Measurement-Based Admission Control for a Flow-Aware Network

Yuming Jiang [a], Peder J. Emstad [a,b], Anne Nevin [a,b], Victor Nicola [c], Markus Fidler [d]
[a] Center for Quantifiable Quality of Service in Communication Systems; [b] Department of Telematics
Norwegian University of Science and Technology, Norway
[c] Department of Electrical Engineering, University of Twente, The Netherlands
[d] Department of Computer Science, RWTH Aachen University, Germany
ymjiang@ieee.org, {peder, anne.nevin}@q2s.ntnu.no, nicola@ewi.utwente.nl, fidler@ieee.org

***Abstract*— To provide statistical service guarantee and achieve high network utilization, measurement-based admission control (MBAC) has been studied for over one decade. Many MBAC algorithms have been proposed in the literature. However, most of them belong to *aggregate MBAC* algorithms which assume or require that (1) First-In-First-Out (FIFO) is used for aggregating flows; (2) statistical service guarantees are provided to the aggregate of admitted flows; (3) each flow requires and experiences the same statistical service guarantees as the aggregate.**

In this paper, we focus on *per-flow MBAC* that aims to provide possibly different statistical service guarantees to individual flows in an aggregate. Particularly, we propose a *simple* per-flow MBAC algorithm in which dynamic priority scheduling (DPS) is adopted to aggregate flows. With this DPS-based per-flow MBAC algorithm, a newly admitted flow is always given a lower priority level than all existing flows, and its priority level is improved if an existing flow leaves the system. Consequently, once a flow is admitted, its received service will not be adversely affected by other flows admitted after it. Because of this, there is no need to re-check or adjust network resources allocated to existing flows due to the admission of a new flow.

I. Introduction

While many network applications such as VoIP and streaming audio and video are both delay and loss sensitive, they can tolerate some delay and loss. As a result, statistical service guarantees have attracted a lot of research interest in the past decade. For providing statistical service guarantees (SSGs), *measurement-based admission control* (MBAC) has long been recognized as an important technique because of its ability in achieving high network utilization when ensuring SSGs.

In measurement-based admission control, an MBAC algorithm uses the *a priori* source characterizations only for incoming flows and for existing flows that have been in the system, it uses measurements to characterize them. In the literature, many MBAC algorithms have been proposed and investigated [15][6][23]. While these algorithms use different analytical bases for admission test, they commonly assume or require that [15] *(1) FIFO is used for aggregating flows; (2) statistical service guarantees are provided to the aggregate of admitted flows; (3) each flow in the aggregate requires and experiences the same statistical service guarantees as the aggregate.* We call these algorithms *aggregate MBAC* algorithms. While the above assumptions and requirements have made current MBAC algorithms simple, networked multimedia applications are so diverse that their quality of service (QoS) requirements can be far from each other. In such cases, *per-flow MBAC* algorithms are preferred [22].

In this paper, we focus on providing SSGs to individual flows. Particularly, we consider a flow-aware network where each flow may have different SSG requirements. We propose an MBAC algorithm that adopts priority to schedule flows in the system and uses measurements of existing traffic to determine if or not a requesting flow can be admitted based on its required SSGs. A newly admitted flow is always given the lowest priority and its priority level is improved if an existing flow leaves the system or other flows are admitted after its admission. For the admission control, both delay and loss are taken into account. Analytical results for the proposed dynamic priority scheduling (DPS) MBAC are presented. The proposed DPS MBAC implies that the experienced SSGs of an admitted flow is not adversely affected by flows admitted after it.

The rest is organized as follows. Sec. II introduces flow-aware networking briefly. Sec. III provides some preliminaries for analysis. Sec. IV introduces the DPS MBAC algorithm and its analysis. Sec. V presents some numerical results. Finally, conclusion is made in Sec. VI.

II. Network Model

A. Flow-Aware Networking

We consider a flow-aware network. In the network, a flow is defined to be and identified as a set of packets related to an instance of some network application observed at a given network point with an inter-packet interval less than a certain *time-out* period. Specifically, a flow consists of packets having the same values in certain header fields. A flow is said to have ended or left when no packet with the same header field values is observed for the time-out period. There are several possible ways to identify a flow. One is to use the five-tuple of IP addresses, protocol and port numbers. Another is to use the flow label field in the IP header as specified by IPv6 associated with the source and/or destination addresses. Here, we simply assume each flow can be identified, but how this is done is out of the scope.

Flow-aware networking was proposed recently as an alternative QoS architecture for the Internet [5][22]. While an IntServ network also requires flow-level identification, per-flow service guarantees are mainly provided in the deterministic Guaranteed Service manner, which can cause significant under-utilization of network resources as pointed out by researchers (e.g. [22] and references therein). Although IntServ has defined Controlled Load service for utilizing statistical multiplexing gain to achieve higher network utilization, the requirement of a signaling protocol and using TSpec (token bucket traffic specification) for specifying the traffic has imposed significant constraints to customers and limited its use [22].

A flow-aware network is designed to achieve high network utilization and provide SSGs without the need of using signaling and TSpec. Particularly, it achieves this by assuming that the amount of traffic generated by an individual flow is always less than a certain ratio of the total capacity on a link [22]. Let C be the total link capacity and α_n be the ratio. Then, this assumption implies that approximately the cumulative amount of traffic $A_n(t)$ generated in $[0,t]$ by the flow satisfies,

$$A_n(t) \le \alpha_n C t. \tag{1}$$

Using (1) as the *implicit* traffic descriptor of an incoming flow, no signaling is needed to convey explicit traffic information from the sender to the network. In addition, the service requirement of the flow may be implicitly set in the network or can be carried by some header field of the flow's first packet. Under DiffServ architecture, similar approach has been used. Particularly, the DSCP field carried by each packet tells each node along its path the service it requires [4]. The detailed way of mapping the header field to the service requirement is out of this paper's scope.

In flow-aware networks, per-flow MBAC is required [5] [22], which is different from MBAC for IntServ Controlled Load service where MBAC algorithms are aggregate-based.

