Non-Parametric and Self-Tuning Measurement-based Admission Control

Thomas Bohnert[†], Edmundo Monteiro[†], Yevgeni Koucheryavy[‡], Dmitri Moltchanov[‡]

[†]University of Coimbra
Polo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal
{tbohnert,edmundo}@dei.uc.pt
[‡]Tampere University of Technology
P.O.Box 553, FI-33101, Tampere, Finland
{yk,moltchan}@cs.tut.fi

Abstract. The Measurement-based Admission Control algorithm presented in this paper has been devised to overcome three, widely known shortcomings inherent to common design. Firstly, its configuration parameter strictly corresponds to standard QoS definitions in Service Level Agreements, namely packet loss probability. While the latter is featured by a set of alternative designs too, the second issue, that of considerable performance fluctuations under varying traffic conditions, is a rather general problem. Applying a purely non-parametric approach the presented algorithm's estimation model is free from assumptions, e.g. the Central Limit Theorem, and simulations assert our target, consistent performance for a set of various conditions. Finally, the third improvement is certainly the most appealing feature, the algorithm's independence of human fine-tuning. In fact, the optimal value of the performance determining parameter is being estimated from the actual, measured sample. In conclusion, the algorithm is highly adaptive as well as autonomous and simulations confirm near optimal performance and accuracy in terms of QoS commitments.

1 Introduction

It is common practice to oversubscribe access links at network edges in order to maximise the number of concurrent customers in the network, and therefore the revenue of Internet (ISP) and Content Service Providers (CSP). However, as resources are inherently limited, Quality of Service (QoS) can only be granted if explicit Admission Control (AC) is deployed to regulate resource contention in busy hours.

By definition a preventive congestion avoidance mechanism, the MBAC rationale is to limit the number of concurrent resource consumers in a network in advance. This is achieved by explicit access regulation for each and any new service request, i.e. a new flow, based on the expected availability of resources. Generally speaking, a new flow is granted admission to a network segment if its characteristics in terms of resource demand superimposed with that of previously admitted flows would be up to an extent, such that committed QoS grants are being sustained.

A coarse taxonomy divides existing AC approaches into two categories, Parameter-based Admission Control (PAC) and Measurement-based Admission Control (MBAC). The former, PAC, is solely based on *a priori* knowledge meaning that the resources required to cater a traffic aggregate, frequently termed Equivalent Bandwidth (EB), are derived from known traffic specifications of individual sources, for instance by QSPEC [1] in an NSIS architecture [2]. The simplest PAC is the so-called *Simple Sum* algorithm, which sums up individual peak rates of admitted flows plus the service requestor's.

Among others, there are three major shortcomings inherent to PAC. First, there are never enough matching source models for an ever-growing multi-service network like the Internet. Second, source

models are distorted due to cascades of networks, and hence, one has to model the whole network based on the (unknown) number of hops. Finally, PAC generally does not account for multiplexing gain leading to a considerable waste of resources.

To overcome these issues, Measurement-based Admission Control has been introduced. Principle idea is to sample the work arriving process in real time and to compute statistics according to purpose build queuing models. Eventually, the model is evaluated and future bounds of QoS objectives are predicted. The strength of MBAC therefore seems to lie in its independence on a priori knowledge and its capability to cope with changing conditions. An immediate and intuitive conclusion is that MBAC emerges naturally as the only solution for a dynamic environment like the Internet and indeed, MBAC has been shown to be effective in some sense and applicable in an extensive set of previous works, see for instance [3–7].

At this point, considering the plethora of MBAC algorithms devised in the past, a good justification for yet another one is necessary. In order to do so, we point the reader to an interesting fact. While further research in the area of MBAC has been generally questioned in [8], the same authors list in a follow-up [9], a widely cited paper, a set of yet open issues. In summary, by a comprehensively conducted comparison based on simulations the authors came up with the following conclusions.

- Algorithms performance varies considerably for different conditions, i.e. traffic characteristics.
- Performance strongly depends on algorithms individual, model specific performance parameter fine-tuning.
- Performance parameters are rooted in mathematics and lack intuitive meaning in a QoS or networking context.
- Performance parameter mapping to target QoS objectives is inconsistent.
- In most cases, algorithms under scrutiny missed targeted QoS objectives.

In fact, these findings have been further confirmed by an equally thorough evaluation in [10], with the only but significant difference that in this study the set of MBAC algorithms has been implemented in a real system. Eventually, for the sake of completeness, we refer to [11], where the same findings have been discussed based on some level of analytical evidence.

To evaluate the significance of these conclusions, let us review some MBAC context. An acceptable customer-blocking rate is around three percent in the busiest time. Given this, well-known daily service usage patterns inevitably impose that AC only plays an active role in one or maybe two hours during a day. In this short time frame a CSP (or ISP) wants a maximum of customers admitted onto the network as its chance for return of invest is highest. Real-time Traffic (RT), inherently rate limited, does not cause congestion in any other time. Provider's objective, however, is contrary to the customer's as those ideally want exclusive access to resources, i.e. a virtually circuit-switched service. In conclusion, optimal AC admits a maximum number of flows while tightly approaching committed QoS objectives.

