%
Mean(*LR*):	5.0e-5	5.1e-5
Var(*LR*):	1.2e-8	1.0e-8

to the users. By increasing the degree of overbooking when the loss rate tends to drop much below the QoS target, the regulator allows the service provider running the network to make the best usage of the resources.

The regulation algorithm remains very simple and can easily be implemented as it effects only one parameter of the CAC module. We have also determined by simulation the best control parameters.

The sampling rate, at which the losses are measured, and the value of the target loss rate have a strong influence on the performance scheme as shown in this paper. In particular, for low (and hence realistic) values of the targeted loss rate, the loss rate signal becomes a sequence of impulses that is difficult to manage with the current regulator. This suggests to use a virtual system with fewer resources (i.e. smaller buffer and/or bandwidth) as input for taking control action on the real system. Network calculus [2] will be used for dimensioning such systems adequately.

The graphs and the results show that a CAC-Regulation can be configured to react to changes of traffic types and to transient critical situations on a link, while ignoring the random fluctuations of the loss rate. While visual impression of quality of the regulation given by the graphs can sometimes be misleading or at least hard to estimate, the variance of the loss rate provides a quantitative measure with which different regulations can be easily compared. Further work will include the refinement of the variance estimation.

The assumption of ON-OFF sources has been taken for the sake of using an existing CAC algorithm, and as it is considered to be one of the worst kind of traffic in the literature. The regulation scheme is even better suited to situations where the traffic is quite different from the worst case or – contrary to a non regulated CAC algorithm – it will do some overbooking if the sources send less than their declared arriving curve envelope permits them to do. It will make the acceptance region return safely to more conservative values if the sources begin to send more.

The regulation algorithm allows us also to use a simple, not computationally intensive CAC algorithm without seeking an efficient but potentially complex algorithm. Indeed, the dynamical adaptation to the current situation makes more complex a-priori optimization needless.

Lastly, if the service provider runs a network on a bandwidth on demand infrastructure, he can periodically renegotiate his bandwidth with the network operator from whom he leases his network. Therefore C becomes also a varying parameter, but which can be adapted much less frequently than the degree of overbooking ε. On the other hand, in addition to the loss rate in virtual or actual systems, other variables can be controlled, like the rejected connection rate. This topic is presently under investigation.

ACKNOWLEDGMENTS

We would like to thank Prof. Paul J. Kühn (IND, University of Stuttgart, Germany) for his support. Special thanks to Monique Calisti, Christian Frei, Jean-Philippe Martin-Flatin and Steven Willmott for the good teamwork. This work was supported by the Priority Programm in Information and Communication Structures of the Swiss National Science Foundation, under grant SPP ICS 5003-45311.

REFERENCES

[1] J.-Y. Le Boudec and P. Thiran, *Network Calculus viewed as a Min-plus System Theory applied to Communication Networks*, Ecole Polytechnique Fédérale de Lausanne, Technical Report SSC/1998/016.

[2] A. Elwalid, D. Mitra and R. Wenworth, *A new Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node*, IEEE JSAC, vol. 13(6), pages 1048-1056, August 1995.

[3] K. J. Åström, B. Wittenmark, *Computer Controlled Systems*, Prentice-Hall, 1990, ISBN: 0-13-172784-2.

[4] Commission of the European Communities, *COST 224 Performance Evaluation and Design of Multiservice Networks*, Office for Official Publications of the European Communities, 1992, ISBN: 92-826-3728-X.