We consider a single link in the flow-aware network and investigate MBAC on this link. For ease of exposition, we assume that there is only one traffic class, all considered flows belong to the same class and the total capacity and total buffer are allocated to this class. While this single-link single-class configuration as in [6] is rather simple, the basic performance aspects of MBAC to be investigated are most easily revealed.

B. Measurement-Based Admission Control

Fig. 1 depicts the MBAC structure that shows an MBAC algorithm typically includes three elements: (1) admission decision algorithm; (2) traffic estimator; (3) resource estimator. The MBAC algorithm keeps measuring traffic in the system and/or remaining system resources such as available bandwidth and buffer size. Based on the measurement, the traffic estimator estimates how much traffic is in the system and what its characteristics are; the resource estimator estimates how much resource remains. When a flow requests admission to the system, the MBAC algorithm uses the admission control algorithm to decide if this flow can be admitted. This decision is based on the inputs from the traffic estimator and the resource estimator. In addition, the decision also relies on some input from the requesting flow, which typically includes its quality of service requirement and its traffic description.

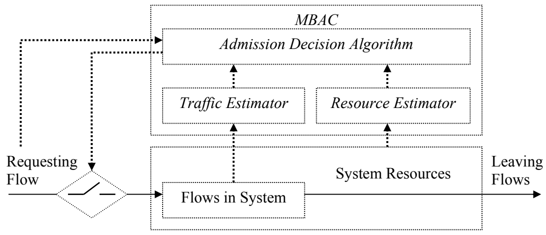

Fig. 1. Structure of MBAC

Fig. 1 and research in the literature (e.g. [6]) show that the three elements can be decoupled. In this paper, we focus on the admission decision part. In the rest, we simply use MBAC algorithm to refer to this part. In the past decade, many such MBAC algorithms have been proposed, and several reviews are available [23][6]. While using different approaches and/or different analytical bases, most of these MBAC algorithms assume FIFO aggregation and belong to aggregate MBAC.

In this paper, we focus on a flow-aware network. For such networks, flow level MBAC is the key to QoS control [5]. For this, flow aggregation approaches other than FIFO may be needed. We shall use dynamic priority scheduling (DPS) to aggregate flows and perform per-flow MBAC.

III. Preliminaries for Stochastic QoS Analysis

This section provides an introduction to a stochastic traffic model and the calculus developed based on this traffic model for stochastic QoS analysis. The traffic model is based on the generalized stochastically bounded burstiness (gSBB) concept [25]. Intuitively, in gSBB, the input process of a flow is modeled with its queue length distribution in a virtual system. The virtual system is a single server queue (SSQ) system with constant rate server and the same input flow. Specifically, let $A(s,t)$ and $Q(t;r) = sup_{s \le t}\{A(s,t) - r(t-s)\}$ respectively denote the amount of traffic arriving between s and t, and the queue length in the virtual SSQ system with constant service rate r at time t for the input process. For ease of exposition, we simply use $A(t)$ to replace $A(0,t)$ representing the cumulative traffic generated in $[0,t]$. Then, gSBB is defined as follows [25]:

Definition 1: Let BF be the set of all functions on $[0,\infty)$ such that $f(\cdot) \in BF$ implies that $1 - f(\cdot)$ is a distribution function. A stochastic process $A(t)$ is said to have a generalized Stochastically Bounded Burstiness (gSBB) with upper rate r and bounding function $f(x)(\in BF)$, iff for all $x \ge 0$ and $t \ge 0$, there holds

$$P\{Q(t;r) > x\} \le f(x).$$

Throughout the rest of the paper, we use the notation $A(t) \sim \langle f, r\rangle$ to denote that process $A(t)$ is gSBB with upper rate r and bounding function f. This corresponds to the notation $A(t) \sim (\sigma, \rho)$ used in the context of deterministic traffic model [9], which defines that the amount of traffic generated in any period $[s,t]$ is upper-bounded or $A(s,t) \le \sigma + \rho(t-s)$.

It has been shown in [25] that many types of traffic belong to gSBB, which include Poisson, Weibull and Gaussian types of traffic. In addition, many non-Gaussian types of traffic such as α-stable self-similar traffic are also gSBB [25].

A direct property of gSBB from its definition is that if a flow is gSBB with bounding function f and upper rate r, then it is gSBB with the same bounding function and any upper rate not less than r. Another interesting property of gSBB is that the multiplexing of multiple gSBB flows results in a gSBB (aggregate) flow [25][20].

In the remaining part of the paper, the following operator $\otimes$ is used, which is defined under the min-plus algebra [3] [18]:

Definition 2: Consider two functions $f(x)$ and $h(x)$.

$$f \otimes h(x) \equiv inf_{0\le y\le x}\{f(x-y)+h(y)\}.$$

Many properties of $\otimes$ have been proved [3] [18] [20]. One is the commutativity property [3] [18], which is

$$f \otimes h(x) = h \otimes f(x). \tag{2}$$

Another property, which we shall call the multiplexing property of $\otimes$, is that for any two stochastic processes $Q_1(t)$ and $Q_2(t)$, if $P\{Q_1(t) > x\} \le f_1(x)$ and $P\{Q_2(t) > x\} \le f_2(x)$, then [20]:

$$P\{Q_1(t)+Q_2(t) > x\} \le f_1 \otimes f_2(x). \tag{3}$$

Another concept that will be used for later analysis is stochastic service curve (SSC) [10][19] which is a general stochastic server model defined as follows:

Definition 3: A system is said to provide a stochastic service curve (SSC) $(\beta(t), g(x))$ to a flow $A(t)$, iff for all $t \ge 0$ and $x \ge 0$,

$$P\{A\otimes\beta(t) - A^*(t) > x\} \le g(x),$$

where $A^*(t)$ denotes the output process of the flow from the system, $\beta(t)$ is a non-negative non-decreasing function of t with $\beta(0) = 0$, $g(x)$ is a non-increasing function of x and $g(x) \ge 0$ for all x.