Eventually, there is the operator's point of view. Service Level Agreements (SLA) does specify parameters like Bandwidth, Jitter, Delay or Loss [12] [13] and thus, an algorithm's configuration parameter ought to be of the same type. Moreover, setting of QoS targets must be reliable to ensure customers satisfaction, since customers have proven to be of little lenity to poor service quality and undoubted their satisfaction is the ultimate measure.

In the following discourse we present an MBAC algorithm designed to incorporate the identified requirements. In Sec. 2 we present a general queuing model, review related publications and derive a non-parametric estimation algorithm, which is independent from fine-tuning. Thereafter, we present the evaluation methodology. In order to do so, we present the evaluation setup in Sec. 3 and introduce a performance and accuracy metric in Sec. 4. Performance results for a set of different configurations are presented in Sec. 5 followed by an accuracy evaluation, in both cases jointly with a discussion of the results. Main findings together with the resulting conclusions are discussed in Sec. 7.

2 A General Model for Measurement-based Admission Control

Speaking in mathematical terms, the principle MBAC issue can be expressed as a real-time estimation problem. A sample $S = \{\zeta_1, \ldots, \zeta_n\}$ of Random Variables (RV) is captured by continuous measurement of the work arrival process and model specific statistics are calculated. Feeding these statistics in a queuing model allows predicting resource availability. Thus, the very first step is to define a queuing model.

As identified in the preceding section, an algorithm's configuration parameter shall have a meaning in a QoS context. In order to incorporate this, we use a simple buffer overflow model with the standard QoS criterion P_{loss} as configuration parameter.

Assume $\{\zeta_t\}$ denoting the work arrival process, and let $A[s,t]$ be the amount of work arriving in the interval $(s,t]$. Further let $\zeta_t = A[-t,0]$ such that the queue length at time zero is

$$\Omega = \sup_{t \geq 0}(\zeta_t - Ct). \tag{1}$$

with C denoting link capacity. The probability that the queue length exceeds ω is herewith

$$P\{\Omega > \omega\} = P\{\sup_{t \geq 0}(\zeta_t - Ct) > \omega\}. \tag{2}$$

As (2) is difficult to compute, we use the lower bound approximation

$$P\{\sup_{t \geq 0}(\zeta_t - Ct) > \omega\} \geq \sup_{t \geq 0} P\{\zeta_t > \omega + Ct\}. \tag{3}$$

With $\rho = \omega + Ct$, and let $F_t(x) = P\{\zeta_t \leq x\}$ be the Cumulative Distribution Function[1] (CDF) of $\{\zeta_t\}$ we get

$$P\{\Omega > \omega\} \geq \sup_{t \geq 0} P\{\zeta_t > \omega + Ct\} = \sup_{t \geq 0}(1 - F_t(\rho)). \tag{4}$$

Now lets set ω in $\rho = \omega + Ct$ to be the buffer size and lets further denote r_p as the peak rate of a flow. The flow's worst-case resource demand would be if it emits continuously with r_p and hence ρ is set to $\rho = \omega + Ct - r_p t$. This yields the final admission criteria based on the buffer overflow probability $\hat{P}_{loss}(\rho)$ which reads as

$$Admit_{bool} = \begin{cases} true \quad if \quad \hat{P}_{loss}(\rho) < P_{loss} \\ false\, if \quad \hat{P}_{loss}(\rho) \geq P_{loss} \end{cases}. \tag{5}$$

As the r.h.s of (4) reveals, this simple model's accuracy depends on the marginal distribution function. This is fundamental to probability modelling and the generally adopted approach obeys that of *Parametric Statistics*. Parametric means, that the nature of the CDF is assumed to be known. Motivated by the Central Limit Theorem (CLT) for instance, many MBAC models assume Gaussianity (or Normality), like [3] [5] [6] [7] and [14], while for instance [6] and [4] assume a Gumbel distribution motivated by Extreme Value Theory. Once the CDF is known, the problem is to estimate the moments from the captured sample S to get $\hat{P}_{loss}(\rho)$.

With rising aggregation levels in access networks and due to its conformance with Self-Similarity (SS) and Long-Range-Dependency (LRD) findings in network traffic [15] [16] [17] [18], much attention has been paid to Gaussian approximation recently. An analytical argument in favour of is documented in [19], while [20] further supports it empirically based on real traffic traces, though for backbone traffic and for large time scales. The latter results, however, are questionable as common Goodness-of-fit techniques were applied though known to be error prone for large samples and in the context of SS and LRD [21, Chap. 10] [22] [23] [24, Page 33]. In [22] a more profound mathematical approach has been applied to test the Gaussian approximation, but the result is rather inconclusive, stating that under certain conditions the approximation can be met but also grossly missed, particularly for small time scales.

