KBAC: Knowledge-Based Admission Control

Doreid Ammar *, Thomas Begin *, Isabelle Guérin-Lassous * and Ludovic Noirie†
*Université Lyon 1 / LIP (UMR ENS Lyon - INRIA - CNRS - UCBL)
Email: {doreid.ammar, thomas.begin, isabelle.guerin-lassous}@ens-lyon.fr
†Alcatel-Lucent Bell Labs, Nozay, France
Email: ludovic.noirie@alcatel-lucent.com

***Abstract*—Many methods have been proposed in the literature to perform admission control in order to provide a sufficient level of Quality of Service (QoS) to accepted flows. In this paper, we introduce a novel data-driven method based on a time-varying model that we refer to as Knowledge-Based Admission Control solution (KBAC). Our KBAC solution consists of three main stages: (i) collect measurements on the on-going traffic over the communication link; (ii) maintain an up-to-date broad view of the link behavior, and feed it to a *Knowledge Plane*; (iii) model the observed link behavior by a mono-server queue whose parameters are set automatically and which predicts the expected QoS if a flow requesting admission were to be accepted. Our KBAC solution provides a probabilistic guarantee whose admission threshold is either expressed, as a bounded delay or as a bounded loss rate. We run extensive simulations to assess the behavior of our KBAC solution in the case of a delay threshold. The results show that our KBAC solution leads to a good trade-off between flow performance and resource utilization. This ability stems from the quick and automatic adjustment of its admission policy according to the actual variations on the traffic conditions.**

I. Introduction

Over the last few years, new usages such as streaming or live video watching are increasingly representing a significant part of Internet traffic. Network operators face the challenge of satisfying the quality of experience expected by end-users while, in the same time, avoiding the over-provisioning of transmission links. Bandwidth management offers a wide spectrum of policies to overcome this issue. Possible options include congestion control, scheduling algorithms, traffic shaping and admission control. In this paper, we focus on admission control.

Admission control is a mechanism used to prevent some flows from accessing a computer network with regard to the current utilization level of the network resource. By regulating the number of on-going flows, admission control aims at preventing overloading, congestion and performance collapses, so that, accepted flows receive a sufficient level of Quality of Service (QoS), which is of utmost importance for delay-sensitive applications (*e.g.*, Telephony over IP) and resource-intensive applications (*e.g.*, streaming video).

Admission control has been an active field of research for many years. Despite the number and the variety of proposed solutions, virtually all of them, if not all, are hampered by the difficulty to calibrate correctly their tuning parameters so as to maximize the resource utilization and the QoS expected by the end-users. This issue has been related in several former studies. For instance, [2], [5], [12], [13], [14] compare different measurement-based admission control (MBAC) solutions using various traffic conditions. These studies show that, for some specific scenarios, some solutions meet the QoS but often at the cost of a very small utilization level. However, for many other scenarios, in which the traffic condition differs, they violate the QoS target. One can therefore think that there is still a lack of effectiveness for existing admission control solutions. It is the authors point of view that a possible means to enhance MBAC solutions is to include a *Knowledge Plane* in their measurement algorithms.

This paper introduces a novel admission control solution based on a *Knowledge Plane*. Our Knowledge-Based Admission Control solution (KBAC) consists of three main stages: (i) collect measurements on the on-going traffic over the communication link; (ii) maintain an up-to-date broad view of the link behavior, and feed it to a *Knowledge Plane*; (iii) model the observed link behavior by a mono-server queue whose parameters are set automatically and which predicts the expected QoS if a flow requesting admission were to be accepted. Our KBAC solution provides a probabilistic guarantee whose admission threshold is either expressed, as a bounded delay or as a bounded loss rate. In this article, we present its application to the case of an admission threshold expressed as a maximum tolerable (bounded) delay.

Our new KBAC solution avoids the critical step of precisely calibrating key parameters. The experimental results show that our KBAC solution leads to a good trade-off between flow performance and resource utilization. This ability stems from the quick and automatic adjustment of its admission policy according to the actual variations on the traffic conditions.

The remainder of this paper is organized as follows. Section II discusses the state of the art on admission control solutions. In Section III, we describe our new Knowledge-Based Admission Control (KBAC). Section IV is devoted to our experimental framework. Section V presents several simulation results illustrating the performance of our proposed solution. Finally, Section VI concludes this paper.

II. Related Work

There are different approaches to perform admission control. First, endpoint admission control solutions make use

This work has been partly supported by the project *Semantic Networking* within the common laboratory INRIA - Alcatel Lucent-Bell Labs.

of probing packets that aim at reproducing the traffic pattern that the source is on the verge to transmit through the network [6]. This approach is referred to as an active technique since artificial traffic is injected into the network to perform admission control. Second, admission control solutions can be based on the use of traffic descriptors. The underlying idea primarily consists in theoretically assessing the current network workload using traffic descriptors. Then, the admission control uses the found value to decide, given the incoming flow traffic descriptor, whether or not to let it come into the network. Clearly, such an approach requires to know traffic descriptors for every on-going (accepted) flow as well as for any incoming flow [12]. Third, measurement-based admission control (MBAC) solutions rely exclusively on measurements to assess the workload of on-going traffic over each communication link. Unlike the first type of solutions, these solutions are categorized as passive techniques. MBAC solutions differ from the second type of solutions since they do not require any explicit knowledge on the traffic descriptors of on-going flows. Several MBAC solutions have been proposed in the literature. These solutions are generally thought to operate on a single communication link, and the admission control must be repeated for each link along the path of the flow. These solutions are basically made up of two parts. First, they perform measurements on the on-going traffic, and deliver measured metrics (*e.g.*, the residual capacity of the link). Second, they rely on an algorithm that includes a test operation, whose outcome decides whether or not to let a new flow requesting admission come into the network. Existing MBAC solutions mainly differ by their measurement operations and by the theoretical assumptions made on the on-going traffic.

