Noname manuscript No.
(will be inserted by the editor)

A Service-oriented Admission Control Strategy for Class-based IP Networks

Solange Rito Lima · Paulo Carvalho · Vasco Freitas

Received: Jul. 15, 2007 / Accepted: Oct. 5, 2007

Abstract The clear trend toward the integration of current and emerging applications and services in the Internet launches new demands on service deployment and management. Distributed service-oriented traffic control mechanisms, operating with minimum impact on network performance, assume a crucial role as regards controlling services quality and network resources transparent and efficiently. In this paper, we describe and specify a light-weight distributed admission control (AC) model based on per-class monitoring feedback for ensuring the quality of distinct service levels in multiclass and multidomain environments. The model design, covering explicit and implicit AC, exhibits relevant properties which allow managing QoS and SLSs in multiservice IP networks in a flexible and scalable manner. These properties, stemming from the way service-dependent AC and on-line service performance monitoring are proposed and articulated in the model's architecture and operation, allow a self-adaptive service and resource management, while abstracting from network core complexity and heterogeneity. A proof-of-concept is provided to illustrate the AC criteria ability in satisfying multiple service class commitments efficiently. The obtained results show that the self-adaptive behavior inherent to on-line measurement-based service management, combined with the established AC rules, is effective in controlling each class QoS and SLS commitments consistently.

1 Introduction

Managing multiservice networks is a complex and multidimensional problem involving heterogeneous media, protocols and technologies. Achieving a seamless and ubiquitous service management solution is even a more intricate issue attending to the plethora of service providers using distinct technologies, administrative policies and management strategies. Facing this diversity and complexity, the Quality of Service (QoS) quest will hardly be based on a single and general-purpose solution. Each solution requires the assessment of aspects such as its cost of integration into (or migration of) the existing network infrastructure,

Computer Communications Group, University of Minho
Department of Informatics, 4710-057 Braga, Portugal
E-mail: {solange,pmc,vf}@di.uminho.pt

the QoS guarantees provided or the scalability of the solution. Within this scenario, the design of service-oriented networks based on the class-of-service (CoS) paradigm, intends to provide a scalable yet simple support for deploying multiple IP services. To allow efficient management of each class resources and fulfill service level specification (SLS) commitments, admission control (AC) mechanisms are convenient to keep classes under controlled load and assure the required QoS levels. Although overprovisioning might be an attainable solution to provide QoS in network backbones for some service providers, it should not be assumed to be a generic and permanent answer. Apart from not being widely available, it is likely that the number of users and the demands of their applications will continue to outgrow the availability of resources. Thus, there is the need for additional service and traffic control mechanisms to guarantee that QoS commitments can be precisely specified and honored. Despite this need a key aspect and a major objective in the deployment of such mechanisms in real networks should be to kept the network control plane as simple as possible.

Achieving an encompassing AC approach that is simultaneously simple and easy to deploy for multiservice environments is, however, a challenge. When considering its operation across multiple domains, where distinct QoS solutions are likely to be in place and existing SLSs' need to be fulfilled, the challenge is even higher. Despite the wide range of AC approaches proposed in the literature (covered in Section 2), few studies deal with the management of multiple intradomain QoS levels and interdomain SLSs simultaneously, lacking in formalizing a generic model with concrete and flexible AC equations to be deployed in CoS networks. In this context, the distributed AC model based on edge-to-edge on-line QoS and SLS monitoring described in this paper brings new insights to perform encompassing and lightweight AC in multiservice class-based environments The proposed AC model aims to: (i) support multiple services with distinct assurance levels; (ii) control the QoS levels inside each domain and the existing SLSs between domains; (iii) operate intra and interdomain providing an unified end-to-end solution; (iv) be simple, flexible, efficient, scalable and easy to deploy in real environments.