[1] Strictly speaking the Marginal Distribution Function

In contrast to Gaussian approximation, the authors of [25] and [26] state traffic obeys a Poisson law and is non-stationary. However, these results have been recently called to be far unrealistic [23]. Finally, and only for the sake of further highlighting diversity of findings, we cite [27] which recently recommended the Gamma distribution as the *"best choice on average"* for a comprehensive set of recorded traffic traces.

This short discourse indicates the uncertainty incurred by assuming in advance. The inflicted consequence is two fold. First, assuming for example a Gaussian nature implies a heavy-traffic approximation. Only then the CLT becomes valid. Furthermore, densities tend to be heavy-tailed, and in this case, moments beyond first order may not exist at all.

There is ample evidence that exaclty these assumptions subject MBAC performance so strongly to traffic conditions. Based on this finding, the focus of this work was to seek and investigate an alternative.

To free $F_f(\rho)$ in (4) from assumptions, *Non-Parametric Statistics* are to be applied. The simplest way would be to compute a histogram as estimate for $F_f(\rho)$. Indeed, several histogram based MBAC algorithms, e.g. [28] [29] [30], have been published in the past. However, in none of these works the choice for histograms has been justified by statistical uncertainty, but has been used rather implicitly. This is evidenced by the fact that histograms are solely used in an ad-hoc fashion and parameters, namely bin width and bin origin, are chosen intuitively. That this salvages the risk of great imprecision is widely unknown but elaborately presented in [24, Chap. 3] for instance. Furthermore, histograms are poor for heavy-tailed distributions as they only represent the statistical information of the actual sample. As sample size is inherently limited, rare (tail) events are only in a few of sequentially captured samples.

A non-parametric method superior to histograms to estimate an unknown density f is Kernel Density Estimation (KDE). Following [24, Chap. 6, Equ. 6.1] its definition reads

$$\hat{f}_h(y) = \frac{1}{nh} \sum_{i=1}^{n} k\Big(\frac{y - Y_i}{h}\Big) \tag{6}$$

where Y is sample of n RVs, Y=$\{Y_1, \ldots, Y_n\}$, h is the window width, also called the *smoothing* or *bandwidth* and k is the *kernel*, which has to satisfy the condition

$$\int_{-\infty}^{+\infty} k(u)du = 1 \tag{7}$$

and is therefore often chosen from a density function. In brief, the estimator can be considered as a sum of superimposed bumps centred at the observations Y_i. The shape of the bumps is defined by the kernel function while the window width h defines their width [31].

Integrating (6) yields the distribution estimator

$$\hat{F}_h(y) = \frac{1}{nh} \int_{-\infty}^{y} \hat{f}_h(u)du = \frac{1}{n} \sum_{i=1}^{n} K\Big(\frac{y - Y_i}{h}\Big) \tag{8}$$

where

$$K(x) = \int_{-\infty}^{x} k(u)du \tag{9}$$

is the integrated kernel function, short the kernel.

The metric for accuracy evaluation for this type of estimator is the *Mean Integrated Squared Error* (MISE) defined as

$$MISE(h) = E \int_{-\infty}^{+\infty} \{\hat{F}_h(y) - F(y)\}^2 dy. \tag{10}$$

Letting $hn^{1/2} \to \infty$ as $n \to \infty$ yields the *Asymptotic Mean Integrated Squared Error* (AMISE). It has been shown in [32] that the AMISE is minimised by setting h equal to

$$h^o = (\Upsilon(K)/R_1)^{1/3} n^{-1/3} \tag{11}$$

with $\Upsilon(K)$ being a function of the kernel only. For instance, using a standard normal density as kernel, $k(x) = \phi(k)$ and $K(x) = \Phi(x)$, yields $\Upsilon = 1/\sqrt{\pi}$.

Provided an estimate of R_1, the so-called *roughness* defined as

$$R_m = \int_{-\infty}^{+\infty} (f^{(m)}(y))^2 dy \tag{12}$$

where f^m denotes the mth derivative of the true, unknown density f, the convergence rate of this estimator, determined by (11), is in the order of $n^{-1/3}$. This is slower as the rate of n^{-1} for parametric estimators under optimal conditions, i.e. a full match between reality and model. This condition is, however, practically never given.

One can see that the performance of this estimator depends on the smoothing parameter h^o, which is a function of R_1, and the kernel. In [24, Sec. 6.2.3.2] it has been shown that optimal kernels have compact support but as they are "best in average", recall the AMISE is a mean, these kernels are optimal for the body of the density. To estimate tail probabilities, a kernel with infinite support is the better choice. However, the kernel choice is of minor significance in any case [24, Sec. 6.2.3].

Thus, the performance knob for this type of estimator is h^o, and thusly for any therefrom derived MBAC algorithm. Moreover, this performance parameter masks the configuration parameter as its setting rules how precisely the QoS target is being approached. Unfortunately, this parameter lacks any meaning in an QoS context and we are facing one of the critics of MBAC algorithms as presented in Sec. 1.