If the SSC of a system is known and the input process belongs to gSBB, then the following propositions can be used to derive the stochastic delay and buffer length in the system. Their proofs could also be found from [10] [19].

Proposition 1: Consider a system with an input flow. Suppose the flow has input process $A(t) \sim \langle f, r\rangle$ and the system provides a SSC $(\beta(t), g(x))$ to the flow. Let $B(t)$ denote the backlog of the flow in the system at time t. Then,

$$P\{B(t) > b(x)\} \le f \otimes g(x),$$

where $b(x) = sup_{t\ge0}\{rt + x - \beta(t)\}$.

Proof: By definition, $B(t) = A(t) - A^*(t)$, where $A^*(t)$ denotes the amount of output traffic up to time t. For any t, suppose $A \otimes \beta(t)$ reaches its minimum at some t_0. In other words, with the commutativity property (2) of $\otimes$, $A \otimes \beta(t) = \beta(t-t_0) + A(t_0)$. Then,

$$\begin{aligned} & B(t) - sup_{s\ge0}\{rs - \beta(s)\} \\ \le\ & A(t) - A^*(t) - r(t-t_0) + \beta(t-t_0) \\ =\ & A(t) - A(t_0) - r(t-t_0) + \beta(t-t_0) + A(t_0) - A^*(t) \\ \le\ & Q(t;r) + A\otimes\beta(t) - A^*(t). \end{aligned} \tag{4}$$

Note that in the last relation in (4), t_0 disappears and (4) holds for all such t_0. Consequently, for any t,

$$\begin{aligned} & P\{B(t) - sup_{s\ge0}\{rs-\beta(s)\} > x\} \\ \le\ & P\{Q(t;r) + A\otimes\beta(t) - A^*(t) > x\}. \end{aligned}$$

Since by gSBB definition, $P\{Q(t;r) > x\} \le f(x)$ and by SSC definition, $P\{A\otimes\beta(t) - A^*(t) > x\} \le g(x)$, then with the multiplexing property (3) of $\otimes$, we get, $P\{B(t) - sup_{s\ge0}\{rs-\beta(s)\} > x\} \le f\otimes g(x)$, which ends the proof. ∎

Proposition 2: Consider a system with an input flow. Suppose the flow has input process $A(t) \sim \langle f, r\rangle$ and the system provides a SSC $(\beta(t), g(x))$ to the flow. Let $D(t) \equiv inf_{\tau\ge0}\{\tau : A(t) \le A^*(t+\tau)\}$ denote the delay seen by the flow at time t. Then,

$$P\{D(t) > d(x)\} \le f\otimes g(x),$$

where $d(x) = sup_{t\ge0}\{inf\{\tau \ge 0 : rt + x \le \beta(t+\tau)\}\}$.

Proof: Consider any time t. By definition, $D(t) = inf_{\tau\ge0}\{\tau : A(t) \le A^*(t+\tau)\}$, which implies that, for any d, if $D(t) > d$, there must be $A(t) > A^*(t+d)$, since otherwise if there would be $A(t) \le A^*(t+d)$, we would have $D(t) \le d$ which contradicts the condition $D(t) > d$. In other words, if event $\{D(t) > d\}$ happens, event $\{A(t) > A^*(t+d)\}$ must happen.

We now assume $A(t) > A^*(t+d_t)$. Let $d(x) = sup_{t\ge0}\{inf\{\tau \ge 0 : rt + x \le \beta(t+\tau)\}\}$. Note that for any t, there exists some $t_0(\le t+d)$ that makes $A\otimes\beta(t+d)$ reaches its minimum: $A\otimes\beta(t+d) = A(t+d-t_0) + \beta(t_0)$.

Let $d_0 = d_0(s) = inf\{\tau \ge 0 : rs + x \le \beta(s+\tau)\}$ for any s. Clearly, $d(x) \ge d_0$ and by letting $s = t_0 - d_0$ in $d_0(s)$, $r(t_0-d_0) + x \le \beta(t_0)$.

We then have

$$\begin{aligned} & Q(t;r) + A\otimes\beta(t+d(x)) - A^*(t+d(x)) \\ \ge\ & [A(t) - A(t+d(x)-t_0) - r(t_0-d(x))] \\ & +[\beta(t_0) + A(t+d(x)-t_0)] - A^*(t+d(x)) \\ \ge\ & \beta(t_0) - r(t_0-d(x)) \ge \beta(t_0) - r(t_0-d_0) \\ \ge\ & x. \end{aligned} \tag{5}$$

Note that in the last step above, t_0 is no more there. We hence can conclude that for any t, if $A(t) > A^*(t+d_t)$ holds, then (5) holds. Consequently, we can further conclude that if $D(t) > d(x)$, then (5). Based on this, we get

$$\begin{aligned} & P\{D(t) > d(x)\} \\ \le\ & P\{Q(t;r) + A\otimes\beta(t+d(x)) - A^*(t+d(x)) > x\}, \end{aligned}$$

with which, the definition of gSBB, the definition of SSC, and the multiplexing property (3) of $\otimes$, the proposition is proved. ∎

IV. DPS-Based Per-Flow MBAC Algorithm

In this section, we present the dynamic priority scheduling (DPS) based per-flow MBAC algorithm.

A. Flow Scheduling and Buffer Allocation

In the proposed DPS MBAC, each admitted flow has its dedicated queue. Priority scheduling is performed among admitted existing flows in the system. An earlier admitted flow has higher priority over all later admitted flows. This is achieved by always giving the newly admitted flow a lower priority than all existing flows. When a flow is detected non-active for a certain time period, known as the time-out period, it is considered having left the system and its corresponding buffer is released together with its priority. A common buffer pool is maintained in the MBAC algorithm. When an incoming flow is admitted, a certain size of buffer is allocated to the flow. The allocated buffer size is determined based on the analysis presented later.

As can be seen from the above description, although the relative priority of a flow with respect to other flows is nearly fixed, its exact priority level is dynamic over time. In particular, the leave of an existing flow increases by one priority level all existing flows admitted after it.