The remainder of this section is restricted to highlight the measured metrics required by some of the most known MBAC solutions. *Guerin et al.* were the first to introduce in [9] the concept of *Equivalent Capacity* used in several admission control solutions. The Equivalent Capacity of aggregated traffic over a communication link, $C(\epsilon)$, is defined as being such that the probability for the arrival data rate of aggregated traffic to exceed $C(\epsilon)$ is at most ϵ. Basically, any MBAC solution based on Equivalent Capacity attempts to ensure that, for any link on the path between the source and the destination, the rate of the flow requesting admission summed to the actual Equivalent Capacity keeps below the nominal link capacity. The formula for the Equivalent Capacity given in [9] assumes a buffer-less model and an aggregate arrival rate that follows a Normal distribution. Floyd proposed in [7] an alternative formula for the measurement of Equivalent Capacity based on Hoeffding bounds. This formula uses an upper bound of the peak rate for each admitted connection along with the measured aggregate arrival rate. In [8], *Georgoulas et al.* present an admission control solution based on the Equivalent Capacity given in [9]. This solution uses measurements of the aggregate bandwidth only, without keeping the state of any per-flow information. In addition, *Georgoulas et al.* include an Admission Policy Factor in their admission control algorithm that allows the operator to tune its degree of conservativeness in terms of packet loss rate. These three latter solutions require measurements only on the utilization rate of each communication link to be run. In [11], *Jamin et al.* were the first to integrate in their admission control the queueing delay constraint. To be performed, this solution requires, in addition to a measurement on the actual utilization rate of the link, a measurement on the waiting time being spent in the queue (buffer). *Qiu and Knightly* improve in [15] the works of *Jamin et al.* by proposing an alternative measurement of the utilization rate of the link in order to have a better traffic characterization over this link. To do this, the authors introduce the notion of aggregate traffic envelopes.

Overall, existing admission control solutions may be difficult to calibrate (and very often no clue is given to calibrate them), and their performance may be greatly dependent on the traffic condition.

III. Knowledge-Based Admission Control scheme

As opposed to MBAC solutions, our KBAC solution includes an additional step, namely the *Knowledge Plane*, that comes in between the measurement algorithm and the decision algorithm. We now detail each of these steps.

A. Measurement algorithm

The measurement algorithm continuously monitors the activity of the communication link so as to collect measurement data. These data are measured on a short time window W_T, and hence reflect the "instantaneous" behavior of the on-going traffic. For each time window of length 200 ms, we measure the actual throughput of the on-going traffic, denoted by X (packets/ms), together with another QoS performance parameter, say P (*i.e.*, packet delay, packet loss rate). The measured values of X and P are gathered together into one pair of measurements. We refer to the pairs of measurements, (X,P), as *measurement points*.

B. Knowledge Plane

Once *measurement points* have been collected, we aim at characterizing the evolution of P as a function of X, denoted by $P = f_P(X)$. This second part of our KBAC solution consists itself of two phases.

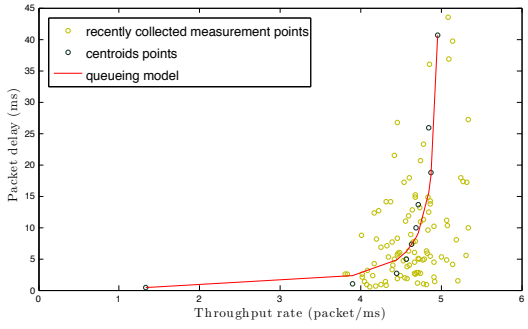

Fig. 1. Example of a Knowledge Plane, where P is the packet delay

First, we aim to partition n *measurement points* into k clusters in which each *measurement point* belongs to the cluster

with the nearest mean. To do this, we use *K-means* clustering method [17]. Elements within a cluster are represented by a single point, denoted by *centroid point*. We thus end up with k *centroid points*.

Second, using *Begin et al.* method [4], we attempt to automatically discover a queueing model that correctly reproduces the behavior exhibited by *centroid points*. The parameters of the discovered queueing model are automatically determined accordingly. In our work, we limit the search for the model to a single server queue model, namely, the *M/G/1* queue when we deal with the packet delay, and the *M/G/1/K* queue when we deal with the packet loss rate. The discovered queue supplies the function f_P, which is of utmost importance for our *Knowledge Plane*. This method requires about 10 points to operate. Therefore, in our experiments, we use 10 *centroid points*. Though we do not have a formal proof, in all our experiments, the modeling method always succeeds to provide an adequate and good fitting queueing model f_P.

Figure 1 illustrates the measurement methodology described above. It shows an example of how we discover a queueing model, f_P, whose performance match as closely as possible those known from the *centroid points*. We observe that a single *M/G/1* queue (with a mean rate of service of 5.01 packets/ms, a coefficient of variation equal to 2.02 and an offset equal to 0.08, see [4] for more details) adequately reproduces the behavior exhibited by *centroid points*. Note that, in this example, P corresponds to the packet delay.