Having a closer look, however, we see that the smoothing parameter $h^o = f(R_1)$ and (12) poses the same problem as (8) with the only difference that we have to differentiate (6) and not to integrate it. Thus, and we can apply KDE to estimate the derivative R_1 from the sample the same way we do to estimate $\hat{F}(x)$. This method, is known as *plug-in smoothing*.

This yields an particular advantage as the MBAC algorithm becomes herewith *"self-tuning"* because the estimator does compute its only performance parameter from the captured traffic sample. No human intervention, i.e. fine tuning, is necessary.

By repeated integration by parts, (12) becomes

$$R_m = (-1)^m \int_{-\infty}^{+\infty} f^{2m}(y) f(y) dy = (-1)^m E[f^{(2m)}(Y)]. \tag{13}$$

Using a Gaussian kernel with smoothing a, one gets an estimate of $\hat{f}^{(2m)}(y)$

$$\hat{f}^{(2m)}(y) = \frac{1}{n} \sum_{i=1}^{n} \phi^{(2m)} \left(\frac{y - Y_i}{a} \right). \tag{14}$$

Based on the last two equations, the Jones and Sheather estimator reads [33]

$$\hat{R}_m(a) = (-1)^m \frac{1}{na} \sum_{i,j=1}^{n} \phi^{(2m)} \left(\frac{Y_j - Y_i}{a} \right). \tag{15}$$

Its AMISE optimal smoothing is

$$a_m(R_{m+1}) = \left(\frac{2^{(m+1/2)} \Gamma(m + 1/2)}{\pi R_{m+1} n} \right)^{1/(2m+3)} \tag{16}$$

and is an explicit function of R_{m+1}. Thus,

$$\hat{R}_m(R_{m+1}) = \hat{R}_m(a_m(R_{m+1})). \tag{17}$$

This sequential relationship motivates the *sequential plug-in rule* published in [34]. Briefly, "fix $J \geq 1$ and take R_{J+1} as given", hence

$$\hat{R}_J(R_{J+1}), \hat{R}_{J-1}(\hat{R}_J), \hat{R}_{J-2}(\hat{R}_{J-1}), \ldots \tag{18}$$

until one obtains $\hat{R}_1(\hat{R}_2)$ for (11). The recommended value for J is 4 [34].

A so-called reference estimate for R_{J+1} in (18) has been published in [35]. With the samples standard deviation σ, it becomes

$$R_{J+1} = \frac{\Gamma(J+3/2)}{\sigma^{2J+3}2\pi}. \tag{19}$$

and we finally can write R_1 as

$$\hat{R}_{1,J} = f(R_{J+1}, J) \tag{20}$$

Given (20), one simple has to *"plug-in"* this estimate in (11) to get

$$h^o = (\Upsilon(K)/R_{1,J}^{1/3} n^{-1/3}. \tag{21}$$

Based on this framework, we can define a non-parametric, self-tuning MBAC algorithm. In order to do so, we insert (20) in (8) and the latter in (4). Thus, $\hat{P}_{loss}(\rho)$ in (5) finally reads

$$\hat{P}_{loss} = P\{\Omega > \rho\} = \sup_{t \geq 0} \Big\{ 1 - \frac{1}{nh^o} \sum_{i=1}^{n} K\Big(\frac{\rho - \zeta_t^i}{h^o}\Big) \Big\}. \tag{22}$$

In conclusion, the features of (22) account for three major issues discussed in Sec. 1. First, its configuration parameter P_{loss} is a common SLA QoS parameter. Further, its performance knob, optimal smoothing h^o, is independent from human fine-tuning but the algorithm computes it autonomously. Eventually, the algorithm's performance does not depend on any traffic condition, like for example heavy-traffic approximation (CLT) or an exponential decay of the loss probability, usually assumed in EB models [36].

3 Evaluation Setup

Common practice in this field of research is to implement and evaluate using the NS-2 Network Simulator. In order to gain optimal insight we defined a simple "n sources, one router, one destination" topology where MBAC is deployed at the bottleneck link between the router and the destination. As traffic model, we opt for Voice Over IP (VoIP) traffic as it is the first QoS requiring service with a penetration on a global scale.

As one motivation for the model was independence from assumptions, the questions we ask is how well the algorithm performs if subjected to different load conditions. Thus, we varied the bottleneck link capacity C in the range of 1 to 5 Mbps, an range where the CLT assumption far from reasonable. For each of this configurations, we evaluated the performance for a set of QoS targets P_{loss} in (5), namely $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$.

For service request inter-arrival distribution we chose an Exponential distribution and so we did for holding times. The latter has a fixed mean λ_h of 300s, while the former, λ_s, is adjusted such that for each capacity C, the link is continuously overloaded and the MBAC works on its limit.

By choosing two different On/Off models with either Exponential (EXPOO) or Pareto (POO) distributed On/Off times, another dimension of diversity is added. While the former represents a short-range dependent model, the latter produces work arriving processes with SS/LRD characteristics [37]. In the style of [38], On and Off times are respectively 0.3s and 0.6s and in On state sources emit 125-Bytes-packets with rate 64kbps. Thus, sources can be considered as G.711 encoded with Voice Activity Detection.