B. Admission Control Algorithm

When the first packet of a new flow is detected, implying an incoming flow requesting for admission, the admission control algorithm admits the flow only when the following criteria are met:

$$r_n + \hat{r} \leq \alpha C \quad (6)$$

$$f_d(d_n) \leq \epsilon_n^d \quad (7)$$

$$b_n + \hat{b} \leq M \quad (8)$$

where C denotes the link capacity, $\alpha(< 1)$ the maximum allowed utilization level of C, M the total buffer size, $r_n(= \alpha_n C)$ the (implicit upper) rate of the incoming flow, ϵ_n^d the delay requirement, and b_n the buffer size that will be allocated to the flow if it is admitted. In addition, $\hat{r}$ denotes the mean traffic rate of all existing flows in the system, which is measured, and $\hat{b}$ the total buffer size allocated to flows in the system.

In (6) to (8), (6) represents the admission criterion for rate or throughput. (7) represents the criterion for delay: the probability that the delay D_n to be experienced by the incoming flow is greater than required delay d_n is less that ϵ_n^d. In (7), the delay function $f_d(y)$, which satisfies $P\{D_n > y\} \leq f_d(y)$, will be given in the next subsection. For (8), b_n is the minimum buffer size with which the required loss probability is met. Specifically,

$$b_n = min\{x : f_l(x) \leq \epsilon_n^l\}, \quad (9)$$

where ϵ_n^l denotes the loss requirement of the incoming flow. The loss function $f_l(x)$ in (9) will also be given later.

If no b_n satisfying (9) exists, the flow is rejected. In addition, if either one of (6) to (8) cannot be satisfied, the flow is rejected. The admission control algorithm rejects a flow by simply dropping packets from the flow, or if there is a best-effort traffic class, the flow is added to this class. It is left as the responsibility of the sender and/or receiver of the flow to re-act to the possible dropping, but how such a reaction is done is out of the scope of this paper.

C. Analysis

This subsection presents analytical support for determining $f_d(y)$ in (7) and $f_l(x)$ in (9).

Note that in the proposed DPS MBAC algorithm, if an incoming flow is admitted, it will be placed at the lowest priority level as compared to all existing flows. As a result, the incoming flow will see an integrated effect from these existing flows. Particularly, we can view the existing flows as an aggregate and equivalently consider a priority server with two inputs: the aggregate and the incoming flow.

For the aggregate, owing to the central limit theorem, when the number of flows in the system becomes large, the aggregate of these flows tends toward Gaussian. In addition, the amount of traffic $\hat{A}(t)$ generated by these flows also tends to a Gaussian process when time t increases. The mean and variance of this Gaussian process can be estimated from measurements [11]. Let $\hat{r}t$ and $\hat{v}(t)$ respectively be the mean and variance of the Gaussian process $\hat{A}(t)$. Then, available results in the literature [8] [16] [1][21] suggest the following approximation for $\hat{Q}(t;r)$, the queue length at time t of a virtual single server queue system with server rate $c > \hat{r}$ fed with the same traffic $\hat{A}(t)$:

$$P\{\hat{Q}(t;c) > x\} \approx exp\left(-inf_{s\geq 0}\frac{(x+(c-\hat{r})s)^2}{2\hat{v}(s)}\right). \quad (10)$$

Simulation results under various cases indicate that the approximation (10) may in fact be a general upper bound for $P\{\hat{Q}(t;c) > x\}$ [1], and under some general conditions, it has been proved in [8] that (10) is an asymptotic upper bound for $P\{\hat{Q}(t;c) > x\}$. Based on these, we re-write (10) as

$$P\{\hat{Q}(t;c) > x\} \lessapprox \hat{f}(x) \equiv exp\left(-\frac{(x+(c-\hat{r})s)^2}{2\hat{v}^*}\right) \quad (11)$$

where $\hat{v}^* \equiv \hat{v}(s^*)$ and s^* is chosen such that $\frac{(x+(c-\hat{r})s)^2}{2\hat{v}(s)}$ reaches its minimum at s^*. Hence, for the aggregate traffic $\hat{A}(t)$ of existing flows in the MBAC system, we have $\hat{A}(t) \sim \langle \hat{f}, c\rangle$, which holds for any $c > \hat{r}$. For ease of explanation and later analysis, we let $c = \hat{r} + (1-u)C$ and consequently,

$$\hat{A}(t) \sim \langle \hat{f}, \hat{r} + (1-u)C\rangle. \quad (12)$$

Here, u can be interpreted as a utilization parameter that can be chosen between the actual utilization $\frac{\hat{r}+r_n}{C}$, which will result from admitting the requesting flow, and the maximum allowed utilization level α. In other words,

$$\frac{\hat{r}+r_n}{C} \leq u \leq \alpha. \quad (13)$$

Now, we consider the service provided by the equivalent two-level priority server to the flow at the low priority level. Specifically, we have the following result.

Proposition 3: Consider a constant rate priority server with two inputs. Suppose the total service rate is C, and the input at the high priority level has $\hat{A}(t) \sim \langle \hat{f}, \hat{r} + (1-u)C \rangle$. Then the server provides to the flow at the low priority level a SSC $(\beta(t), g(x))$ with $\beta(t) = (uC - \hat{r})t; g(x) = \hat{f}(x)$.

Proof: Let $A_n(t)$ and $A_n^*(t)$ respectively be the input and output process of the flow at the low priority level; $A(t)$ and $A^*(t)$ the aggregate input and output process of the system respectively; $\hat{A}^*(t)$ the output process of the flow at the high priority level. Then, by definition,

$$\begin{aligned} A(t) &= A_n(t) + \hat{A}(t) \\ A^*(t) &= A_n^*(t) + \hat{A}^*(t). \end{aligned}$$

Note that the priority system provides to the aggregate of input flows at both high and low priority levels a deterministic service curve Ct, which is, $A^*(t) \geq A(t) \otimes Ct$.