This paper, fully describes the proposed AC model, presenting its architecture, the formalization of its components and AC criteria rules for the operation in a multiclass and multidomain environment. To sustain the real applicability of the defined rules, a proof-of-concept illustrating the model's self-adaptive ability in controlling distinct service levels is provided. The performance evaluation carried out in this paper, covering a wide range of operational scenarios, aims at assessing the model's effectiveness and efficiency in satisfying each class QoS levels and existing SLSs commitments for a multiclass domain. The model's key points and contributions regarding a scalable, self-adaptive and consistent management of multiple service levels is also debated.

The remaining of this document is organized as follows: a debate of representative AC approaches and the motivation for the present AC model is included in Section 2; the generic model architecture and operation is described in Section 3; the main network domain entities concerning multiservice AC, SLS and QoS management are formalized in Section 4, where an intuitive and expressive notation is introduced; this notation supports the intra and interdomain AC criteria specification provided in Section 5; the major model design key points are highlighted in Section 6; the AC model evaluation results are discussed in Section 7; finally, the conclusions are drawn in Section 8.

2 Related work and motivation

Defining an AC strategy for a multiservice network constitutes a particular challenge as service classes have distinct characteristics and require different QoS assurance levels. As the service predictability required is closely interrelated to the complexity and overhead of the AC strategy, finding an encompassing and light service-oriented AC model assumes a relevant role in controlling network resources and service levels efficiently.

The main advantage of centralized AC approaches [Duan et al.(2004)] [Teitelbaum et al.(1999)] is that centralizing state information and control tasks allow a global vision of the domain's QoS and operation, relieving the control plane inside the network. However, central entities need to store and manage large amounts of information, which in large and highly dynamic networks with many signaling messages and information state updates needing to be processed in real-time is even hard or prohibitive. The congestion and functional dependence on a single entity is another well-known problem of centralization.

To improve reliability and scalability in large networks, several approaches consider distributed AC with variable control complexity depending on the QoS guarantees and predictability required. 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 [Stoica and Zhang(1999), Westberg(2003)]. To provide predictive services (e.g., for soft real-time traffic) measurement-based AC (MBAC) [Jamin et al.(1997), Breslau and Jamin(2000)] and end-to-end MBAC (EMBAC) solutions [Cetinkaya et al.(2001), Elek et al.(2000)] have deserved special attention. Taking into account the burden of MBAC in performing AC in all network nodes, EMBAC measures edge-to-edge network status without requiring additional processing in the network core. AC is then left for network edges nodes or end-systems. Despite, the simplicity and scalability of EMBAC solutions, requiring reduced changes from networks, several disadvantages are commonly pointed out, namely: (i) the problematic of controlling SLSs is not covered; (ii) in [Cetinkaya et al.(2001)] the need for ingress-egress continuous measurements and updates in all real packets makes the solution more oriented to a single domain than to end-to-end; (iii) in [Elek et al.(2000)] the overhead of per-flow probing traffic may lead to bandwidth stealing and thrashing regimes [Breslau et al.(2000)], and the measurements' dependency on instantaneous network congestion increase estimation errors and QoS degradation. As regards fairness, a common concern of MBAC and EMBAC solutions is that, usually, both have implicit a single decision policy that tends to privilege small flows, flows with more relaxed QoS objectives and flows that traverse smaller path lengths [Breslau and Jamin(2000), Breslau et al.(2000)]. To control elastic traffic, for more efficient network utilization, implicit AC strategies have also been defined [Mortier et al.(2000), Fredj et al.(2001)].

As discussed, in these studies, detailed in [Lima et al.(2007a)], aspects such as the trade-off between service assurance level and network control complexity for a scalable and flexible support of distinct service types and corresponding SLSs, intra and interdomain, are not covered or balanced as a whole. The AC model discussed in this paper is a step forward in achieving a flexible and encompassing solution toward a scalable management of multiservice networks able to deal with the management of multiple intradomain QoS levels and interdomain SLSs simultaneously. A brief overview of the model operation principles is provided next so that the model formalization is better understood.