Obviously the centroid points and the discovered queueing model, f_P, that drives the behavior of the link need to be regularly updated. A significant asset of our method stems in this update, which guarantees that f_P adapts its evolution to the real traffic condition. In our experiments, we update the centroid points and f_P every period T_{kp}. We let T_{kp} be equal to the highest value we found that yields to an up-to-date broad view of the link state, namely $T_{kp} = 20$ s

C. Decision algorithm

Finally the decision algorithm, which determines whether to accept or not a flow, is based on a performance prediction. Thanks to the function f_P delivered by the queueing model, it attempts to adequately estimate the expected performance of the link if the traffic workload was to be increased by this new flow.

Let $\widehat{P}$ be the expected value of P if a new flow requesting admission, with a peak rate r, is accepted. Then we have:

$$\widehat{P} = f_P(\widehat{X} + r) \tag{1}$$

where f_P defines the evolution of P against the throughput X, and $\widehat{X}$ reflects the adjusted throughput of the on-going traffic. Note that, we use an adjusted value of the throughput to avoid the erratic behavior of X, since it is measured on a short time window W_T. We explain later on how $\widehat{X}$ is estimated.

It follows that our decision algorithm can be formalized as: a new flow is accepted if

$$\widehat{P} + \alpha\widehat{\sigma}_p < P^* \tag{2}$$

where P^* represents the target performance (recall that, it is typically a maximum tolerable delay or loss rate), $\widehat{\sigma}_p$ is the standard deviation of $\widehat{P}$, as delivered by the discovered queueing model, and α is a conservativeness tuning parameter.

We now detail carefully the parameters listed above. The value of α is set so that on-going flows do not exceed the QoS target, with a probability Q. We define α using the one-sided *Chebyshev's inequality* [1]. Typically, we set the value of α to 1.7, so that $Q = 0.75$.

In the current form of our KBAC solution, we simply consider $\widehat{\sigma}_p = \widehat{P}$ (which is true, if we assume an exponential distribution with mean $\widehat{P}$).

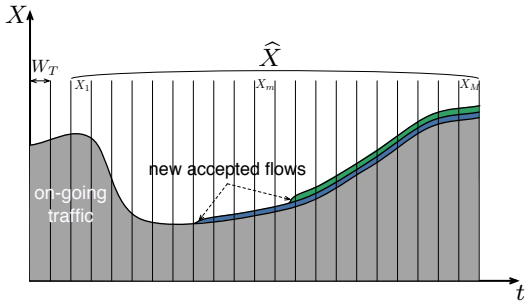

Fig. 2. Estimation of the adjusted throughput $\widehat{X}$

$\widehat{X}$ is computed over the last M measurement windows of length W_T, as follows:

$$\widehat{X} = \frac{1}{M}\sum_{m=1}^{M} X_m + \sum_{m=1}^{M}\frac{m}{M} \times \sum_{f=1}^{F_m} r_m^f \tag{3}$$

where X_m is the value of the throughput computed over the m^{th} measurement window, F_m is the number of accepted flows over the m^{th} measurement window and r_m^f is the estimated peak rate of the f^{th} new flow in the m^{th} measurement window. Figure 2 illustrates the computation of $\widehat{X}$. By doing so, we provide a smooth throughput $\widehat{X}$, comparable to X, that takes into account a ponderation of the peak rate of the accepted flows added to the average value of the throughput. Of course the value of $\widehat{X}$ needs to be regularly updated. In our experiments we update its value at the end of each measurement window W_T.

We also provide a means to accommodate the potential burstiness of traffic (*i.e.*, several new flows arrive within a measurement window W_T). Whenever a new flow, with a peak rate r, is accepted, the value of $\widehat{X}$ is immediately updated to be $\widehat{X} + r$.

D. Avoid the flood of information while ensuring centroids diversity

As said above, the *Knowledge Plane* maintains in real time an up-to-date broad view of the link state. This knowledge is obtained through the *measurement points*. Given the huge number of collected *measurement points* (*e.g.*, 300 new *measurement points* per minute with $W_T = 200$ ms), our KBAC solution will rapidly be overwhelmed by *measurement points*

when computing *centroid points*. To avoid this flood of information, we limit our focus to a subset made of n of these points.

On the other hand, limiting the number of *measurement points* may cause a loss of information (since the n points may fall in the same range of throughput). To address this problem, we split the throughput interval $[0,\ X_{max}]$ into S intervals of equal length. Each *measurement point* necessary belongs to one of these intervals. After each measurement window W_T, we replace the oldest *measurement point* by the latest computed *measurement point*, while ensuring that there are at least n_s *measurement points* within each throughput interval. By doing so, we both avoid the flood of information and ensure the centroids diversity which is required for adequately discovering the queueing model. Figure 1 illustrates the centroids diversity. Although recently collected *measurement points* are concentrated at the highest level of the throughput, *centroid points* are widely distributed and covers a broader range.

It is also worth noting that KBAC solution requires a warm-up period to ensure a wide enough distribution of the *centroid points*. This warm-up period typically lasts for less than a couple of minutes for an active link.

In our experiments, we limit the number of *measurement points* to 1000 ($n = 1000$) and we select $S = 6$ and $n_s = 20$.

