Ensuring IP Services Consistency through Lightweight Monitoring-based Admission Control

Solange Rito Lima, Paulo Carvalho, Vasco Freitas
University of Minho, Dept. of Informatics, 4710-057 Braga, Portugal
{solange,pmc,vf}@di.uminho.pt

Abstract— **This paper evaluates the performance of a distributed and lightweight AC model based on per-class edge-to-edge monitoring feedback for ensuring the quality of multiple services in class-based IP networks. The model resorts to service-dependent AC rules for controlling QoS parameters and SLSs utilization, both intradomain and end-to-end. To provide a proof-of-concept of the proposed AC solution, a prototype of the AC model has been developed and tested using a simulation platform. The devised test scenarios aim at exploring the AC criteria's ability in satisfying each service class QoS levels and existing SLSs commitments. Generically, the results show that the proposed AC model, using a two-rule AC criterion defined on a service class basis, is able to control service levels and achieve high network utilization, without adding significant complexity to the network elements. The use of systematic edge-to-edge on-line monitoring and of a controlled degree of overprovisioning proved to be essential design aspects contributing for reaching a good compromise between simplicity and performance.**

I. INTRODUCTION

The road toward a consistent service integration in the Internet has been motivating the evolution of the underlying protocol stack. The research community has been making strong efforts to endow the TCP/IP stack with new service models, enhanced protocols and mechanisms to allow such integration with enhanced quality and performance. Class of Service (CoS) networks, where flows with similar characteristics and service requirements are aggregated in the same class (e.g. Diffserv [1]), are a step forward in pursuing scalable QoS solutions. Service-oriented traffic control mechanisms, operating with minimum impact on network performance, assume a crucial role to control services' quality and network resources transparent and efficiently. Within traffic control mechanisms, Admission Control (AC) is recommended for keeping service classes under controlled load and assuring the required QoS levels [2], [3]. In fact, AC allows preventing instability and congestion and assuring QoS and Service Level Specifications (SLSs) fulfillment. Although AC has been extensively studied in the literature, few studies deal with the simultaneous management of domain QoS levels and interdomain SLSs, falling short in establishing and formalizing concrete and flexible AC equations to be applied to multiservice and multidomain networks.

Considering the above reasoning, a new AC model has been proposed in [4], [5] for controlling services quality in multiclass IP networks. An important underlying idea driving the model operation is to take advantage of the consensual need for on-line service monitoring and for traffic control at network domain boundaries, using the resulting monitoring information to perform distributed AC. In the proposed AC model, AC decisions are driven by feedback from systematic edge-to-edge measurements of relevant QoS parameters for each service type and SLSs utilization. While ingress nodes perform implicit or explicit AC, resorting to service-dependent rules for QoS and SLS control, egress nodes collect service metrics providing them as inputs for AC. To improve the trade-off between complexity and QoS assurance, the AC criteria comprise service-dependent degrees of overprovisioning in order to simplify AC while improving QoS guarantees.

The proposed AC model being distributed, service-oriented, based on per-class on-line monitoring and involving only edge nodes should be able to abstract from network core complexity and heterogeneity, to sense each service classes' dynamics and to perform AC with reduced state information, latency and overhead. These characteristics are expected to contribute to pursue important design goals such as simplicity, scalability and flexibility. However, a fundamental question raising from the model properties is the following: *Will service-dependent AC rules driven by edge-to-edge on-line monitoring be able to control distinct QoS guarantees and SLSs commitments properly?* This paper, extending the previous work, intends to answer this question, providing a proof-of-concept of the proposed solution, illustrating its self-adaptive ability in controlling QoS and SLSs in a multiclass domain. In this way, the performance evaluation of the proposed AC model carried out in this paper aims at assessing its effectiveness and efficiency in satisfying each service class QoS levels and existing SLSs commitments. The model's architecture is here introduced and its implementation issues, regarding the services to be supported and related monitoring decisions, are also debated.

The remaining of this paper is structured as follows: related work is discussed in Section II; the multiservice AC model architecture is presented in Section III; the AC criteria description and specification is summarized in Section IV; the simulation prototype and implementation details of the proposed AC model covering a multiclass domain is described in Section V; the proof-of-concept, involving the performance evaluation of the AC model, is provided in Section VI; finally, the conclusions are drawn in Section VII.

II. RELATED WORK

Common AC approaches for CoS IP networks are either centralized (e.g. based on bandwidth brokers [6], [7]) or distributed, parameter or measurement-based differing on the type of services being supported. To provide guaranteed services (e.g., for hard real-time traffic), AC proposals tend to require significant network state information and, in many cases, changes in all network nodes [6], [8], [9]. To provide predictive services (e.g., for soft real-time traffic) measurement-based AC (MBAC) [10], [11] and end-to-end MBAC (EMBAC) solutions [12]–[14] have deserved special attention. These solutions leads to reduced control information and overhead, but eventually to QoS degradation. To control elastic traffic, for more efficient network utilization, implicit AC strategies, i.e., without requiring explicit signaling between the application and the network, have also been defined [15], [16].

Despite the wide range of AC approaches proposed in the literature, aspects such as: (i) the trade-off between service assurance level and network control complexity; (ii) the flexible support of distinct service types; (iii) the simultaneous control of QoS levels and existing SLSs; and (iv) the intradomain and end-to-end AC operation is not covered or balanced as a whole. Therefore, handling AC in multiservice class-based networks is still an open research topic and it is within this context that the motivation for the present work lays on.

III. AC MODEL ARCHITECTURE

In the AC model, admission decisions are made taking into account both the levels of QoS being offered for each service type and the corresponding SLSs utilization. Therefore, AC is performed resorting to QoS and SLS control equations, specifically defined according to each service characteristics. In this context, the model architecture strongly lays on service definition, QoS/SLS monitoring and CoS traffic characterization to sustain the definition and operation of the AC decision criteria, interrelated as shown in Figure 1.