Routers buffer sizes vary with the link capacity too. For high quality VoIP communications, the maximum mouth-to-ear delay has to be less than 150ms. Assuming a 10 hop path as common, and further delay equally distributed over all hops, the buffer size in (4) is set to $\omega = C * 15ms$. Thus, the likelihood $P_{delay}(D \geq 15ms)$ is the same as a loosing a packet.

The time scales of interest in (4) are set to $t = \{0.02s, 0.04s, 0.06s, 0.08s, 0.1s\}$ and the total simulation time was set to 4100s. By visual inspection, we found that for any configuration the time to reach steady state was well below 500s. Given this, the first 500s have been discarded for performance evaluation, resulting in an effective time of $t_{sim} = 3600s$, i.e. a busy hour.

4 Performance Metrics

In [9], the authors stated that all algorithms under test perform equally given *homogeneous* sources. This is a sizeable statement. The conclusion has been drawn after evaluating the algorithms using the MBAC standard Load-Loss metric. The approach is to overload the network with flow arrivals and to measure

$$Loss = \frac{\#pkts_{lost}}{\#pkts_{sent}}, \;\; Load = \frac{\#pkts_{fw}}{Ct_{sim}}. \tag{23}$$

where $pkts_{fw}$ denotes the packets forwarded on the bottleneck.

But what does this metric tell us? Actually only little about MBAC performance. Indeed, it simply reveals that for a given load in the network, the queue overload level is X. The point here is that to achieve a target network load, an algorithm's particular tuning parameter, typically some λ, δ depending on the model, is adjusted until it admits the required numbers of flows to get the desired load. This number of flows is in the case of homogeneous sources almost the same for each and any MBAC algorithm and hence, the loss is also the same. There is almost no difference except in the sequence of admissions. Thus, this metric doesn't really tell us much about the admission behaviour of a particular algorithm.

What further confirms this finding is that in case of heterogeneous sources the performance did actually vary, see Fig. 8 in [9]. Clearly, for this scenario each algorithm admits a particular sequence, and thus set of sources to get the desired load resulting in individual aggregates with different characteristics. Consequently loss for each configuration is also different. Hence, this metric is merely useful for heterogeneous sources.

In [39], it has been shown that the worst-case traffic pattern is that of homogeneous On/Off sources; as used for this evaluation. This requires a modification of the performance metric by replacing Loss in (23) with P_{loss} in (5). As a result we get a utilisation measure, implicitly expressing the number of flows admitted onto the network, for a given QoS target. For example, a more aggressive algorithm results in higher resource usage, a more restrictive one leaves more spare resources. This metric therefore expresses a criteria for an ISP/CSP as discussed in Sec. 1.

The former metric is used to provide a criteria from a providers perspective. Given an QoS target, how much use does the algorithm make of available resources. The second metric we introduce reflects customers as well as operators point of view. More precisely, we are interested in how tightly the algorithm approaches required QoS targets. For example, if a providers commits in an SLA a P_{loss} of 10^{-5} and the operator configures the algorithm respectively, how accurate does the algorithm approach this limit.

5 Load-Loss Performance

Figure 1 depicts the results for the 5Mbps configuration and both source models. One can see that the algorithm does perform well, i.e. the network resources are used up to ~ 0.94 times the link capacity C ($0.94 * C$) for POO sources and a QoS target set to $P_{loss} = 10^{-2}$. For tighter QoS requirements, the algorithm rejects more flows and thus, the lowest link utilisation is for EXPOO sources $\sim 0.82 * C$ and an QoS target of 10^{-5}.

In Fig. 2 results for the other extreme are illustrated, a link capacity C as low as 1Mbps. Similar to the previous example, the algorithm makes use of available resources for relaxed QoS targets while the performance curve decays rapidly along with tighter requirements. However, compared to the 5Mbps case, the maximum link utilisation is much lower. The reason appears clear. A smaller capacity C, means less flows admitted and the resulting aggregate resembles much more that of a single source, i.e. a On/Off pattern. It is therefore more impulsive and as the algorithm does account for this, it does reject more flows.

The remaining results for configuration 2, 3 and 4Mbps exhibit similar patterns and as space is rare here, we omit their graphical presentation. For each configuration and the POO model exact numbers are presented in Tab. 1.

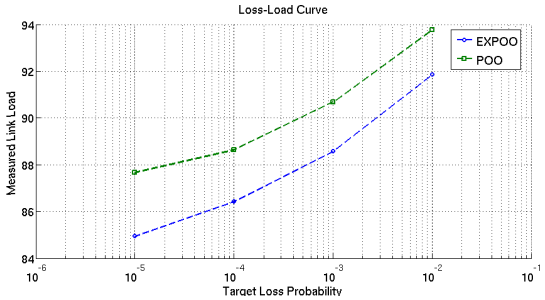

Fig. 1. Target QoS objective versus Load performance for a 5Mbps configuration. The X-Axis shows P_{loss} in (5), the Y-Axis Load as defined in (23) but in percentage.