E. Temporal coherence

As a matter of fact, our solution relies on the assumption of a temporal coherence for the behavior of a communication link. We suppose that, within a certain period of time T' (typically tens of seconds), the observed performance afford a precious information for accurately predicting the future performance of the communication link. Said differently,

$$\forall (t_1, t_2) \in [t, t+T']^2,\ X_{t_1} = X_{t_2} \Rightarrow P_{t_1} \simeq P_{t_2} \qquad (4)$$

where X_{t_1} (resp. X_{t_2}) is the throughput of the on-going traffic over the communication link at time t_1 (resp. t_2), and P_{t_1} (resp. P_{t_2}) is the performance parameter at time t_1 (resp. t_2).

IV. Experimental framework

In this section, we detail the framework we use to assess the behavior of our admission control.

A. Description of the scenario

We consider a communication link of capacity 10 Mb/s. The size of the buffer is set to 60 ms. The queueing discipline is FIFO (*First In First Out*) and the queue management algorithm is Drop-Tail.

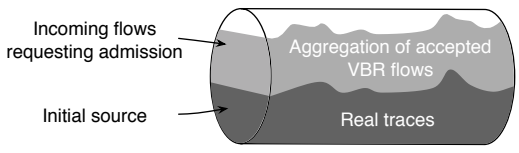

Fig. 3. On-going traffic conditions over the communication link

In our experiments, the on-going traffic is as a two-layered process. It consists of an initial source, to which is summed up the aggregation of VBR flows accepted by the admission control (see Figure 3). Note that this initial traffic is sent without admission control. It can correspond, for instance, to priority traffic or VPN traffic under no or limited access control, or to previous flows already accepted. By doing so, we guarantee that (or at least, a proportion of) on-going traffic matches some key statistical properties (*e.g.*, long-range dependency, autocorrelation, etc.) of real-life IP networks. We now detail how we model each of the two processes involved in the on-going traffic.

1) Initial source: We choose to represent the initial on-going traffic by a real traffic trace. We consider two traces coming from different networks. Trace 1 was gathered by the University of Stuttgart [16] on Sunday October 31^{st} 2004, between 6pm and 10pm, on a 100 Mb/s link in the dormitory network "Selfnet". Trace 2 was collected by the University of Brescia [10] on three consecutive working days in September/October 2009, on a 100 Mb/s link in the edge router of the campus network. In our experiments, we adjust each trace to a 10 Mb/s link by scaling it down such that its average rate of transmitted packets is equal to 2.5 Mb/s.

2) Incoming flows: Each incoming flow that requests access to the communication link will generate variable bit rate (VBR) traffic. Departures times of its packets are determined as follows: with a probability p, the next packet departure is scheduled t_p milliseconds later after the previous packet, and with a probability q = 1 - p, the next packet departure occurs t_q milliseconds later. Overall, the average sending rate of each VBR flow is given by:

$$\overline{r} = \frac{p}{t_p} + \frac{1-p}{t_q} \qquad (5)$$

In our experiments, we select p = 0.95, t_q = 28 $\times$ t_p and a constant packet size equal to 190 bytes. Hence, each VBR flow will generate packets with an average sending rate $\overline{r}$ of 64 kb/s and a coefficient of variation equal to 2.5 (remind that it is 0 for a CBR flow and 1 for a Poisson source).

The VBR flows arrive randomly to the communication link according to a Poisson process with a constant rate, denoted by γ. Their durations are drawn from an exponential distribution with mean d_{vbr}. Then, if no admission control were to be performed, the cumulated sending rate of VBR flows would be equal to:

$$\Lambda_{vbr} = \overline{n}.\overline{r} \qquad (6)$$

where $\overline{n} = d_{vbr}.\gamma$ (Little's law [1]) represents the average number of VBR flows over the communication link (without any admission control policy). We choose d_{vbr} = 120 s and γ = 0.717 arrivals per second. Hence, we have: $\Lambda_{vbr} =$ 5.5 Mb/s.

As said above, the initial source has an average rate of 2.5 Mb/s. The total sending rate of the initial source and VBR flows would be of 8 Mb/s if no admission control is performed. With such a level of workload, QoS can not be guaranteed

since accepting all flows requesting admission leads to congestions which in turn would yield to excessively high levels of packet delays.

It is then the goal of admission control to limit the number of VBR flows so as to keep the total rate of all combined traffic at the "right" level, and thus preventing packets from experiencing excessive queueing time in the buffer.

B. Estimating the peak rate of incoming flows

In our experiments, we assume no explicit knowledge on incoming flows. In some cases, this knowledge can be obtained via signaling and/or the use of a token bucket. However, token buckets are difficult to parameterize and may induce conservative results for the admission control (since the decision algorithm uses a conservative value for r). In this work, we opt rather for a simple approach that does not need any signaling as it is only based on data packets. We detail here the procedure we implement to let the network estimate the peak rate of a new flow requesting admission.

To estimate the peak rate of a new incoming flow, we track the first A packets of this flow[1]. We use a sliding window of length equal to a packets. For every possible window on the first A packets, we compute the average rate. Finally, the peak rate corresponds to the highest value among the $(A - a + 1)$ windows. In this work, the estimated peak rate of an injected flow is computed based on the 20 first packets (A = 20) with a sliding window of length equal to 5 packets (a = 5). Note that in our experiments, the VBR flows may achieve a maximal rate of 150 kb/s.