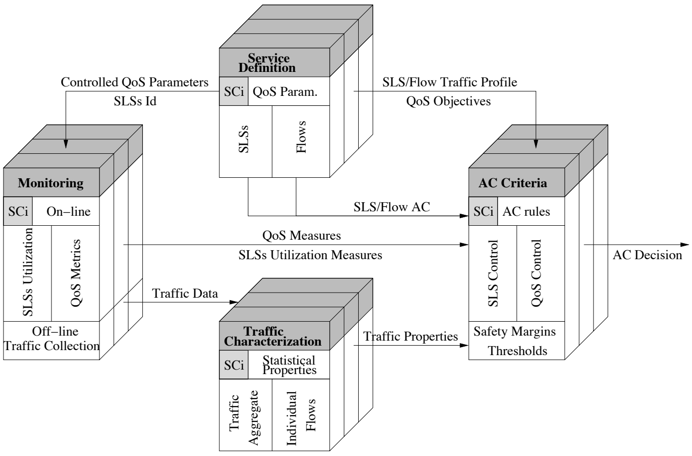

Fig. 1. AC model architecture

Service definition, involves the definition of basic services oriented to different application requirements, the definition of relevant QoS parameters to control within each service type and the definition of SLSs' syntax and semantics. Through systematic edge-to-edge measures of QoS parameters and SLSs utilization, *on-line monitoring* keeps track of QoS and SLS status in the domain through well-defined metrics, providing feedback to drive AC decisions. As an *off-line monitoring* process, CoS traffic aggregates may also be collected for subsequent off-line analysis and characterization. This analysis allows to determine the statistical properties of each class as a result of traffic aggregation so that more realistic service-oriented AC rules, thresholds and safety margins can be established. The knowledge resulting from interrelating these areas and from comparing existing measurement-based or hybrid AC algorithms provides the basics for defining a multiservice *AC decision criteria.*

IV. AC MODEL SPECIFICATION

Before providing a brief overview of the multiservice AC model, few remarks are made in order to clarify the description. In a transit domain, the way an SLS is viewed varies according to whether a client or service provider perspective is taken. For upstream SLSs, the domain acts as a service provider to the previous domain; for downstream SLSs the domain acts as a client of the next domain. A set of upstreams SLSs with identical requirements share a service class in the domain.

A. Generic model operation

Apart from the usual classification and traffic conditioning functions present in CoS networks, ingress nodes perform explicit or implicit AC depending on the application type and corresponding traffic class. Egress nodes perform on-line QoS monitoring and SLS control. The *Ingress-Egress QoS Monitoring* task measures relevant parameters for each service (service metrics) using appropriate time scales and methodologies. The resulting measures are expected to reflect the service available from each ingress node, and will be used by a QoS Control rule to drive AC decisions. The *SLS Control* task monitors the usage of downstream SLSs at each egress, to ensure that traffic to other domains does not exceed the negotiated profiles and packet drop will not occur due to a simple and indiscriminate TC process. An SLS Rate Control rule checks if the SLS can accommodate the traffic profile of the new flow, complementing the AC decision process.

QoS monitoring statistics and SLS utilization are then sent to the corresponding ingress routers to update an ingress-egress service matrix used for distributed AC and active service management. This notification may be carried out periodically, when a metric value or its variation exceeds a limit, or the SLS utilization exceeds a safety threshold.

B. AC Criteria Specification

As mentioned, the service-dependent AC criteria resort to (1) rate-based SLS control rules and (2) QoS parameters control rules. These rules follow the notation introduced in [4], where domain entities such as (i) service classes; (ii) upstream SLSs; (iii) downstream SLSs and (iv) traffic flows are defined. In brief, the model considers a multiclass domain D_x comprising N ingress nodes and M egress nodes, where a set of service classes SC_i is defined.

1) Rate-based SLS Control Rules: In a domain D_x, for each ingress node I_n and each egress node E_m one or more SLSs can be in place. As each upstream SLS_{i,I_n} and downstream SLS^{+}_{i,E_m} have specified a negotiated rate, R_{i,I_n} and R^{+}_{i,E_m} respectively, a rate-based Measure-Sum algorithm can be applied to control SLSs utilization at each network edge node.

For a service class SC_i under *explicit AC*, verifying if a new flow $F_j \in SLS_{i,I_n}$ can be admitted at each ingress node I_n involves testing if the SLS_{i,I_n} can accommodate the new flow traffic profile, i.e.

$$\tilde{R}_{i,(I_n,*)} + r_j \leq \beta_{i,I_n} R_{i,I_n}. \quad (1)$$

In (1), $\tilde{R}_{i,(I_n,*)}$ is the current measured rate of flows using SLS_{i,I_n} independently of the E_m nodes involved; r_j is the rate of the new flow F_j; $0 < \beta_{i,I_n} \leq 1$ is a safety margin defined for the negotiated rate R_{i,I_n}. When the destination of flow F_j is outside D_x, verifying if the new flow can be admitted involves also testing if the downstream SLS^{+}_{i,E_m} can accommodate the new flow traffic profile, i.e.

$$\tilde{R}^{+}_{i,(*,E_m)} + r_j \leq \beta^{+}_{i,E_m} R^{+}_{i,E_m}. \quad (2)$$

In (2), $\tilde{R}^{+}_{i,(*,E_m)}$ is the current measured rate of flows using SLS^{+}_{i,E_m}, considering all ingress-to-E_m estimated rates of flows going through E_m, i.e. $\tilde{R}^{+}_{i,(*,Em)} = \sum_{k=1}^{N} \tilde{r}_{i,(I_k,E_m)}$; r_j is the rate of the new flow F_j; $0 < \beta_{i,E_m} \leq 1$ is the safety margin for the rate R^{+}_{i,E_m} defined in SLS^{+}_{i,E_m}. This safety margin determines the degree of overprovisioning for SC_i.