3 A Monitoring-based AC Model for QoS and SLS control

A primary idea of the proposed AC model is to take advantage of the need for on-line QoS and SLS monitoring in today's class-based networks and use the resulting monitoring information to perform distributed AC. Other crucial characteristic of this AC model is to consider a service-dependent degree of overprovisioning in order to achieve a simple and manageable multiservice AC solution. These levels of overprovisioning, controlled by the AC rules, allow to relax the AC process widening the range of service types covered by a monitoring-based AC solution.

To pursuit design goals such as flexibility, scalability and easy deployment, the AC model comprises: (i) distributed control between edge nodes; (ii) no control tasks within the network core; (iii) reduced state information and control overhead; (iv) measurement-based self-adaptation regarding network dynamics. This model, oriented to accommodate multiple services, intends to allow AC irrespectively of the applications' ability to signal the network.

3.1 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.

Service definition, involves the definition of basic services oriented to different application requirements, the definition of relevant QoS parameters to control within each service

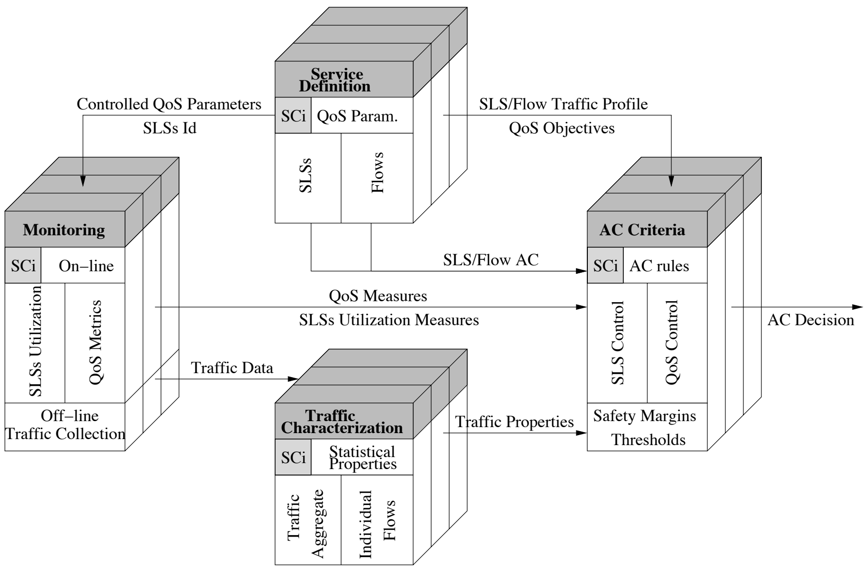

Fig. 1 AC model architecture

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.*

3.2 Model Overview

The proposed AC model resorts to edge-to-edge on-line monitoring to obtain feedback of each class performance so that proper AC decisions can be made. To dynamically control traffic entering a network domain, the model underlying AC rules control both QoS levels in the domain and the sharing of active SLS between domains. As illustrated in Figure 2, while ingress routers perform explicit or implicit AC depending on the application type and corresponding service class, egress routers perform on-line QoS monitoring and SLS control. *On-line QoS Monitoring*, carried out on an ingress-egress basis, measures specific metrics for each service type. These measures reflect a quantitative view of the service level available from each ingress node. *SLS Control* monitors the usage of downstream SLSs at each egress, to ensure that traffic to other domains does not exceed the negotiated profiles. The obtained measures are periodically sent to the corresponding ingress routers to update an Ingress-Egress service matrix used for distributed AC and active service management.

The *end-to-end case* is viewed as a repetitive and cumulative process of AC and available service computation, performed at ingress nodes. At each domain, an ingress node decides if a flow can be accepted, and if so the domain service metric values are added to the flow request to inform the downstream domain of the service available so far. Using the incoming and its own measures each domain performs AC. When a rejection occurs, the source is notified directly from the rejection point. This solution leads to a generic AC model, which can be applied both to source and transit domains. A cumulative process for end-to-end QoS computation is consistent with the cascade approach for the support of interoperator IP-based services, which is in conformance with the Internet structure and operation, and more scalable than the source-based approach [Georgatsos et al.(2004)].