C. Ideal admission control

For sake of comparison, we include in our experiments the results that should be obtainable by an *ideal admission control* so as to benchmark the performance of our new KBAC solution. This *ideal admission control* should accept the maximum number of flows, thus achieving the maximum utilization rate, while successfully meeting the QoS target (*i.e.*, neither false positives nor false negatives). Note that this *ideal admission control* can be viewed as an "Oracle" since its requires knowledge, not only of the past and the present, but also of the future incoming flows.

Given the huge number of flows coming into the link during the numerical experiment (more than 1000), an exhaustive approach that will consider every feasible combination of accepted / rejected flows will lead to approximately $2^{1000} \simeq 10^{301}$ possible sequences, and thus would be intractable. We rather rely on an iterative method to determine the sequence of flows accepted by the ideal admission control under the policy *First come, First served* (if the flow does not violate the QoS target). At iteration (i), k flows have been accepted (some of them may still be going on) and j have been refused. As soon as a new flow will arrive, we will accept it, and then we will keep the simulation running until this flow ends but, meanwhile, any subsequent VBR flow will be refused. Once the flow is done, we check whether the QoS target was preserved for this flow as well as for any previously accepted flow. If this is the case, then we grant this flow as acceptable by the ideal admission control and the value of k is incremented. Otherwise, the flow will not be part of the sought sequence of flows and j is incremented.

[1]This property implies that a flow can be rejected even though its first packets were transmitted.

D. Investigated MBAC solutions

In this section, we outline the two investigated MBAC solutions which we will use to compare to our new admission control solution. We limit both solutions to the case of delay.

1) Measured Sum (M.S.): This solution rejects an incoming flow requesting admission if admitting this new flow violates the following constraint:

$$\widehat{D} + \frac{b_i}{C} < D, \tag{7}$$

where D is the delay bound, $\widehat{D}$ is the measured delay and b_i is the burstiness of the flow (see details in [12]). The measured delay, denoted by $\widehat{D}$, tracks the maximum queueing delay of every packet computed over a time window of length T. The value of $\widehat{D}$ is updated at the end of each measurement window. Whenever an individual delay measurement exceeds the estimated maximum queueing delay, the value of $\widehat{D}$ is also updated to be λ times this sampled delay. Finally, we update the measured delay to the left side of (7), whenever a new flow is admitted. In our experiments, we select $T = 4$ s.

2) Aggregate Traffic Envelopes (Env.): Qiu and Knightly present in [15] a MBAC solution that aims to characterize the aggregate traffic rate by the maximal rate envelope. To do this, they consider a time window of length T divided into t sampling periods of equal length. Within a time window, maximal rate measurements are done on different time scales. R_l^m represents the maximal observed rate in the time scale l. This time scale is equal to l sampling periods in the m^{th} measurement window. The rate of the aggregate traffic and its standard-deviation are estimated over the last M measurement windows as follows:

$$\overline{R}_l = \sum_{m=1}^{M} \frac{R_l^m}{M} \text{ and } \sigma_l^2 = \frac{1}{M-1} \sum_{m=1}^{M} (R_l^m - \overline{R}_l)^2. \tag{8}$$

This measurement-based admission control ensures that no packet is too long delayed. A new flow requesting admission with a peak rate r is accepted if and only if:

$$\max_{l=1,\ldots,t} \{l\tau(\overline{R}_l + r + \alpha_E \sigma_l - C)\} \leq C \times D \tag{9}$$

where D is the maximum delay requirement and α_E is a constant specifying the confidence level, $\Phi(\alpha_E)$, that on-going flows do not experience any packet loss. $\Phi(\alpha_E)$ is defined as:

$$\Phi(\alpha_E) \approx \frac{1}{\sqrt{2\pi}\sigma_l} \int_{-\infty}^{\overline{R}_l + \alpha_E \sigma_l} \exp\left(-\frac{(r - \overline{R}_l)^2}{2\sigma_l^2}\right) \mathrm{d}r. \tag{10}$$

In our experiments, we consider 20 measurement windows ($M = 20$) of length 200 ms ($T = 200$ ms), which is inline with the considered measurement window for the *Measured Sum*. Note that we also evaluate our KBAC solution under these conditions.

TABLE I
NUMERICAL VALUES OF THE PARAMETERS USED FOR EACH ADMISSION CONTROL SOLUTION

		KBAC	Measured Sum	Aggregate Traffic Envelopes
Measured quantities	**Aggregated rate**	$\widehat{X}$	-	$\overline{R_k}$ $(k = 1, \ldots t)$
	History	$M = 20$	-	$M = 20$
	Standard-deviation	-	-	σ_k $(k = 1, \ldots t)$
	History	-	-	$M = 20$
	Estimated delay	P	$\widehat{D}$	-
	History	Single measurement window	Single measurement window	-
	Measurement window	$T = 200$ ms	$T = 4$ s	$T = 200$ ms
				10 ms *Sampling periods* $(t = 20)$
Knowledge Plane	**Time window**	$T_{kp} = 20$ s	-	-
		$k = 10$, $S = 6$, $n_s = 20$	-	-
		$n = 1000$	-	-
Calibrated parameters	**Admission threshold**			
	D^* : 10 ms or 20 ms	$P^* = D^*$	$D = D^*$	$D = D^*$
	Tuning parameter	$\alpha = 1.7$	$\lambda = 1$ or $\lambda = 2$	$\alpha_E = 0.01$, $\alpha_E = 1.3$ or $\alpha_E = 3.62$

V. NUMERICAL RESULTS

In this paper, we limit our experimental framework to the case of an admission threshold expressed as a maximum tolerable delay (another option would consist in considering packet loss rate instead). We consider two different values of the target delay, namely $D^* = 10$ ms and $D^* = 20$ ms. We evaluate the performance of our KBAC solution using ns-3 simulations. Each simulation is run for a period of 30 minutes. It is also worth noting that we compare our KBAC solution with two other solutions (*i.e.*, *Measured Sum* and *Aggregate Traffic Envelopes*), and with an *ideal admission control*. Table I relates the parameter values selected for each solution.