For a service class SC_i under *implicit AC*, as flows are unable to describe r_j, the SLS control equations above become similar to the QoS control equation (3), considering $P_{i,p}$ as a rate-based parameter. Therefore, traffic flows are accepted or rejected implicitly according to the value of a variable $AC\_Status_{\Delta t_i}$ computed once for the measurement interval Δt_i.

2) QoS Parameters Control Rules: At each ingress node I_n, the $AC\_Status_{\Delta t_i}$ variable, used to control the admission of new flows in the measurement interval Δt_i, is updated after checking the controlled parameters $P_{i,p}$ of each service class SC_i, measured and provided by egress nodes, against the corresponding pre-defined parameter thresholds $T_{i,p}$, i.e.

$$\forall (P_{i,p}, \beta_{i,p}) \in P_{SC_i} : \tilde{P}_{i,p} \leq T_{i,p}, \quad (3)$$

where $\tilde{P}_{i,p}$ is the ingress-to-egress measured parameter, $\beta_{i,p}$ is the corresponding safety margin, and $T_{i,p}$ is the parameter's upper bound or threshold, given by $T_{i,p} = \beta_{i,p} P_{i,p}$, used to trigger AC. Equation (3) is not flow dependent, i.e. it is checked once during Δt_i to determine $AC\_Status_{\Delta t_i}$. The $AC\_Status_{\Delta t_i}$ `accept` indicates that the measured QoS levels for SC_i are in conformance with the QoS objectives and, therefore, new flows can be accepted. The $AC\_Status_{\Delta t_i}$ `reject` indicates that no more flows should be accepted until the class recovers and restores the QoS target values. This will only be checked at Δt_{i+1}.

3) End-to-end AC rules: The end-to-end operation is viewed as a repetitive process of AC at each domain ingress nodes and cumulative computation of the service metrics available at each domain. Assuming a consistent mapping between the service classes in upstream and downstream domains, when AC is taken from an end-to-end perspective making an AC decision at ingress node I_n of D_x involves considering the following rule:

$$\forall P_{j,p} \in P_{F_j} : (\mathtt{op}_1\ (P^{acc^-}_{j,p}, P_{i,p}))\ \mathtt{op}_2\ (\gamma_{j,p} P_{j,p}), \quad (4)$$

where each flow requested QoS parameter $P_{j,p}$, allowing a tolerance factor $\gamma_{j,p}$, is checked against the cumulative value computed for the parameter ($P^{acc^-}_{j,p}$) when crossing previous domains affected by the corresponding target value of $P_{i,p}$[1]. It should be noticed that a cumulative process for end-to-end QoS computation is consistent with the cascade approach for the support of interoperator IP-based services, which has the merit of being more realistic, i.e., in conformance with the Internet structure and operation, and more scalable than the source-based approach [17].

Depending on each parameter semantics, $\mathtt{op}_1$ and $\mathtt{op}_2$ may express different operations, i.e., $\mathtt{op}_1 \in \{add \mid sub \mid max \mid min \mid mul \mid f^{spec}\}$ and $\mathtt{op}_2 \in \{\leq \mid < \mid \geq \mid > \mid =\}$. If the flow can be accepted in D_x, the new available service computation to be included in the flow request is given by $P^{acc}_{j,p} = \mathtt{op}_1\ (P^{acc^-}_{j,p}, P_{i,p})$.

V. MODEL IMPLEMENTATION

Considering the architecture presented in Section III, the AC model implementation involves the definition of a limited set of service classes with distinct QoS requirements and the configuration of corresponding traffic handling policies. To evaluate the AC model's ability to meet service commitments efficiently in a multiclass Diffserv domain, a simulation prototype was devised and set up based on the Network Simulator (NS-2).

A. Definition of service classes

Taking into consideration current service configuration guidelines [18], three initial service classes are defined. As basic policy, TCP and UDP traffic are treated separately; UDP traffic is further divided according to its QoS requirements.

Service Class 1 (SC1), oriented to conversational services, provides a high quality service guarantee and is supported by the Expedited Forwarding (EF) Per-Hop Behavior (PHB). This class may comprise traffic with hard real-time constraints such as VoIP or circuit emulation over IP [18]. Service Class 2 (SC2), oriented to a range of multimedia streaming services with soft real-time constraints, provides a predictive service with low delay, low loss and minimum bandwidth guarantee and is supported by the Assured Forwarding (AF) PHB. This

---

[1] Note that $\tilde{P}_{i,p}$ could also be used in the cumulative process of metric computation instead of $P_{i,p}$. This option would lead to less stable AC decisions and end-to-end available service computation. Despite being more conservative, taking $P_{i,p}$, i.e., the class parameter target value, allows more robust, stable and reliable AC decisions.

class may comprise audio and video streaming or webcasting [18]. In SC1 and SC2 the AC criteria, apart from considering the flow traffic profile description (explicit AC), take into account measures of network loss and delay of those classes attending to their relevance for the supported conversational and streaming applications. Service Class 3 (SC3) oriented to elastic data applications, generically, supports TCP adaptive traffic. Depending on the nature of TCP flows (e.g., high throughput vs. undifferentiated traffic), this class can be implemented using the AF or Default Forwarding PHB. For service SC3, the AC criterion will be implicit and relaxed. Giving the nature of TCP traffic and associated congestion control mechanism, IP packet loss will be considered the parameter under control.

B. Monitoring Decisions

In this study, the objective of on-line monitoring is twofold. First, it provides inputs for the AC decision module and corresponding MBAC algorithms which, apart from flow traffic profile and/or QoS requirements information, require a realistic view of the service classes' status and performance. Second, it allows SLS and QoS auditing in the domain.