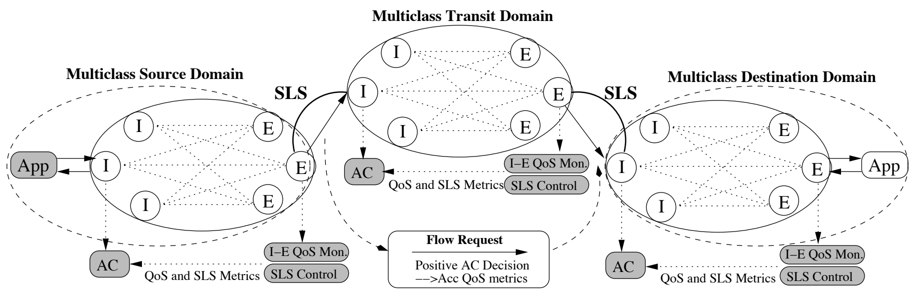

Fig. 2 Distributed monitoring-based AC model

4 Specification of Multiservice Domain Entities

In this section, we specify the main components of a generic network domain as regards the provision of multiple services. Following [Lima et al.(2005)], we specify: (i) service classes; (ii) upstream SLSs; (iii) downstream SLSs and (iv) traffic flows. Network resources are implicitly considered and controlled by the edge-to-edge monitoring process. The proposed notation, summarized in Table 1, helps to clarify the context of service-oriented management and it will be used in Section 5 for specifying the service-oriented AC criteria.

4.1 Service Classes Specification

Considering a multiclass domain D_x comprising N ingress nodes and M egress nodes, lets $I^{D_x} = \{I_1, I_2, ..., I_N\}$ and $E^{D_x} = \{E_1, E_2, ..., E_M\}$ represent the set of ingress and egress nodes, respectively[1]. The set of service classes supported within D_x is defined as $SC^{D_x} = \{SC_1, SC_2, ..., SC_Y\}$. For each class $SC_i \in SC^{D_x}$, the set of QoS parameters under control is defined as $P_{SC_i} = \{(P_{i,1}, \beta_{i,1}), ..., (P_{i,P}, \beta_{i,P})\}$, where each $P_{i,p} \in P_{SC_i}$ is the class parameter target value and $0 \leq \beta_{i,p} \leq 1$ is the parameter safety margin. Each parameter upper bound or threshold, used for triggering traffic control mechanisms such as AC, is given by $T_{i,p} = \beta_{i,p} P_{i,p}$.

In practice, the service classes supported in D_x are closely related to the service levels negotiated with both upstream and downstream domains. In this way, for class SC_i, we define the set of SLSs accepted in D_x coming from any upstream domain D_x^- as $SLS_{SC_i}^{D_x^-} = \{SLS_{i,I_n} | I_n \in I^{D_x}\}$, and the set of SLSs negotiated with any downstream domain D_x^+ as $SLS_{SC_i}^{D_x^+} = \{SLS_{i,E_m}^+ | E_m \in E^{D_x}\}$. SLS_{i,I_n} identifies a specific SLS accepted for SC_i with D_x^-, connecting D_x through I_n, and SLS_{i,E_m}^+ identifies a specific SLS negotiated for SC_i with D_x^+, accessible from D_x through E_m (see Figure 3). Therefore, D_x is a service provider for D_x^- and a customer of D_x^+. The case of flows entering the domain D_x without pre-negotiated SLSs (usually dial-up users) is also covered, and the notation $\notin SLS$ is introduced for this purpose.