To properly assess the behavior of each admission control, we consider several metrics: (i) the "instantaneous" values of the packet delay computed on a sliding window of length equal to 4 s; (ii) the percentage of accepted flows; (iii) the percentage of violation that represents the ratio of time during which the QoS target is violated. These two latter values are computed over the entire duration of the simulation.

Recall that in our scenario (Section IV-A), we consider two traces coming from different networks to represent the initial source. The remainder of this section details simulation results for each case.

A. Calibration of the admission control algorithms according to a target delay, D^*

We describe here how we parameterize each admission control according to a target delay. Each admission control has one tuning parameter to adjust the stringency level. Broadly speaking, the greater these values, the more conservative the admission control is, and the less accepted flows.

1) KBAC: We simply set P^* equal to the target delay, D^*. Referring to the one-sided *Chebyshev's inequality*, the selected value for the tuning parameter, α, determines the expected probability Q, that on-going flows do not exceed the QoS target. In our KBAC solution the value of α is equal to 1.7, so that $Q = 0.75$.

2) Measured Sum: We set D equal to the target delay, D^*. As no specific guidelines are given by the authors in [12] for setting the value of λ ($\lambda \geq 1$), we consider two different values, namely $\lambda = 1$ and $\lambda = 2$.

3) Aggregate traffic envelopes: We set D equal to D^*. There is no clear recommendation from the authors in [15] on the choice for α_E. In our experiments, we consider three different values for α_E. Therefore, we respectively set $\alpha_E = 0.01$, $\alpha_E = 1.3$, and $\alpha_E = 3.62$ (leading the confidence level $\Phi(\alpha_E)$ equal to 0.5, 0.9, and 0.9999, respectively).

B. Trace 1

First, we consider the case where the initial source is represented using "Selnet" traffic traces gathered by the University of Stuttgart [16].

Figure 4 represents the instantaneous packet delay with regards to the target delay, $D^* = 10$ ms, for each admission control. Our KBAC solution yields satisfactory results since it almost constantly meets the target delay. It is also worth noting that it exhibits a behavior roughly close to the ideal admission control. More specifically, our solution fulfills the admission threshold more than 97 % of the time. The *Measured Sum* solution leads to steadily and excessively low levels of packet delay. In Figure 4, we depict the less conservative case (*i.e.*, $\lambda = 1$), typically more than 9 ms below D^*. The results of the *Aggregate Traffic Envelopes* solution widely differ depending on the specific tuning parameter. For $\alpha_E = 0.01$ (labeled as *ENV.1*), it severely and almost constantly violates the target delay. For $\alpha_E = 1.3$, it leads to conservative behavior. For $\alpha_E = 3.62$, not represented in Figure 4, it leads to too overwhelmingly conservative behavior.

Table II relates complementary results on the overall performance of each admission control solution. Several observations can be made. First, it indicates that our KBAC solution leads to a number of accepted flows (*i.e.*, 357 flows) close to the one delivered by the ideal admission control (*i.e.*, 379 flows). Second, it states that the *Measured Sum* solution always accepts significantly less flows (138 and 133, respectively) than the ideal admission control when λ is set to 1 and 2, respectively. Finally, when we deal with the *Aggregate Traffic Envelopes* solution, the number of accepted flows widely differs depending on the selected value of the tuning parameter

TABLE II
ADMISSION CONTROL SOLUTIONS PERFORMANCE OVERALL THE SIMULATION TIME USING TRACE 1

	KBAC	**Measured Sum**		**Aggregate Traffic Envelopes**			**Ideal Admission Control**
		$\lambda = 1$	$\lambda = 2$	$\alpha_E = 0.01$	$\alpha_E = 1.3$	$\alpha_E = 3.62$	
Number of accepted flows	357	246	241	400	310	156	379
Percentage of accepted flows	28.18%	19.42%	19.02%	31.57%	24.47%	12.31%	29.91%
Percentage of violation	2.22%	0%	0%	55.11%	0%	0%	0%