1) Controlled QoS and SLS metrics: At each egress monitoring module, the controlled QoS parameters for SC1 are IP Transfer Delay (IPTD) (similar to One-Way Delay (OWD)), IP Delay Variation (ipdv) and IP Loss Ratio (IPLR) (similar to One-Way Packet Loss (OWPL)). These one-way QoS parameters are considered the most critical for the real-time services supported in this class. AC for SC2, being more flexible, takes the flow mean rate (instead of peak rate) for SLS control and IPTD and IPLR for QoS control. For SC3, oriented to TCP traffic, AC is implicit and decisions are based essentially on IPLR control. This QoS parameter influences the goodput of TCP sessions, which is a common measure of the quality of TCP based applications.

Considering [19], [20] inputs, the corresponding metrics are defined in Table I for a measurement time interval Δt_i. The mean value of each metric in Δt_i, measured for an ingress-egress (I_n, E_m) pair and service class SC_i, is controlled by the AC module as described in IV-B. The estimation of SLS^+_{i,E_m} usage is not ingress dependent, therefore it is not controlled necessarily on an ingress-egress basis. For each class, the metrics in Table I are estimated and controlled resorting to passive and active measurements. Comparing the outcome of both approaches allows to assess and tune the probing process as discussed in [21].

C. AC criteria parameterization

Table II illustrates the main parameters used to configure the AC rules used for controlling both SLS utilization and domain QoS levels. Three downstream SLSs have been considered, one per service class, with a negotiated rate (R^+_{i,E_m}) defined according to the traffic load share intended for the corresponding class in the domain. The Measure-Sum algorithm, which rules SLS utilization, has specific utilization target (β^+_{i,E_m}) values depending on how conservative the AC decisions must be. For instance, a $\beta^+_{i,E_m} = 0.85$ corresponds to impose a safety margin of 15% to absorb load fluctuations and optimistic measures. This value can be viewed as a degree of overprovisioning. The AC thresholds $T_{i,p}$ which rule the control of each class QoS levels in the domain are set taking into account the domain topology dimensioning, queuing and propagation delays, and perceived QoS upper bounds for common applications and services. As shown in Table II, the parameterization of the AC rules is service-dependent and larger β^+_{i,E_m} and tighter $T_{i,p}$ are defined for more demanding classes.

TABLE I

CONTROLLED QoS AND SLS METRICS

Rate Parameters	
Rate (bps)	$r_{i,\Delta t_i} = (\sum bits\_recv_i)_{\Delta t_i} / \Delta t_i$
Utilization	$U_{i,\Delta t_i} = r_{i,\Delta t_i} / C$
Delay Parameters	
IP Transfer Delay	$IPTD_{i,pkt} = t_{E_m,pkt} - t_{I_n,pkt}$
Mean IPTD	$\overline{IPTD}_{i,\Delta t_i} = (\sum IPTD_{i,pkt} / \sum pkts\_recv_i)_{\Delta t_i}$
Maximum IPTD	$IPTD^{max}_{i,\Delta t_i} = max(IPTD_{i,pkt})_{\Delta t_i}$
Minimum IPTD	$IPTD^{min}_{i,\Delta t_i} = min(IPTD_{i,pkt})_{\Delta t_i}$
Delay Var.	$ipdv_{i,2pkt} = IPTD_{i,pkt} - IPTD_{i,pkt-1}$
Mean ipdv	$\overline{ipdv}_{i,\Delta t_i} = (\sum \lvert ipdv_{i,2pkt} \rvert / \sum pkts\_recv_i)_{\Delta t_i}$
Signed mean ipdv	$\overline{ipdv^s}_{i,\Delta t_i} = (\sum ipdv_{i,2pkt} / \sum pkts\_recv_i)_{\Delta t_i}$
Maximum ipdv	$ipdv^{max}_{i,\Delta t_i} = max(ipdv_{i,2pkt})_{\Delta t_i}$
Minimum ipdv	$ipdv^{min}_{i,\Delta t_i} = min(ipdv_{i,2pkt})_{\Delta t_i}$
Loss parameters	
IP Loss Ratio	$IPLR_{i,tot} = tot\_pkts\_lost_i / tot\_pkts\_sent_i$
IPLR in Δt_i	$IPLR_{i,\Delta t_i} = (\sum pkts\_lost_i / \sum pkts\_sent_i)_{\Delta t_i}$

TABLE II

SERVICE PARAMETER CONFIGURATION

SC i	SLS Rate R^+_{i,E_m} (share)	S.M. β^+_{i,E_m}	QoS Parameter $P_{i,p}$	Threshold $T_{i,p}$
1	3.4Mbps (10%)	0.85	IPTD,ipdv,IPLR	35ms,1ms,10^{-4}
2	17.0Mbps (50%)	0.90	IPTD,IPLR	50ms,10^{-3}
3	13.6Mbps (40%)	1.0	IPLR	10^{-1}

D. Simulation topology

The simulation topology is illustrated in Figure 2. The network domain consists of ingress routers I_1, I_2, a multiclass network core router C_1 and an edge router E_1. The service classes SC1, SC2 and SC3 are implemented in all the domain nodes. While I_1 multiplexes three types of sources, each type mapped to a different class, I_2 is used to inject concurrent or cross traffic (referred as CT-I2). This allows to evaluate concurrency effects on distributed AC and assess the impact of cross traffic on the AC model performance. Despite its simplicity, this topology allows to emulate a wide range of test scenarios, including relevant aspects of real environments. For instance, the scenarios with cross traffic allow to contemplate the presence of unmeasured traffic within the core, having an impact on the domain's QoS and load but without being explicitly measured by E_1 SLS rate control rules.