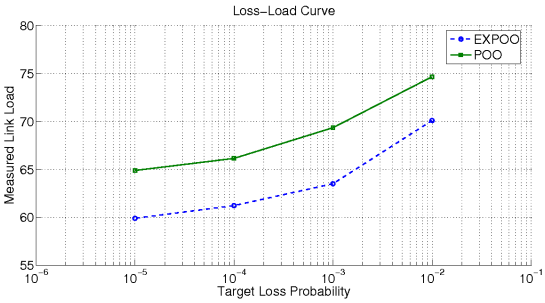

Fig. 2. Target QoS objective versus Load performance for a 1Mbps configuration. The X-Axis shows P_{loss} in (5), the Y-Axis Load as defined in (23) but in percentage.

Table 1. Loss - Load performance for each configuration and the POO source model

P_{loss}	1Mbps	2Mbps	3Mbps	4Mbps	5Mbps
0.010000	74.64	84.36	88.61	92.07	93.77
0.001000	69.33	79.50	84.83	88.30	90.68
0.000100	66.13	77.85	82.48	86.77	88.64
0.000010	64.87	75.65	81.52	85.39	87.67

6 Control Accuracy Evaluation

In the style of the previous section, Fig. 3 depicts the precision of the algorithm for a 5Mbps configuration. The measured loss as defined in (23) is plotted as a function of the configured QoS objective. The plotted performance curve is roughly divided in two parts and its midpoint is located at a target loss probability $P_{loss} = 10^{-3}$. Approaching this point from the right, QoS objectives are closely met, i.e. $Loss \sim P_{loss}$ for $P_{loss} = 10^{-2}$ and $P_{loss} = 10^{-3}$.

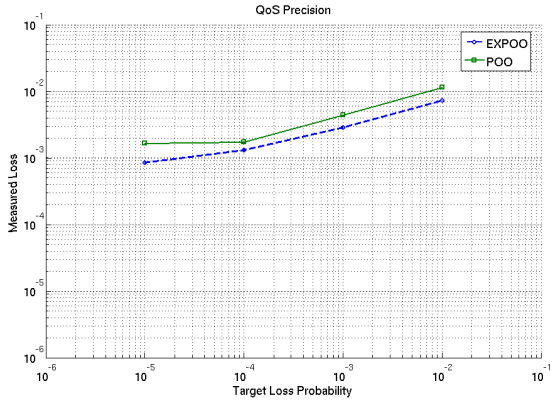

Fig. 3. Target QoS objective versus Loss performance for a 5Mbps configuration. The X-Axis shows P_{loss} in (5), the Y-Axis Loss as defined in (23).

The opposite is the case for $P_{loss} = 10^{-4}$ and $P_{loss} = 10^{-5}$. For this configuration the algorithm is slightly too daring and the QoS objective deviates from its optimum. A similar behaviour is also depicted in Fig. 4, which illustrates the results for the 1Mbps configuration. The optimal working

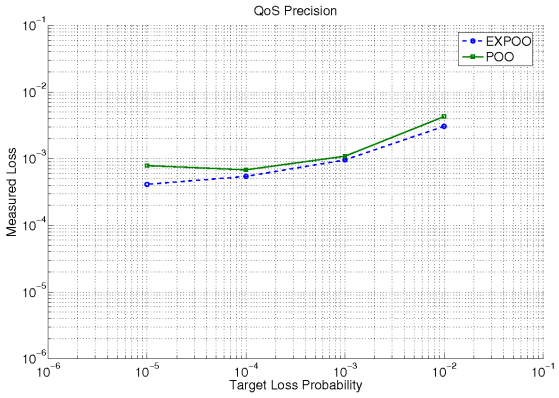

Fig. 4. Target QoS objective versus Loss performance for a 1Mbps configuration. The X-Axis shows P_{loss} in (5), the Y-Axis Loss as defined in (23).

point in this case lies exactly at $P_{loss} = 10^{-3}$. For the less stringent requirement $P_{loss} = 10^{-2}$ it does guarantee the QoS objective and approaches the optimal point tightly for both traffic models.

The same cannot be claimed for more tighter QoS objectives, namely $P_{loss} = 10^{-4} or 10^{-5}$. As in the 5Mbps configuration, the algorithm is too daring, admitting too many flows into the network with the result of greater packet loss as demanded.

There are basically two reasons, which contribute, to this imprecision. As already mentioned before, for such a small link capacity the aggregate is very impulsive and the true marginal distribution therefore rather discrete than continuous, implying consequences on the optimality of the estimator.

Besides the former, the dominant factor lies in the queuing model itself. Observant readers might have noticed that (22) poses an overflow model rather than a loss model. Consequently, the algorithm does not differentiate between a one packet and 100-packet loss but does consider both cases as an equal overflow. Further, the definition of ρ assumes that the buffer is always empty at any observation point.