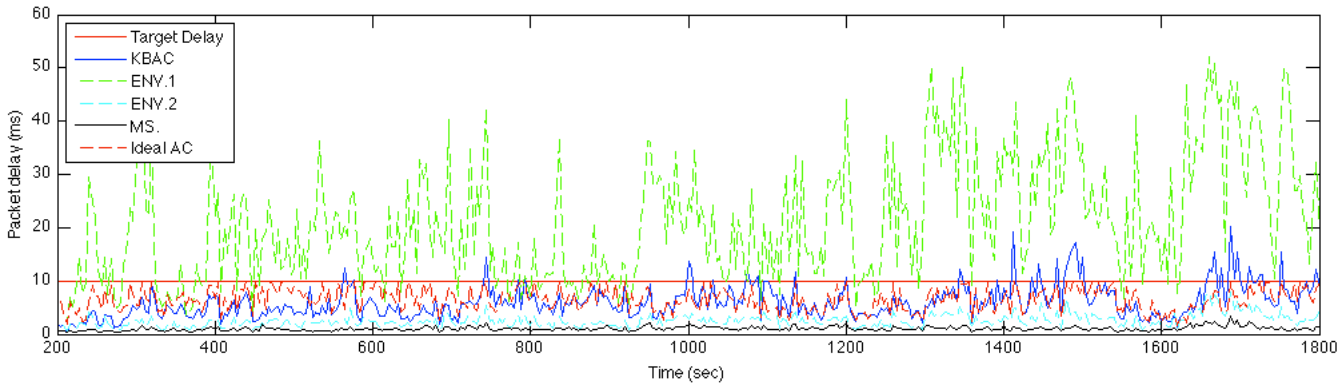

Fig. 4. Instantaneous performance of admission control solutions using trace 1

α_E. It respectively accepts around 70 and 220 flows less than the ideal admission control for $\alpha_E = 1.3$ and 3.62, respectively. On the other hand, when α_E is set to 0.01, this latter solution accepts an exceedingly large number of flows which results in frequent instantaneous packet delays much above the admission threshold (up to 55% of time).

C. Trace 2

We now consider the case where the initial source is represented using traffic traces gathered from the University of Brescia [10]. Figure 5 and Table III relate the results for a target delay $D^* = 20$ ms.

Figure 5 depicts the instantaneous packet delay obtained by each admission control solution. Our KBAC solution fulfills the admission threshold about 75 % of the time (Table III). On the other hand, when the QoS target is violated, one should note that it lasts only for relatively short periods of time (typically less than 10 s). Furthermore, the magnitudes of these departures are generally of moderate size (less than 10 ms over D^*). This result highlights the ability of our KBAC solution to rapidly and automatically adjust its admission policy according to the actual variations on the traffic conditions. The *Measured Sum* solution leads to a slightly more conservative behavior which is in line with previous results (Section V-B). Regarding the *Aggregate Traffic Envelopes* solution, its results widely differ depending on the specific tuning parameter. For $\alpha_E = 0.01$ and $\alpha_E = 1.3$, it severely violates the target delay. For $\alpha_E = 3.62$, not represented in Figure 5, it leads to very low level of packet delay.

We now turn to Table III. It states that the ideal admission control can accept up to 406 flows. It also indicates that our KBAC solution leads to a number of accepted flows (*i.e.*, 400 flows) close to the one delivered by the ideal admission control. Finally, when we deal with the investigated solutions, the number of accepted flows widely differs. The *Measured Sum* solution accepts around 32 flows less than the ideal admission control. Furthermore, the *Aggregate Traffic Envelopes* accepts an exceedingly large number of flows when α_E is equal to 0.01 and 1.3 or alternatively less number of flows when α_E is set to 3.62 than the ideal admission control.

To conclude, it is worth noting that in the case of *Aggregate Traffic Envelopes* a given value of α_E can lead to an exceedingly conservative behavior or conversely to a weak control depending on the nature of the traffic. This underlines the difficulty of precisely calibrating key parameters so as to fulfill the QoS target. Our solution avoids this critical step.

D. Computing overhead

In this section, we attempt to quantify the computational overhead of our KBAC solution. To begin with, we focus on the computational complexity brought by its measurement process. Clearly, the number of measurement points collected meanwhile is M. Note that it is also M for the *Measured Sum* solution whereas it is approximately $M^2/2$ for the *Aggregate Traffic Envelopes*.

Now we turn to the complexity brought by the *Knowledge Plane*. First, the complexity of the *K-means* clustering algorithm is well-known to be in $O(k.n.t)$, where k is the number of *centroid points*, n the number of *measurement points* to be classified, and t the number of iterations (typically, $t << n$). Second, regarding the discovery of the fitting queueing model f_P, the search for parameter values greatly depends on the number of parameters and on the number of centroids points. In our case, since we are considering a simple *M/G/1* queue with 3 parameters and 10 *centroids points*, the appropriate queue is quickly found. See [3] for more details. Thus, and not surprisingly, our KBAC solution, which includes an extra stage (i.e., the *Knowledge Plane*), leads to additional overheads

TABLE III
ADMISSION CONTROL SOLUTIONS PERFORMANCE OVERALL THE SIMULATION TIME USING TRACE 2

	KBAC	**Measured Sum**		**Aggregate Traffic Envelopes**			**Ideal Admission Control**
		$\lambda = 1$	$\lambda = 2$	$\alpha_E = 0.01$	$\alpha_E = 1.3$	$\alpha_E = 3.62$	
Number of accepted flows	400	374	358	471	427	351	406
Percentage of accepted flows	31.57%	29.52%	28.26%	37.17%	33.7%	27.7%	32.04%
Percentage of violation	26.2%	4%	1.78%	74.67%	64.44%	0.89%	0%