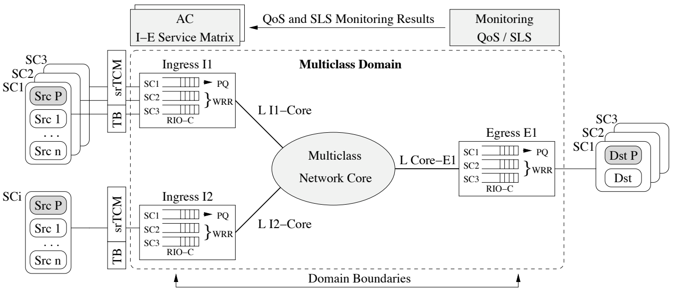

Fig. 2. Simulation topology

The domain routers implement the service classes according to a hybrid Priority Queuing - Weighted Round Robin (PQ-WRR(2,1)) scheduling discipline, with RIO-C as AQM mechanism. The PQ-WRR(2,1) discipline applies to the highest priority class (SC1) a strict priority treatment with a tight limit on a pre-defined rate (10% of the link capacity), whereas the queues of the remaining class (SC2, SC3) are served with a 2 to 1 proportionality. Each class queue is 150 packets long. The domain internodal links capacity is 34Mbps, with a 15ms propagation delay. $L_{C1,E1}$ link works as a bottleneck in this network topology. At network entrance, SC1 is policed and marked using a Token Bucket (TB) which controls both rate and burst size, whereas SC2 and SC3 are policed and marked using a single-rate Three-Color Marker (srTCM).

1) Source models: Generically, three source models have been considered: Constant Bit Rate (CBR) sources, Exponential on-off ($EXPOO$) and Pareto on-off (PAR) sources. PAR sources with $1 < \alpha < 2$ under aggregation will allow to generate traffic exhibiting long-range dependence. SC1 comprises low rate UDP traffic sources (64kbps) with on/off periods of 0.96ms/1.69ms and small packet sizes (120bytes), reflecting voice-like traffic; SC2 also comprises UDP traffic with higher peak rates (256kbps), on/off periods of 500ms/500ms and larger packet sizes (512bytes), as generated by other real-time applications with higher variability. SC3 comprises long-lived high throughput TCP traffic, resulting from an FTP application generating packets of 512bytes. The flow arrival process is Poisson with exponentially distributed interarrival (0.4-2s) and holding times (90, 120, 180s for SC1, SC2 and SC3, respectively).

VI. PERFORMANCE EVALUATION OF THE MODEL

The performance analysis of the AC criteria involves: (i) verifying if QoS parameters are in conformance with the established QoS levels; (ii) quantifying QoS violations, at class and packet level; (iii) evaluating each class blocking probabilities; (iv) measuring the utilization level of each class individually and of the network domain globally, verifying the conformance of each SLS rate share (R^{+}_{i,E_m}). This evaluation process takes into consideration distinct test conditions. Test1 and Test2 are devoted to an initial assessment and tuning of the explicit and implicit AC criteria. Test3 considers that the traffic injected into ingress I_2 is cross-traffic. Hence, E_1 is not aware of cross-traffic apart from the impact it may have on QoS estimation. This aspect is of major relevance as, due to the internal traffic dynamics and topology characteristics, a given amount of traffic may constitute an additional load just in parts of an edge-to-edge path. Tuning safety margins and exploring new thresholds, identifying the most relevant QoS parameters under control, are aspects explored in Test4. By default, the values of β^{+}_{i,E_m} and $T_{i,p}$ in Table II are applied, with exception of Test4 where several values of those variables are tested. Most of the presented results correspond to tests performed under high demanding conditions, with a flow interarrival of 300ms for SC1 and 500ms for SC2 and SC3. The measurement time interval Δt_i is set to $5s$. The results were obtained running a large number of simulations of about ten minutes each, after discarding an initial convergence period. Simulations up to forty minutes were also carried out in order to verify the consistency of the behavior under evaluation.

A. Test1 - Generic model operation

Detailing AC results: A detailed view of some of the controlled metrics for each class is shown in Figure 3. This figure represents the evolution of IPTD, ipdv in Δt_i, and the continuous evolution of IPLR. From the graphs in this figure, it is visible that:

(i) SC1 is very well controlled presenting a stable QoS behavior. IPTD is kept almost constant throughout the simulation period. The mean ipdv assumes a low value as a result of small variations, bounded by a well-defined maximum and minimum value;
(ii) for SC2, although the mean IPTD is well-bounded, in some time intervals, the maximum IPTD exceeds the defined thresholds. From the analysis of the plots at packet level and corresponding histograms, it is clear that the number of packets exceeding the QoS thresholds is very small. This is sustained by the statistical analysis of the involved time series, included in Table III;
(iii) SC3 IPLR evolution tends to the defined IPLR threshold of 10^{-1}. For SC2 traffic, IPLR has a less continuous behavior as it results from occasional loss events, converging to the defined threshold of 10^{-3}.

Table III summarizes statistical results obtained for each service class SC_i with regards to: the average number of active flows; the corresponding utilization; the percentage of packets exceeding the pre-defined IPTD and ipdv bounds; and the total loss ratio. The results show that: (i) the global utilization is kept high, and each class rate share is well accomplished (see Table II); (ii) the percentage of QoS violations at packet level is very small, in special for SC1, and the total IPLR is within the pre-defined thresholds. Note that, a QoS threshold violation does not necessarily imply a service QoS violation, as the defined concept of threshold comprises a safety margin to the QoS parameter target value.