Consider any $0 \leq s_0 \leq t$ when $A(t) \otimes Ct$ reaches minimum, or $A(t) \otimes Ct = A(t-s_0) + Cs_0$. Then, $A^*(t) \geq A(t-s_0) + Cs_0$. Letting $\beta(t) = (\alpha C - \hat{r})t$, we have,

$$\begin{aligned} & A_n \otimes \beta(t) - A_n^*(t) \\ = \; & inf_{0 \leq s \leq t}\{A_n(t-s) + \beta(s)\} - A_n^*(t) \\ \leq \; & A_n(t-s_0) + \beta(s_0) - A_n^*(t) \\ = \; & \alpha C s_0 - \hat{r} s_0 - [A_n^*(t) - A_n(t-s_0)] \\ = \; & \alpha C s_0 - \hat{r} s_0 - [A^*(t) - A(t-s_0) - (\hat{A}^*(t) - \hat{A}(t-s_0))] \\ \leq \; & \hat{A}(t) - \hat{A}(t-s_0) - [\hat{r} + (1-\alpha)C]s_0 \\ \leq \; & sup_{s \leq t}\{\hat{A}(t) - \hat{A}(t-s) - [\hat{r} + (1-\alpha)C]s\} \end{aligned}$$

Here, in the 2nd last step, the fact $\hat{A}^*(t) \leq \hat{A}(t)$ has been used, which tells that the output up to time t cannot exceed the input up to time t for the flow at the high priority level.

By assumption, $\hat{A}(t) \sim \langle \hat{f}, \hat{r} + (1-\alpha)C \rangle$ from which and the definition of gSBB, we get

$$P\left\{sup_{s \leq t}\{\hat{A}(t) - \hat{A}(t-s) - [\hat{r} + (1-\alpha)C]s\} > x\right\} \leq \hat{f}(x).$$

Finally, we get $P\{A_n \otimes \beta(t) - A_n^*(t) > x\} \leq \hat{f}(x)$, with which and SSC definition, the proposition follows. ■

With Proposition 3 and applying its result to Propositions 1 and 2, we obtain:

Corollary 1: Under the same condition as Proposition 3, if the input flow at the low priority level has $A_n(t) \sim \langle f_n, r_n \rangle$, then its backlog $B_n(t)$ and delay $D_n(t)$ in the system satisfies

$$\begin{aligned} P(B_n(t) > x) &\leq f_l(x) \\ P(D_n(t) > y) &\leq f_d(y) \end{aligned}$$

where

$$f_l(x) = f_n \otimes \hat{f}(x), \quad (14)$$

$$f_d(y) = f_n \otimes \hat{f}\left((uC - \hat{r})y\right). \quad (15)$$

In this paper, we have assumed that the aggregate traffic of existing flows in the system is approximated using Gaussian and (11). In addition, we have adopted the implicit traffic descriptor (1) for the incoming flow as used in [22], which implies $A_n \sim \langle 0, \alpha_n C \rangle$. Applying these to (14) and (15), we can further get

$$f_l(x) = exp\left(-\frac{x^2}{2\hat{v}_x^*}\right), \quad (16)$$

$$f_d(y) = exp\left(-\frac{(uC - \hat{r})^2 y^2}{2\hat{v}_y^*}\right), \quad (17)$$

where $\hat{v}_x^* \equiv \hat{v}(s_x^*)$ and s_x^* is chosen such that $\frac{(x+(c-\hat{r})s)^2}{2\hat{v}(s)}$ reaches its minimum at $s = s_x^*$; $\hat{v}_y^* \equiv \hat{v}(s_y^*)$ and s_y^* is chosen such that $\frac{((uC-\hat{r})y+(c-\hat{r})s)^2}{2\hat{v}(s)}$ reaches its minimum at s_y^*.

Remarks: (i) While in Proposition 3 and Corollary 1, we have considered only two priority levels, the result can be easily extended to more levels. In such cases, it is easy to verify from the proof that we can view $\hat{A}(t)$ and $\hat{A}^*(t)$ respectively as the aggregate input and output of all flows that are not at the lowest priority level and consequently get the proposition proved. In addition, for any flow at a certain priority level, we can view $\hat{A}(t)$ and $\hat{A}^*(t)$ respectively as the aggregate input and output of all flows having higher priority than this flow and consequently prove that the server provides to the flow a stochastic service curve as in Proposition 3.

(ii) Note that Proposition 3 and Corollary 1 are general in the sense that they hold as long as the two input flows are gSBB. Since gSBB is built upon queue length distribution, then based on the internal monotonicity property of queue (e.g. see [24]), virtually all stationary input processes can have a steady-state queue length whose tail distribution can be used as the bounding function for gSBB. Hence, although Gaussian approximation has been used in this paper for its popularity and reasonably good performance as investigated by other researchers (e.g. see [8] [17]), the gSBB model allows to use other approximations for existing aggregate traffic instead. In other words, other approximations for existing traffic and other descriptors for the incoming flow can also be used with Proposition 3 and Corollary 1. Under these cases, by applying the corresponding $\hat{f}$ and f_n to Proposition 3 and Corollary 1, the required $f_d(y)$ and $f_l(x)$ can be derived and applied to (7) and (9) for making admission decision.

(iii) Also note that, u can be selected between $\frac{\hat{r}+r_n}{C}$ and α. Roughly, given $\hat{f}(x)$ by (11), a smaller u results in a tighter $f_l(x)$ and a tighter $f_d(y)$. Because of this, by selecting u properly, higher utilization may be achieved based on the analysis, as to be shown in Section V-B.

V. Numerical Results and Discussion

A. Simulation Results

Simulations have been conducted using ns-2 to verify that under DPS, the service received by an existing flow is not adversely affected by flows admitted after it. The simulation settings are similar to those in [6]. In these simulations, there is only one traffic class and the considered link has total capacity of $10Mbps$ shared by flows in the system. All packets are 128 bytes long. Homogeneous exponential on-off sources are used, which have exponentially distributed on and off times. The average on/off time is $325ms$. The transmission rate during on period is $64kbps$, making the mean rate of each flow $32kbps$. Here, the choice of such source characteristics is also motivated by reasons of analytical tractability, since in the next subsection, an investigation on achievable utilization is conducted based on analysis. Simulations were run for 6000 simulation seconds and data were collected after 1500s.