There is one feature important to be noticed. In both cases the configuration behaviour is practically equal for both traffic models, i.e. the curves are tightly superimposed. This implies that there is a chance for a consistent mapping of the configuration and the experienced QoS level independent from actual traffic characteristics. In other words, it allows to acquire reliable knowledge about the deviation for a certain setting. Furthermore, though slightly imprecise, the deviations are only for very stringent cases. For these settings, imprecision are relatively significant but absolutely at least questionable.

More detailed insight, namely precise numbers for the POO source model are tabulated in Tab. 2, as before for all configurations.

Table 2. Accuracy evaluation for each configuration and the POO source model

P_{loss}	1Mbps	2Mbps	3Mbps	4Mbps	5Mbps
0.010000	0.004293	0.005778	0.005057	0.007468	0.011436
0.001000	0.001092	0.001582	0.001771	0.002409	0.004424
0.000100	0.000683	0.001056	0.000644	0.001801	0.001731
0.000010	0.000789	0.000341	0.000553	0.000938	0.001665

7 Conclusion

The design of MBAC algorithm presented in this work has been devised to target three major critics of MBAC. First, configuration parameter are without meaning in a SLA or QoS context. Second, common MBAC design is based on parametric models and incurs dependencies on assumptions and therefore limits applicability. Hence, MBAC performance depends strongly on a match between model assumptions and traffic characteristic causing reasonable performance for on type of traffic and poor for another. Finally individual performance depends on human fine-tuning of model specific tuning knobs.

Defining a simple queuing model, the algorithms configuration parameter is P_{loss}. This solves the first point, configuration in conformance with common SLA QoS parameters.

By using a purely non-parametric model, i.e. free from any assumption, the algorithm is free of any statistical dependency what distinguishes this approach from common design. The results are promising. The algorithm exhibits consistent performance if traffic characteristics are varied in either dimensions, namely link capacity (aggregation) and traffic nature, i.e. source models.

Furthermore, the proposed model tackles the third major critic of MBAC algorithms, the requirement of human performance parameter fine-tuning. Indeed, the algorithm can be considered as

self-tuning as it estimates the optimal value for its only performance parameter, namely h^o from the sample. No human intervention is needed.

Simulations confirmed that the algorithm does perform efficiently, i.e. makes near optimal use of resources, is reliable, i.e. it does approach granted QoS commitments tightly, and most important, its behaviour is consistent, i.e. its performance and configuration is near independent from traffic conditions.

The alert reader, however, did recognise the computational complexity of the presented framework. Indeed, Equ. 15 poses a non-trivial computational burden and limits practical deployment. Therefore, ongoing research is focused on reducing complexity, e.g by replacing (15) with a finite mixture model where the impact of trading-off accuracy, fitting finite mixture models depend on Expectation-Maximisation, with lower computational burden seems particularly intriguing.

Acknowledgement

The authors sincerely acknowledge the invaluable discussions with Prof. Paulo Oliveira of the Department of Mathematics and Prof. Helder Arajo of the Department of Electrical Engineering, both associated with the University of Coimbra.

This work has been jointly supported by the EU IST Integrated Project WEIRD (WiMAX Extensions for Remote and Isolated Data Networks) and ESF COST 290 Action.

References

1. J. Ash, A. Bader, and C. Kappler. *QoS NSLP QSPEC Template*. IETF Internet Draft. IETF NSIS Working Group, oct 2006. Accessed on 26.10.2006.
2. R. Hancock, G. Karagiannis, J. Loughney, and S. Van den Bosch. *Next Steps in Signaling: Framework*. RFC4080. IETF, jun 2005.
3. S. Georgoulas, P. Trimintzios, and G. Pavlou. Joint Measurement-and Traffic Descriptor-based Admission Control at Real-Time Traffic Aggregation. In *ICC '04: IEEE International Conference on Communications*, 2004.
4. E. Knightly and J. Qiu. Measurement-based admission control with aggregate traffic envelopes. In *IEEE ITWDC '98*, Ischia, Italy, sep 1998.
5. M. Grossglauser and D. N. C. Tse. A framework for robust measurement-based admission control. *IEEE/ACM Transactions on Networking*, 7(3):293–309, 1999.
6. J. Qiu and E. W. Knightly. Measurement-based admission control with aggregate traffic envelopes. *IEEE/ACM Transactions on Networking*, 9(2):199 – 210, 2001.
7. T. K. Lee and M. Zukerman. Practical approaches for connection admission control in multi-service networks. In *Proceedings of IEEE ICON '99*, pages 172–177, Brisbane, Australia, May 1999.
8. L. Breslau, S. Jamin, and S. Shenker. Measurement-based admission control: what is the research agenda. In *IWQOS*, pages 3–5, London, UK, mar 1999.
9. L. Breslau, S. Jamin, and S. Shenker. Comments on the Performance of Measurement-Based Admission Control Algorithms. In *INFOCOM*, pages 1233–1242, Tel Aviv, Israel, mar 2000.
10. A. W. Moore. An implementation-based comparison of Measurement-Based Admission Control algorithms. *J. High Speed Networks*, 13(2):87 – 102, 2004.
11. Y. Jiang, P. Amstad, F. Victor, and A. Nevin. Measurement-Based Admission Control: A Revisit. In *17th Nordic Teletraffic Seminar, NTS-17*, Fornebu, Norway, August 2004.
12. J. Evans and C. Filsfils. Deploying DiffServ at the network edge for tight SLAs, part 1. *IEEE Internet Computing*, pages 61–65, January 2004.
13. J. Evans and C. Filsfils. Deploying DiffServ at the network edge for tight SLAs, part 2. *IEEE Internet Computing*, pages 61–69, mar 2004.
14. S. Floyd. Comments on measurement-based admissions control for controlled-load services. Technical report, August 1996.

15. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1 – 15, 1994.
16. V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. In *SIGCOMM*, pages 257–268, 1994.
17. M. E. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. In *SIGMETRICS '96*, pages 160–169. ACM Press, August 1996.
18. G. Mao. Finite Timescale Range of Interest for Self-Similar Traffic Measurements, Modelling and Performance Analysis. In *IEEE International Conference on Networks, ICON'03*, pages 7 – 12, Sydney, 2003.
19. R. G. Addie. On the applicability and utility of Gaussian models for broadband Traffic. In *Proceedings of the International Conference on ATM*, Colmar, France, 1998.
20. C. Fraleigh, F. Tobagi, and C. Diot. Provisioning IP backbone networks to support latency sensitive traffic. In *INFOCOM 2003*, volume 1, pages 375–385. IEEE, apr 2003.
21. J. Beran. *Statistics for Long-Memory Processes*. Chapman & Hall / CRC, 1 edition, 1994.
22. J. Kilpi and I. Norros. Testing the Gaussian approximation of aggregate traffic. In *Internet Measurement Workshop*, pages 49 – 61. ACM, 2002.
23. W. Willinger, D. Alderson, and L. Li. A pragmatic approach to dealing with high-variability in network measurements. In Alfio Lombardo and James F. Kurose, editors, *Internet Measurment Conference*, pages 88–100. ACM, 2004.
24. *Multivariate Density Estimation*. Probability and Mathematical Statistics. John Wiley & Sons, 1 edition, 1992.
25. T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A Nonstationary Poisson View of Internet Traffic. In *Infocom*, 2004.
26. J. Cao, W. S. Cleveland, D. Lin, and D. X. Sun. On the nonstationarity of Internet traffic. In *ACM SIGMETRICS '01*, pages 102 – 112, New York, NY, USA, July 2001. ACM Press.
27. A. Scherrer, N. Larrieu, P. Owezarski, and P. Abry. Non Gaussian and Long Memory Statistical Characterisation for Internet Traffic with Anomalies. Technical report, Ecole Normale Superieure de Lyon, 2005.
28. M. Zukerman and P. W. Tse. An Adaptive Connection Admission Control Scheme for ATM Networks. In *Icc (3)*, pages 1153 – 1157, 1997.
29. K. Gopalan, T. C. Chiueh, and Y.-J. Lin. Probabilistic delay guarantees using delay distribution measurement. In *ACM Multimedia*, pages 900 – 907, 2004.
30. M. Menth, J. Milbrandt, and S. Oechsner. Experience Based Admission Control. In *The Ninth IEEE Symposium on Computers and Communications ISCC2004*, Alexandria, Egypt, 2004.
31. B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1 edition, 1986.
32. M. C. Jones. The performance of kernel density functions in kernel distribution function estimation. *Statistics & Probability Letters*, 9(2):129–132, February 1990.
33. M. C. Jones and S. J. Sheather. Using non-stochastic terms to advantage in kernel distribution function estimation. *Statistics and Probability Letters*, 11:511–541, oct 1991.
34. B. E. Hansen. Bandwidth Selection for Nonparametric Distribution Estimation. Technical report, University of Wisonsin, May 2004.
35. J. S. Marron and M. P. Wand. Exact mean integrated squared error. *Annals of Statistic*, 20:712–736, 1992.
36. E. W. Knightly and N. B. Shroff. Admission control for statistical QoS: theory and practice. *IEEE Network*, 13(2):20–29, mar 1999.
37. T. M. Bohnert and E. Monteiro. A comment on simulating LRD traffic with pareto ON/OFF sources. In *CoNEXT'05: ACM Conference on Emerging Network Experiment and Technology*, pages 228–229, Toulouse, France, October 2005. ACM Press.
38. A. P. Markopoulou, F. A. Tobagi, and M. J. Karam. Assessing the quality of voice communications over internet backbones. *IEEE Transactions on Networking*, 11(5):747–760, oct 2003.
39. G. Mao and D. Habibi. A Cell Loss Upper Bound for Heterogeneous On-Off Sources with Application to Connection Admission Control. *Computer Communications*, 25(13):1172–1184, August 2002.