AC rules effectiveness: When examining in detail which AC rules determine the generic behavior of the model discussed above, the following is identified:

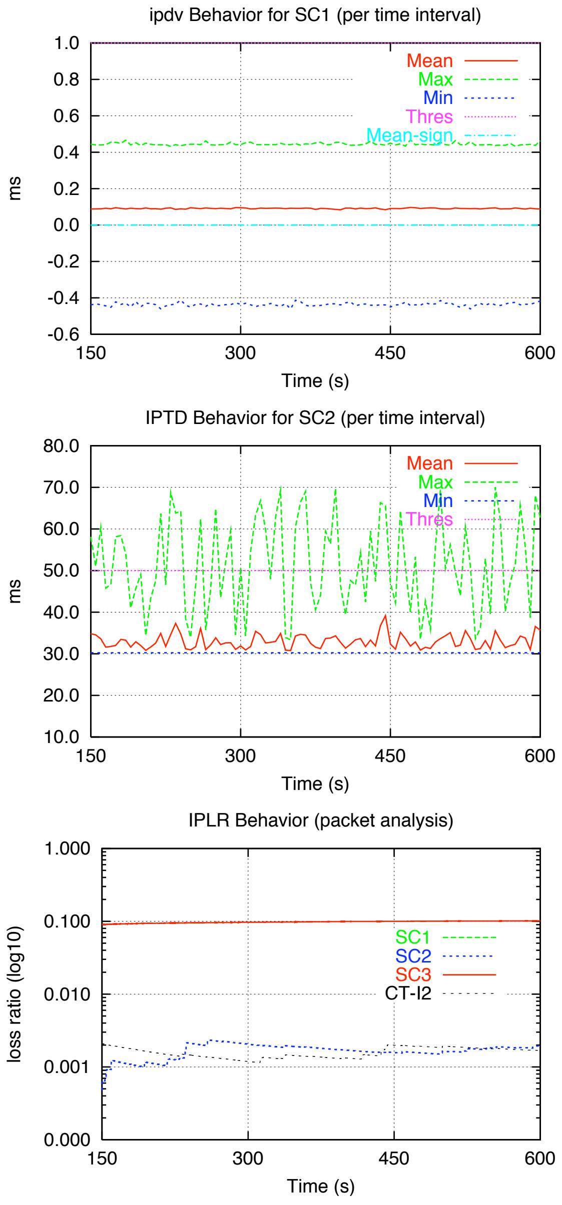

Fig. 3. Results in Δt_i: ipdv for SC1; IPTD for SC2. IPLR for all classes

(i) SC1 flows are controlled essentially by the SLS rate control rule (2) as a result of a stable QoS behavior associated with this high priority class;

(ii) AC for SC2 flows is triggered by SLS and/or QoS control rules ((2) and (3));

(iii) SC3 flows are controlled by the QoS control rule (as explained in Test2, the rate control rule is disabled);

(iv) IPLR violations assume a predominant role in setting the variable $AC\_status_{\Delta t_i}$ to a rejection mode in the QoS control rule.

Although the AC rules are effective in blocking new flows when QoS degradation or an excessive rate is sensed, the effect of previously accepted flows may persist over more than one measurement time interval, depending on these flows' characteristics and duration. Nonetheless, the system tends to recover fast. The eventual overacceptance is mainly caused by traffic fluctuations reflecting a low activity period of the admitted flows. In fact, low estimation in Δt_{i-1} may lead to false acceptance during Δt_i. This effect, likely to be stressed by concurrency and traffic characteristics, is particularly evident when observing the behavior of the SLS rate control rule for SC2 and the resulting AC decision, as shown in Figure 4[2]. To minimize this, more conservative estimates, larger safety margins and/or specific approaches to control concurrency [22], may be required.

TABLE III

TEST RESULTS AND STATISTICS AT PACKET LEVEL

Class	#act_f (avg)	%util(avg)	%viol:(IPTD;ipdv)	Total IPLR
SC1	107.5	7.4	(0.007;0.0005)	0.00009
SC2	59.5	22.9	(2.95; n.a.)	0.0027
SC3	70.2	42.9	(n.a.; n.a.)	0.106
CT-I2	58.6	22.3	(2.82; n.a.)	0.0022

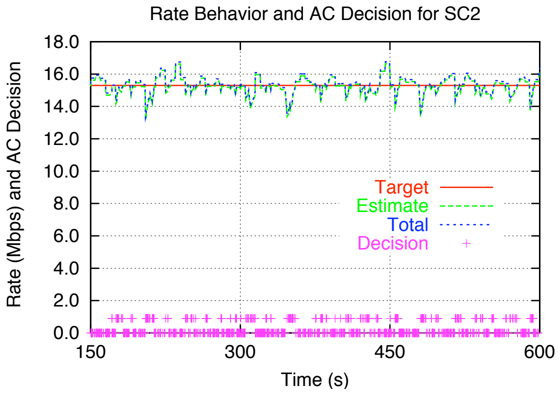

Fig. 4. Rate estimate evolution and AC decision for SC2

Test4 extends the results obtained above for each service class SC_i, exploring how the blocking probabilities evolve with the variation of the safety margins. Exploring new safety margins to avoid eventual QoS violations has resulted in consistent blocking probabilities while keeping high global utilization levels (see Figure 5). Note that, enlarging the default SC1 and SC2 safety margins (see Table II) in 10% is enough to avoid the QoS packet violations presented in Table III. For all test situations, $Total\ IPLR$ for SC3 remains very stable around 10^{-1}.

B. Test2 - Redefining the implicit AC criterion

The experiments assessing the implicit AC criteria effectiveness show that: (i) when rate variables determine the

---

[2] In Figure 4, *Target* line represents the value $\beta^+_{i,E_m} R^+_{i,E_m}$ above which AC rejection occurs, *Estimate* line represents the estimated rate or load of SLS^+_{i,E_m}, i.e., $\hat{R}^+_{i,(*,E_m)}$, and *Total* line reports to the previous estimate by adding the new flow rate r_j. *Decision* dots represent a positive (dots above the x-axis) or negative (dots overlapping the x-axis) AC decision, considering also the QoS control rule evaluation.