Figs. 2 and 3 show simulation results when the dynamic priority scheduling is implemented. For these figures, each flow in the system is allocated a buffer with the same 160 packets buffer size. Initially, before 1500s, 300 flows, numbered Flow 1 to Flow 300, are generated, whose inter-start times are exponentially distributed with mean $400ms$. Among these flows, Flows 101 to 150 stopped at 2000s; Flows 1 to 50 stopped at 3000s; Flows 201 to 250 stopped at 4000s. In addition, some new flows were added to the system during the simulation period. Particularly, 50 new flows were added in each of the following periods: $[2500s, 3000s]$, $[3000s, 3500s]$ and $[4500s, 5000s]$. The inter-start times of each these 50 new flows have the same exponential distribution with a mean of $400ms$.

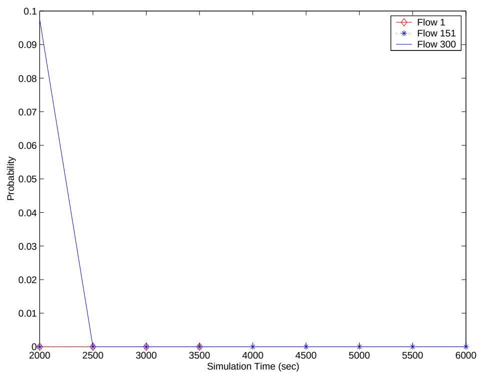

Fig. 2. Average loss of Flow 1, Flow 151 and Flow 300

Fig. 2 illustrates the average loss probabilities respectively experienced by Flow 1, Flow 151 and Flow 300 in each previous 500 simulation seconds period starting from 2000s until 6000s. Fig. 2 shows that no loss has been observed for Flow 1 and Flow 151 during the simulation. This is because, for Flow 1, it is at the highest priority level and not affected by other flows. Consequently, Flow 1 always sees a link with its full capacity available to serve its packets. For Flow 151, the worst case is that Flow 1 to Flow 150 are in the system and Flow 151 gets service only when there is no packet from Flow 1 to Flow 150 in the system. Roughly speaking, Flow 151 sees a system with capacity $C - \sum_{i=1}^{150} r_i$ where C is the total link capacity and r_i is the average rate of flow i. Under the chosen simulation settings, both C and $C - \sum_{i=1}^{150} r_i (\approx 5Mbps)$ are much larger than the average rate of an individual flow, with which and the analysis in the previous section, the expected loss for Flow 1 or Flow 151 was so small that it was not observed during the simulated 6000 seconds.

For Flow 300, Fig. 2 shows that noticeable loss (about 10%) has been experienced during 1500s to 2000s. This is because, during this period, Flow 1 to Flow 299 are in the system and consequently give the residual capacity $C - \sum_{i=1}^{299} r_i \approx 400kbps$ to Flow 300. Based on Proposition 3 and Corollary 1, the average loss of Flow 300 can be much higher than that of Flow 1 or Flow 151. In particular, letting $u = \frac{\sum_{i=1}^{300} r_i}{C} = 96\%$, we can get analytically from (16) that the loss probability for Flow 1 during 1500s to 2000s is bounded by 13.67%. When simulation time proceeds, Fig. 2 shows the loss experienced by Flow 300 became so small that no loss could be found after 2000 sec. This is because Flows 101 to 150 left the system at 2000s, which were initially given higher priorities than Flow 151. Although some new flows were added to the system during the rest of simulation time, they were given lower priorities than Flow 151 based on the DPS algorithm. As a result, the worst case for Flow 151 after 2000s is that it sees a residual capacity $C - \sum_{i=1}^{249} r_i (\approx 2Mbps)$. Consequently, a loss less than 10^{-10} can be expected from (16).

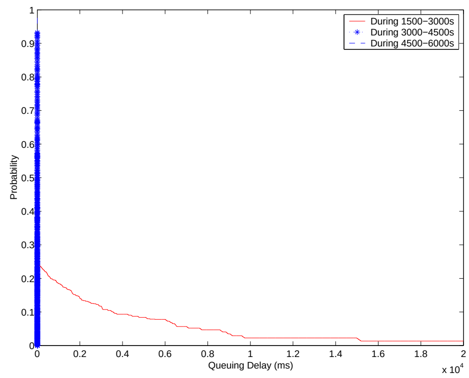

Fig. 3. Delay distribution of Flow 300

Fig. 2 has shown that the loss behavior of an admitted flow becomes better when some flows admitted earlier leave the system and is not affected by newly admitted flows. To verify the same trend for delay, Fig. 3 is presented. This figure shows the tail delay distribution experienced by Flow 300 during 1500s to 3000s, during 3000s to 4500s, and during 4500s to 6000s. It can be observed from the figure that during 1500s to 3000s, more than 20% of the packets have to wait in queue for some time before being transmitted. However, during 3000s to 4500s, a Flow 300 packet almost needs not wait: upon arrival, it is immediately served. This is because some flows, which are Flow 1 to 50 and Flow 101 to 150 initially at higher priorities than Flow 300, stopped before 3000s, giving more residual capacity to serve Flow 300 as explained above for loss. Although some new flows were added to the system before 4500s or during 4500s to 6000s, making the total utilization the same after 5000s as before 2000s, Fig. 3 shows that they do not affect the delay performance of Flow 300 during 4500s to 6000s. Indeed, during 4500s to 6000s, a queueing delay of 0 was always found for Flow 300.