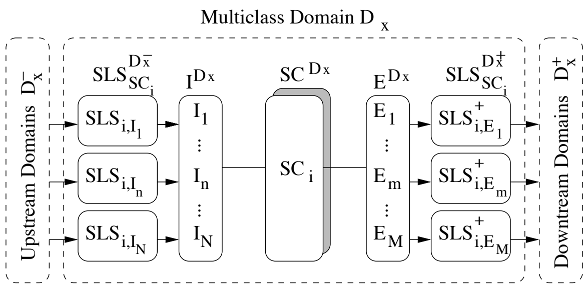

Fig. 3 Domain Main Elements and Notation

[1] When possible, the entities under specification use indexes based on the corresponding service class and involved ingress and egress nodes. As the AC model is class-based and operates edge-to-edge, this approach enriches semantically the notation, while keeping it intuitive. To simplify the notation, and without losing generality, each ingress or egress distinct interface is treated as a virtually distinct ingress I_n or egress E_m.

4.2 Upstream SLSs Specification

Defining a standard set of SLS parameters and semantics is crucial for ensuring end-to-end QoS delivery and for simplifying interdomain negotiations. Several working groups have been committed to SLS definition and management [Morand et al.(2004), Goderis et al.(2003), Sevasti and Campanella(2001)]. Following these inputs, from an AC perspective, an upstream SLS_{i,I_n}, should consider:

1. $SLS_{i,I_n} \rightarrow Scope$ is specified as a pair $(I_n, E^{'})$ where the ingress node I_n is the access point of the upstream domain D_x^- to D_x and $E^{'} \subseteq E^{D_x}$ is identified according to the destination domains D_x^+ defined in SLS_{i,I_n}.

2. $SLS_{i,I_n} \rightarrow SC_{id}$ classifies and identifies the service type to be provided by D_x to packets belonging to SLS_{i,I_n}. The DS Code Point is a possible SC_{id} in Diffserv domains.

3. $SLS_{i,I_n} \rightarrow TrafficProfile$ specifies the traffic characteristics of SLS_{i,I_n}, allowing to identify whether traffic is in or out-of-profile. For instance, when using a token bucket policer, the SLS traffic profile can be specified as $TB(R_{i,I_n}, b_{i,I_n})$ with rate R_{i,I_n} and burst size b_{i,I_n}.

4. $SLS_{i,I_n} \rightarrow ExpectedQoS$ specifies the expected QoS parameters, i.e., $P_{SLS_{i,I_n}} = \{P_{i,I_n,1}, ..., P_{i,I_n,P'}\}$, with $P' \subseteq P$, where P is the cardinality of P_{SC_i}. Note that, regardless the incoming I_n, each QoS parameter $P_{i,I_n,p}$ value is bounded by the corresponding service class QoS parameter $P_{i,p}$. Embedding the expected SLS parameters values in the respective class parameter target values is of paramount importance as QoS and SLS control in the domain is clearly simplified. Examples of $P_{i,I_n,p}$ are IP Transfer Delay ($IPTD_{i,I_n}$), IP Delay Variation ($ipdv_{i,I_n}$), IP Loss Ratio ($IPLR_{i,I_n}$).

5. $SLS_{i,I_n} \rightarrow ServSched$ determines the time interval $[t_{i,I_n,0}, t_{i,I_n,f}]$ in which the service is due to be scheduled, giving that $t_{i,I_n,0}$ expresses the SLS starting time and $t_{i,I_n,f}$ the SLS expiring time. In [Sevasti and Campanella(2001)], this interval is recommended to be month-range.

4.3 Downstream SLSs Specification

The specification of a downstream SLS^+_{i,E_m} follows the SLS template and notation introduced above for upstream SLSs, inserting the downstream identifier "+" and adapting the corresponding definitions accordingly. The negotiated traffic profile for SLS^+_{i,E_m}, given by $SLS^+_{i,E_m} \rightarrow TrafficProfile$, should aggregate the traffic profiles of all accepted SLS_{i,I_n} for SC_i that may use E_m to leave D_x.