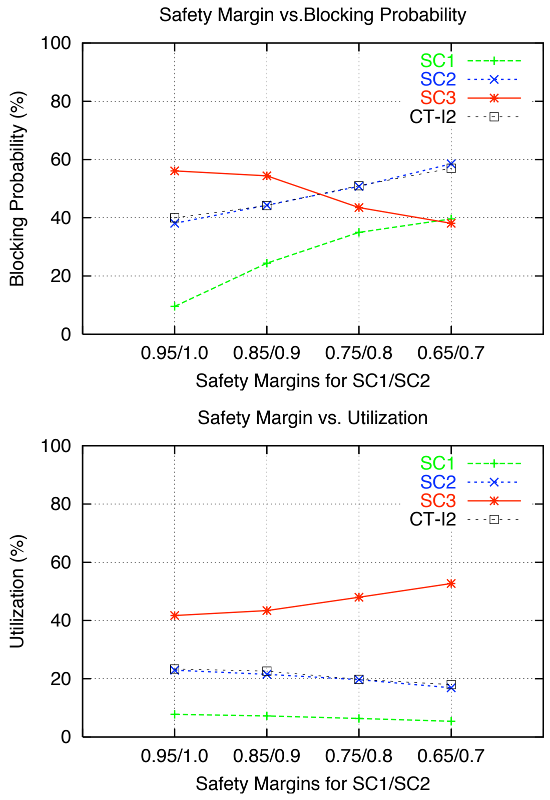

Fig. 5. Influence of varying the safety margins on the blocking probabilities and utilization

$AC\_status_{\Delta t_i}$ admittance value, this AC rule is clearly dominant, causing long rejection periods cyclically. In these periods, whose length depends on the number of admitted flows, Δt_i, and on the flow interarrival and holding time distributions, long-lived TCP flows progressively take over spare resources freed by departing flows. As a consequence, the rate estimate remains high and $AC\_status_{\Delta t_i}$ is kept in rejection mode until few flows are left. When this stage is reached, the $AC\_status_{\Delta t_i}$ enters in an acceptance mode and a new cycle begins; (ii) considering $AC\_status_{\Delta t_i}$ only determined by the QoS control rule has proved to be effective in maintaining IPLR bounded. However, as in (i), SC3 may exceed slightly its defined rate share, taking advantage of SC1/SC2 unused bandwidth resources, increasing the global utilization achieved by the system without an evident QoS degradation of SC1 and SC2.

C. *Test3 - Impact of cross traffic*

From this set of experiments, the relevance of the defined AC rules becomes evident in assuring service commitments in the domain. While the rate control rule assumes a preponderant role for service classes SC1 and SC2 to control the traffic load and indirectly QoS, particularly in situations involving concurrent traffic, the QoS control rule is decisive to assure or bound the domain QoS levels in presence of unmeasured cross traffic (see Figure 6). In real environments, where the two type of situations are likely to occur simultaneously, the two AC rules will complement each other to increase the domain capabilities to guarantee service commitments.

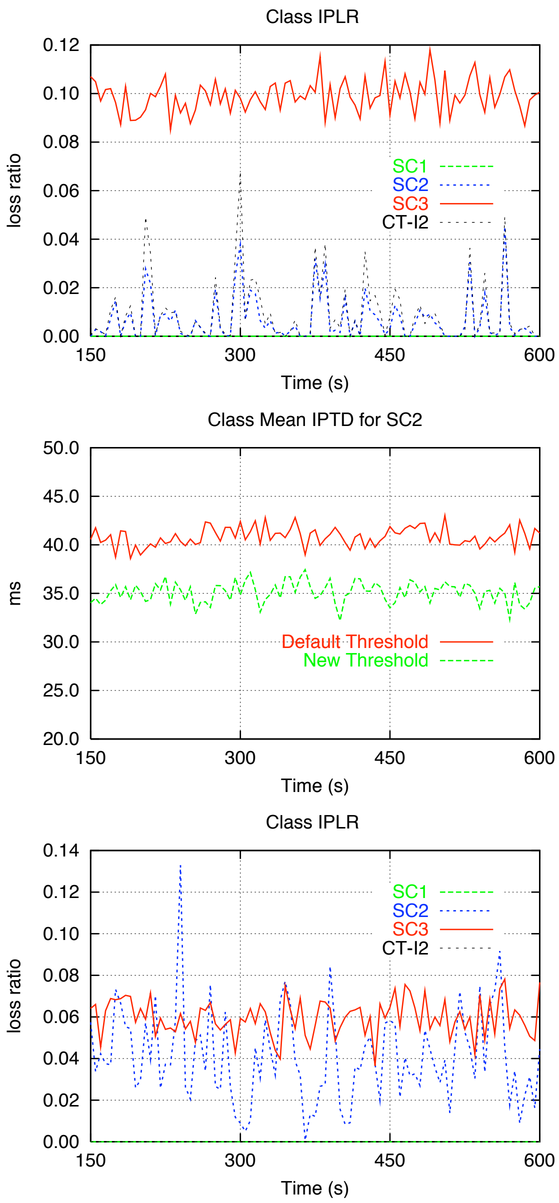

Fig. 6. Class mean IPLR for 20% of SC2 cross traffic; IPTD and IPLR behavior for 10% of SC2 cross traffic under new thresholds

Test3 has also included experiments with distinct QoS thresholds. As an example, Figure 6 shows the model's ability to control delay bounds. When a tighter IPTD threshold is set for SC2, AC is effective in bringing and maintaining IPTD controlled around 35ms. The same occurs when an IPLR threshold of 0.05 is set for SC2 and SC3 (more relaxed and tight than the previous one, respectively). The relevance of using a conservative degree of overprovisioning for more demanding classes became also evident in the presence of cross traffic from SC1.

From the above reasoning, it is important to remark that, knowing which AC rule has more influence on the AC decision process can also bring relevant information and directions for improving service configuration and provisioning both intra and interdomain.