B. Achievable Utilization

This subsection investigates the achievable utilization by the proposed DPS MBAC. For ease of exposition, we again assume sources are homogeneous and each source is exponential on-off [2]. In addition, the buffer size is also set to be 160 Kb. The choice of exponential on-off source is motivated by analytical tractability and comparability. For such a source i, the mean and variance of the arrival process $A_i(t)$ are

$$m_i t = \frac{\lambda_i h_i}{\lambda_i + \beta_i} t$$
$$v_i(t) = \frac{2\lambda_i \beta_i h_i^2}{(\lambda_i + \beta_i)^3} \left[t - \frac{1}{\lambda_i + \beta_i} (1 - e^{-(\lambda_i + \beta_i)t}) \right]$$

where β_i^{-1} denotes the mean length of the on period, λ_i^{-1} the mean length of the off period and h_i the sending rate during the on period. Then, for the Gaussian approximation, its mean and variance are respectively $\hat{r}t = Nm_i t$ and $\hat{v}(t) = Nv_i(t)$, where N is the number of flows in the system.

In Fig. 4, the maximum utilization against link capacity for the DPS-based algorithm is plotted. In addition, for comparison, we have also plotted the maximum utilization curve based on the analysis of bufferless multiplexing, which has been used in [22] to support flow-aware networking and per-flow MBAC. For this figure, the following service requirements, as adopted in [5][22], have been used. For the DPS-based aggregation algorithm, the objective is a 10^{-6} or less probability of exceeding the delay target of $50ms$. For bufferless multiplexing, 10^{-6} is the targeted rate overload probability. Also for Fig. 4, we have required that the maximum allowed utilization level α in (6) be $\alpha = 95\%$. For the DPS-based aggregation curve (1), we have chosen $u = \alpha$ for (16) and (17), while for the DPS-based aggregation curve (2), u is set to be $min\{95\%, \frac{\hat{r}}{C}+5\%\}$.

Fig. 4 shows that with the DPS-based algorithm, excellent (above 90%) system utilization can be achieved when the capacity is higher than $100Mbps$. Even when the capacity is low, the achievable utilization can still be reasonably good (e.g. 70% when the capacity is $10Mbps$). In addition, Fig. 4 shows that, as discussed earlier, the selection of u has great effect on the achievable utilization. Particularly, in this case, when C is smaller than $100Mbps$, choosing $u = min\{95\%, \frac{\hat{r}}{C} + 5\%\}$ can achieve much higher utilization than choosing $u = 95\%$.

Note that bufferless multiplexing approach cannot be used to estimate delay guarantee [5]. For the bufferless multiplexing curve shown in Fig. 4, the corresponding objective is to achieve 10^{-6} (or less) rate overload probability used to approximate the loss. If both loss and delay are considered for admission control, the bufferless multiplexing curve may be considered as a lower bound on the achievable utilization.

For the loss performance of the DPS-based algorithm, Fig. 5 is presented, which shows the loss curve determined from (16). In this figure, the two choices of u as for Fig. 4 are also considered. Based on the analysis, by combining both, we can see from the figure that under most capacity settings, 10^{-6} (or less) loss probability is provided when buffer size is set to $160Kb$. As expected from (16), lower loss can be achieved

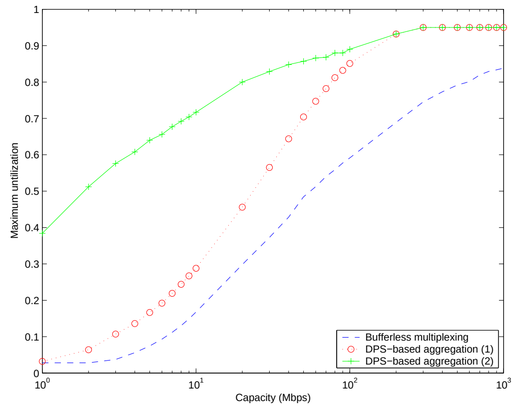

Fig. 4. Achievable utilization

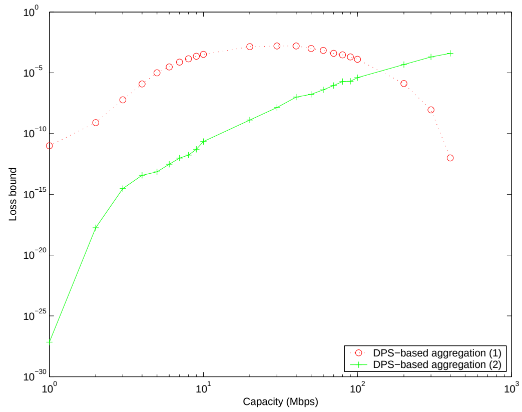

Fig. 5. Loss bound

by allocating more buffer size. In this case, a buffer size of $400Kb$ or more can ensure that 10^{-6} loss is achieved for the capacity range in Figs. 4 and 5.

C. Discussion

The simulation results verify that once a flow is admitted, with DPS, the service guarantees provided to this flow will not be adversely affected by other flows admitted after it. This implies that the admission decision needs not re-check whether the SSGs of an existing flow will be violated after a new flow is admitted. Note that such re-checking usually involves additional analysis and/or requirements/constraints on flows. For example, when FIFO is used in aggregate MBAC algorithms, when a new flow requires more stringent service guarantees than existing flows, such re-checking and possible resource re-allocation are critical to ensure the service requirements of all flows will be met.

Another example is that fair queueing (FQ) may be used to aggregate flows and perform per-flow MBAC [22]. If loss and/or delay guarantees are required in addition to throughput guarantee, the FQ-based MBAC may also need to re-check every existing flow if its loss and/or delay guarantees will be violated and if so, how much additional resource (e.g. buffer) needs to be added to it. This is because with FQ, when a new flow is admitted, existing flows will have to share the server proportionally based on the share weights in FQ with the new flow. As a result, the guaranteed rate to each existing flow is reduced after the admission of the new flow, which could affect the service guarantees received by the existing flow.

Note that various existing MBAC algorithms such as [12], [13] and [14] do not need to re-check admissition condition for existing flows either. These algorithms usually rely on the concept of effective bandwidth or equivalent capacity to determine the admissible region, based on the assumption that flows are homogeneous and require the same SSGs. The DPS MBAC differs from them in that DPS MBAC may be used to support flows with diverse service requirements.