4.4 Flow Specification

Depending on each application ability for signalling its service requirements, traffic flows may undergo either implicit or explicit AC. For implicit AC, the relevant fields to consider are the source, destination and service class identifiers, i.e. Src_{id}, Dst_{id}, SC_{id}. For explicit AC, apart from these fields, specifying a flow F_j includes defining the $TrafficProfile$ and the required QoS parameters $ReqQoS$. In addition, a specific field required for end-to-end AC operation is $AccQoS$; other optional fields, explained below, are I_{src}, SLS_{id} and D_{id}. In more detail:

1. $F_j \rightarrow TrafficProfile$ can be described by a token bucket policer $TB(r_j, b_j)$;

2. $F_j \rightarrow ReqQoS$ is defined as $P_{F_j} = \{(P_{j,1}, \gamma_{j,1}), ..., (P_{j,P''}, \gamma_{j,P''})\}$, with $P'' \subseteq P' \subseteq P$. This subset inclusion also means that, each $P_{j,p}$ value must be bounded by the corresponding $P_{i,I_n,p}$ value which, in turn, must be bounded by the corresponding class target value $P_{i,p}$. An optional tolerance to $P_{j,p}$ degradation, expressed by $\gamma_{j,p}$, may be considered by the AC criteria (see Section 5.3);

3. $F_j \rightarrow AccQoS$ is used to accumulate QoS metric values in a multidomain end-to-end AC operation (see Section 5.3);

4. $F_j \rightarrow I_{src}$ is an optional field which allows to identify the source domain ingress node I_{src}. This is the only ingress node that may need to be self-identified when receiving AC response notification messages for traffic conditioning (TC) configuration; $F_j \rightarrow SLS_{id}$ and $F_j \rightarrow D_{id}$ are also optional fields used for interdomain authentication.

4.5 Monitoring and Controlling per-Class QoS Metrics

For service class SC_i and ingress node I_n, a dynamic Ingress-Egress Service matrix is used to control QoS levels and support AC decisions. The service data stored in the matrix is provided by egress nodes which send monitoring updates at each measuring time interval Δt_i. This data includes the class QoS parameters measured from an (I_n, E_m) perspective, i.e., $\tilde{P}_{i,(I_n,E_m),p}$. Using this measured data and corresponding class thresholds, a QoS status indicator, defined as $AC\_Status_{\Delta t_i}$, is computed and used by AC for determining whether or not incoming traffic from I_n to E_m can be accepted in Δt_i (see QoS control rule in Section 5.2).

Table 1 Model notation summary

Notation	Definition	Description
Domain Notation		
D_x, D_x^-, D_x^+		Current, upstream and downstream domains
I^{D_x}	$\{I_1, ..., I_n, ..., I_N\}$	Set of ingress nodes in domain D_x
E^{D_x}	$\{E_1, ..., E_m, ..., E_M\}$	Set of egress nodes in domain D_x
Service Class Notation		
SC^{D_x}	$\{SC_1, ..., SC_i, ..., SC_Y\}$	Set of service classes supported in D_x
P_{SC_i}	$\{(P_{i,1}, \beta_{i,1}), ..., (P_{i,P}, \beta_{i,P})\}$	Set of controlled QoS parameter for SC_i
$P_{i,p}, \beta_{i,p}$	$1 < p < P$	Target and Safety Margin of parameter p for SC_i
$SLS_{SC_i}^{D_x^-}$	$\{SLS_{i,I_n} \| I_n \in I^{D_x}\}$	SLSs negotiated in D_x with D_x^- for SC_i
SLS_{i,I_n}		Upstream SLS for SC_i connecting D_x through I_n
$P_{SLS_{i,I_n}}$	$\{P_{i,I_n,1}, ..., P_{i,I_n,P'}\}$	Set of expected QoS parameters for SLS_{i,I_n}
$P_{i,I_n,p'}$	$1 < p' < P'$	Target value of QoS parameter p'
$SLS_{SC_i}^{D_x^+}$	$\{SLS_{i,E_m}^+ \| E_m \in E^{D_x}\}$	SLSs negotiated in D_x with D_x^+ for SC_i
SLS_{i,E_m}^+		Downstream SLS for SC_i leaving D_x through E_m
P_{SLS_{i,E_m}^+}	$\{P_{i,E_m,1}^+, ..., P_{i,E_m,P^+}^+\}$	Set of expected QoS parameters for SLS_{i,E_m}^+
Flow Notation		
$F_j \in SLS_{i,I_n}$		Flow j belonging to an upstream SLS requiring AC
P_{F_j}	$\{(P_{j,1}, \gamma_{j,1}), ..., (P_{j,P''}, \gamma_{j,P''})\}$	Set of QoS parameter requirements for F_j
$P_{j,p''}, \gamma_{j,p''}$	$1 < p'' < P''$	Target value and tolerance to QoS parameter p''