VII. CONCLUSION

The evaluation of the model's performance has demonstrated that the self-adaptive behavior inherent to on-line measurements combined with the proposed AC rules is effective in controlling QoS and SLS commitments of each service class. The obtained measures of IPTD, ipdv and IPLR for the defined service classes exhibited a very stable behavior regarding the pre-defined QoS thresholds. IPLR was the most difficult metric to keep tightly controlled in each Δt_i, triggering the QoS control rule more frequently. However, the total IPLR achieved is in-line with the defined threshold and the percentage of QoS violations at packet level was very small for all classes. The bandwidth share configured for each class was well accomplished, and the overall utilization obtained was very high. The presence of cross traffic represents a bigger challenge for the AC criteria making evident the relevance of the two defined AC rules which complement each other to increase the domain capabilities to guarantee service commitments. Generically, the tests with different QoS thresholds have revealed the capacity of the AC criteria in bringing the QoS levels of each class to the established thresholds.

The results, clearly illustrating the role and relevance of the defined AC rules, have showed that service requirements can be efficiently satisfied or bounded, proving that the simplicity and flexibility of the model guided by systematic on-line monitoring can be explored to control successfully the quality of multiple Internet services.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Service," RFC 2475 (Informational), Dec. 1998, updated by RFC 3260. [Online]. Available: http://www.ietf.org/rfc/rfc2475.txt

[2] G. Huston, "Next Steps for the IP QoS Architecture," RFC 2990 (Informational), Nov. 2000. [Online]. Available: http://www.ietf.org/rfc/rfc2990.txt

[3] J. Babiarz, K. Chan, and F. Baker, "Configuration Guidelines for Diffserv Service Classes," IETF Draft : draft-ietf-tsvwg-diffserv-service-classes-02.txt (work in progress), Feb. 2006.

[4] S. R. Lima, P. Carvalho, and V. Freitas, "Self-adaptive Distributed Management of QoS and SLSs in Multiservice Networks," in *IEEE/IFIP International Conference on Integrated Management (IM 2005)*. Nice, France: IEEE Press, May 2005.

[5] S. R. Lima, P. Carvalho, and V. Freitas, "Distributed Admission Control for QoS and SLS Management," *Journal of Network and Systems Management - Special Issue on Distributed Management*, vol. 12, no. 3, pp. 397–426, Sept. 2004.

[6] Z. Duan, Z. Zhang, Y. Hou, and L. Gao, "A Core Stateless Bandwidth Broker Architecture for Scalable Support of Guaranteed Services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 2, pp. 167–182, Feb. 2004.

[7] B. Teitelbaum, S. Hares, L. Dunn, R. N. V. Narayan, and F. Reichmeyer, "Internet2 QBone: building a testbed for differentiated services," *IEEE Network*, vol. 13, no. 5, pp. 8–16, "Sep./Oct." 1999.

[8] L. Westberg, "Resource Management in Diffserv (RMD) Framework," IETF draft: draft-westberg-rmd-framework-04.txt (work in progress), Sept. 2003.

[9] I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per Flow Management," in *ACM SIGCOMM'99*, Oct. 1999.

[10] S. Jamin, P. Danzig, S. Shenker, and L. Zhang, "A Measurement-Based Call Admission Control Algorithm for Integrated Services Packet Networks (Extended Version)," *IEEE/ACM Transactions on Networking*, pp. 56–70, Feb. 1997.

[11] L. Breslau and S. Jamin, "Comments on the Performance of Measurement-Based Admission Control Algorithms," in *IEEE INFOCOM'00*, Mar. 2000.

[12] C. Cetinkaya, V. Kanodia, and E. Knightly, "Scalable Services via Egress Admission Control," *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 69–81, Mar. 2001.

[13] V. Elek, G. Karlsson, and R. Rnngren, "Admission Control Based on End-to-End Measurements," in *IEEE INFOCOM'00*, 2000.

[14] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," in *ACM SIGCOMM'00*, 2000.

[15] R. Mortier, I. Pratt, C. Clark, and S. Crosby, "Implicit Admission Control," *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 12, pp. 2629–2639, Dec. 2000.

[16] S. Fredj, S. Oueslati-Boulahia, and J. Roberts, "Measurement-based Admission Control for Elastic Traffic," in *17th International Teletraffic Congress*, June 2001.

[17] P. Georgatsos, J. Spencer, D. Griffin, P. Damilatis, H. Asgari, J. Griem, G. Pavlou, and P. Morand, "Provider-level Service Agreements for Inter-domain QoS delivery," *Fourth International Workshop on Advanced Internet Charging and QoS Technologies (ICQT04)*, Sept. 2004.

[18] J. Babiarz, K. Chan, and F. Baker, "Configuration Guidelines for Diffserv Service Classes," IETF Draft : draft-ietf-tsvwg-diffserv-service-classes-01.txt (work in progress), July 2005.

[19] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics," RFC 2330 (Informational), May 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2330.txt

[20] K. Glossbrenner, "Internet Protocol Data Communication Service - IP Packet Transfer and Availability Performance Parameters," ITU-T Recommendation I.380, 1999.

[21] S. Lima, P. Carvalho, and V. Freitas, "Measuring QoS in Class-based IP Networks using Multipurpose Colored Probing Patterns," in *SPIE ITCom 2004 - Performance, Quality of Service, and Control of Next-Generation Communication Networks*, vol. 5598. Philadelphia, U.S.A.: SPIE Press, ISBN 0-8194-5551-2, October 2004, pp. 171–182.

[22] S. R. Lima, P. Carvalho, and V. Freitas, "Handling Concurrent Admission Control in Multiservice IP Networks," in *IEEE Consumer Communications and Networking Conference (CCNC 2006)*. U.S.A: IEEE Press, Jan. 2006.