As discussed in the previous section, Proposition 3 and Corollary 1 are general. Although we have used Gaussian to approximate existing traffic, other approximations may also be used and by applying their gSBB representation, Proposition 3 and Corollary 1 can be readily used to derive the corresponding delay and loss statistical service guarantees.

Finally, the implementation of DPS MBAC is simple in the sense that the computational complexity of DPS is $O(1)$. Although classification of flows and maintenance of the list of flows in the system appear to be complex, recent advances in hardware technology have made these feasible even at line rates up to $OC-192$ $(10Gbps)$ [7].

VI. Conclusion and Future Work

We proposed a simple per-flow MBAC algorithm for providing SSGs in a flow-aware network. In the proposed DPS MBAC, a newly admitted flow is always given lower priority than existing flows; admission decision is based on measurements of existing flows together with SSG requirements of the incoming flow. With DPS MBAC, the experienced SSGs of an admitted flow will not be adversely affected by flows admitted after it. Analysis and simulation results supporting the proposed MBAC algorithm have been provided. These results also show that the experienced SSGs of an admitted flow become better and better due to some earlier admitted flows leave the system. An implication is that when admission control is performed, there is no need to re-check the SSGs provided to existing flows nor to re-allocate resources to the existing flows to maintain their service guarantees.

This work leaves many issues for further study. One is to limit the maximum buffer size to provide bounded maximum delay, since the latter is closely related to the former. Another is the selection of the utilization parameter u in (13). As discussed in the paper, different u may result in different achievable utilization level. Work may be conducted to find some adaptive method for selecting u. In addition, it would be interesting to investigate how to set the time-out time and what its effect on MBAC performance could be. Moreover, to make the analysis and comparison tractable, homogeneous on-off sources have been used. Future work is needed to conduct simulations and investigate systems with heterogeneous sources and diverse service requirements.

Acknowledgment

The “Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence” is appointed by The Research Council of Norway and funded by the Research Council, NTNU and UNINETT. (http://www.ntnu.no/Q2S/)

The authors would like to thank Yong Liu for his help in getting the simulation results.

References

[1] R. Addie, P. Mannersalo, and I. Norros. Most probable paths and performance formulae for buffers with Gaussian input traffic. *European Trans. Telecomm.*, 13(3):183–196, 2002.

[2] D. Anick, D. Mitra, and M. M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Tech. J.*, 61(8):1871–1894, 1982.

[3] F. Baccelli, G. Cohen, G. J. Olsder, and J. Quadrat. *Synchronization and Linearity: An Algebra for Discrete Event System*. John Wiley and Sons, 1992.

[4] S. Blake and et al. Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers. *IETF RFC 2474*, Dec. 1998.

[5] T. Bonald, S. Oueslati-Boulahia, and J. Roberts. IP traffic and QoS control: the need for a flow-aware architecture. In *World Telecommunication Congress*, 2002.

[6] L. Breslau, S. Jamin, and S. Shenker. Comments on the performance of measurement-based admission control algorithms. In *IEEE INFOCOM*, 2000.

[7] CASPIAN NETWORKS. *Flow-state routing: rationale and benefits (White Paper)*. 2004.

[8] J. Choe and N. B. Shroff. A central-limit-theorem-based approach for analyzing queue behavior in high-speed networks. *IEEE/ACM Trans. Networking*, 6(5):659–671, Oct. 1998.

[9] R. L. Cruz. A calculus for network delay, part I: network elements in isolation. *IEEE Trans. Information Theory*, 37(1):114–131, Jan. 1991.

[10] R. L. Cruz. Quality of service management in integrated services networks. In *1st Semi-Annual Research Review, CWC, USCD*, 1996.

[11] D. Y. Eun and N. B. Shroff. A measurement-analytic approach for QoS estimation in a network based on the dominant time scale. *IEEE/ACM Trans. Networking*, 11(2):222–235, Apr. 2003.

[12] S. Floyd. Comments on measurement-based admission control for controlled-load services. Technical report, Lawrence Berkeley National Laboratory, 1996.

[13] R. J. Gibbens and F. P. Kelly. Measurement-based connection admission control. In *Proc. the 15th International Teletraffic Congress (ITC)*, 1997.

[14] M. Grossglauser and D. N. Tse. A framework for robust measurement-based admission control. *IEEE/ACM Trans. Networking*, 7(3):293–309, June 1999.

[15] Y. Jiang, P. J. Emstad, V. Nicola, and A. Nevin. Measurement-based admission control: A revisit. In *17th Nordic Teletraffic Seminar*, 2004.

[16] H. S. Kim and N. B. Shroff. Loss probability calculations and asymptotic analysis for finite buffer multiplexers. *IEEE/ACM Trans. Networking*, 9(6):755–768, Dec. 2001.

[17] E. W. Knightly and N. B. Shroff. Admission control for statistical QoS: Theory and practice. *IEEE Network*, pages 20–29, March/April 1999.

[18] J.-Y. Le Boudec and P. Thiran. *Network Calculus: A Theory of Deterministic Queueing Systems for the Internet*. Springer-Verlag, 2001.

[19] Y. Liu, C.-K. Tham, and Y. Jiang. A stochastic network calculus. Technical report, National University of Singapore, 2003.

[20] Y. Liu, C.-K. Tham, and Y. Jiang. Conformance analysis in networks with service level agreements. *Computer Networks*, 2005. in press.

[21] P. Mannersalo and I. Norros. A most probable path approach to queueing systems with general Gaussian input. *COMNET*, 40(3):399–412, 2002.

[22] J. W. Roberts. Internet Traffic, QoS and Pricing. *Proceedings of the IEEE*, Sept. 2004.

[23] K. Shiomoto, N. Yamanaka, and T. Takahashi. Overview of measurement-based connection admission control methods in ATM networks. *IEEE Communications Surveys*, First Quarter 1999.

[24] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Models*. John Wiley & Sons, 1983.

[25] Q. Yin, Y. Jiang, S. Jiang, and P. Y. Kong. Analysis on generalized stochastically bounded bursty traffic for communication networks. In *Proc. IEEE LCN'2002*, 2002.