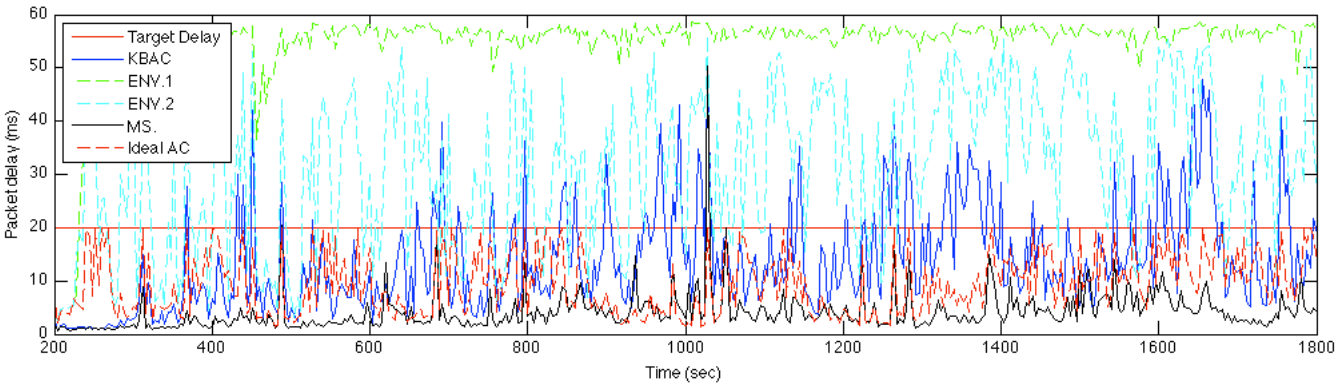

Fig. 5. Instantaneous performance of admission control solutions using Trace 2

as compared to other investigated solutions. In practice, given the high-computational capabilities of routers, the execution of this knowledge plane, which is run every 20 sec in our experiments, should yield an overhead that could be managed by forthcoming routers.

VI. CONCLUSION

In this paper, we introduce a novel data-driven method based on a time-varying model that we refer to as Knowledge-Based Admission Control solution (KBAC). This method consists of three main stages: (i) collect measurements on the on-going traffic over the communication link; (ii) maintain an up-to-date broad view of the link behavior, and feed it to a *Knowledge Plane*; (iii) model the observed link behavior by a mono-server queue whose parameters are set automatically and which predicts the expected QoS if a flow requesting admission were to be accepted.

Unlike existing admission control solutions, our solution avoids the critical step of precisely calibrating key parameters. We demonstrate through simulations, for two different real traces with different features (one being more volatile), the ability of our KBAC solution to provide a fair probabilistic guarantee and a good trade-off between flow performance and resource utilization when the admission threshold is expressed as a bounded delay. This ability stems from the quick and automatic adjustment of its admission policy according to the actual variations on the traffic conditions.

Future work will mainly be devoted to the study of the behavior of our KBAC solution in the case of an admission threshold expressed as a maximum tolerable packet loss rate.

REFERENCES

[1] A. O. Allen, "Probablity, Statistics and Queueing Theory with Computer Science Applications, Second Edition," in *Int. CMG Conference*, 1990.

[2] D. Ammar, T. Begin, I. Guérin-Lassous, and L. Noirie, "Evaluation and comparison of MBAC solutions," in *Proceedings of the 36th Conference on Local Computer Networks, IEEE LCN 2011*, Germany, Oct. 2011.

[3] T. Begin, B. Baynat, A. Brandwajn, and F. Sourd, "A DFO technique to calibrate queueing models," *Computers & Operations Research*, vol. 37, no. 2, pp. 273 – 281, February 2009.

[4] T. Begin, A. Brandwajn, B. Baynat, B. Wolfinger, and S. Fdida, "High-level approach to modeling of observed system behavior," *Performance Evaluation*, vol. 67, no. 5, pp. 386 – 405, May 2010.

[5] L. Breslau, S. Jamin, and S. Shenker, "Comments on the Performance of Measurement-Based Admission Control Algorithms," in *Infocom*, 2000.

[6] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint admission control: architectural issues and performance," *SIGCOMM Comput. Commun. Rev.*, vol. 30, Aug. 2000.

[7] S. Floyd, "Comments on Measurement-based Admissions Control for Controlled-Load Services," Tech. Rep., 1996.

[8] S. Georgoulas, P. Trimintzios, G. Pavlou, and K. Ho, "An integrated bandwidth allocation and admission control framework for the support of heterogeneous real-time traffic in class-based IP networks," *Computer Communications*, vol. 31, pp. 129–152, 2008.

[9] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE JSAC*, vol. 9, no. 7, pp. 968–981, Sep 1991.

[10] T. II, "Brescia university," http://www.ing.unibs.it/ntw/tools/traces/.

[11] S. Jamin and P. B. Danzig, "A measurement-based admission control algorithm for integrated services packet networks," *IEEE/ACM Transactions on Networking*, vol. 5, no. 1, pp. 56–70, Feb 1997.

[12] S. Jamin, S. Shenker, and P. B. Danzig, "Comparison of Measurement-based Admission Control Algorithms for Controlled-Load Service," in *Infocom*, 1997.

[13] A. W. Moore, "An implementation-based comparison of Measurement-Based Admission Control algorithms," *J. High Speed Netw.*, vol. 13, April 2004.

[14] A. Nevin, Y. Jiang, and P. J. Emstad, "Robustness Study of MBAC Algorithms," in *ISCC*, 2008.

[15] J. Qiu and E. W. Knightly, "Measurement-Based Admission Control with Aggregate Traffic Envelopes," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 199–210, Apr 2001.

[16] D. Sass, "Internet traces of the"Selfnet" university dormitory network, Trace UST2, University of Stuttgart, Trace UST2," 2004.

[17] G. Seber, *Multivariate observations*, ser. Wiley series in probability and mathematical statistics. New York, NY [u.a.]: Wiley, 1984.