5 AC Criteria Specification

Following the generic AC model description provided in Section 3.2, for controlling both the QoS levels and the utilization of existing SLSs, the following rules have been defined: (i) rate-based SLS control rules; (ii) QoS parameters control rules; (iii) end-to-end QoS control Rules. The specification of these rules, following the notation introduced in section above, is summarized in Table 2.

5.1 Rate-based SLS Control Rules

As each SLS_{i,I_n} and SLS^+_{i,E_m} have specified a negotiated rate, R_{i,I_n} and R^+_{i,E_m} respectively, a rate-based Measure-Sum (MS) algorithm can be applied to control SLSs utilization at each network edge node.

Explicit AC - At each ingress node I_n, verifying if a new flow $F_j \in SLS_{i,I_n}$ can be admitted 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 Eq. (1), $\tilde{R}_{i,(I_n,*)}$ is the current measured load or estimated rate of flows using SLS_{i,I_n}; r_j is the rate specified by the new flow F_j; β_{i,I_n} (with $0 < \beta_{i,I_n} \leq 1$) is a safety margin defined for the negotiated rate R_{i,I_n} for SLS_{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 Eq. (2), $\tilde{R}^+_{i,(*,E_m)}$ is the current measured load of flows using SLS^+_{i,E_m}, considering all the ingress-to-E_m estimated rates of SC_i 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 specified by the new flow F_j; β_{i,E_m} (with $0 < \beta_{i,E_m} \leq 1$) is the safety margin for the rate R^+_{i,E_m} defined in SLS^+_{i,E_m}. Recall that this safety margin determines the degree of overprovisioning for the corresponding SC_i.

When D_x is a transit domain, verifying if the upstream SLS_{i,I_n} can accommodate the new flow profile (Eq. 1) is optional. In fact, assuming that the upstream domain D_x^- controls the corresponding downstream SLS traffic load through a process equivalent to the one ruled by Eq. (2), the current domain D_x can control SLS_{i,I_n} using a simple TC mechanism based on the negotiated traffic profile. For source and destination domains, unless internal SLS_{i,I_n} and SLS_{i,E_m} are defined, Eqs. (1) and (2) are not applicable.

The rate control rules for the admission of flows not sustained by an SLS, i.e. $F_j \notin SLS_{i,I_n}$, resort to equation $\tilde{R}^{\notin SLS}_{i,(I_n,*)} + r_j \leq \beta_{i,I_n} R^{\notin SLS}_{i,I_n}$. $R^{\notin SLS}_{i,I_n}$ is a rate-based parameter defined to limit traffic not sustained by a specific SLS, allowing a better control of the rate share in D_x and of SLS^+_{i,E_m} utilization, while avoiding possible denial-of-service to flows $F_j \in SLS_{i,I_n}$.

Implicit AC - For a service class SC_i under implicit AC, as flows are unable to describe r_j, SLS control equations become similar to the QoS control equation (Eq. (3)), considering $P_{i,p}$ as a rate-based parameter. Therefore, traffic flows are accepted or rejected implicitly according to the variable AC_Status computed once for Δt_i ($AC\_Status_{\Delta t_i}$). Additionally, the variable $Adm\_Flows_{\Delta t_i}$ may constrain the number of flows which can be implicitly accepted in Δt_i.

5.2 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 monitoring interval Δt_i is updated after checking the controlled parameters $P_{i,p}$ of SC_i against the corresponding pre-defined threshold $T_{i,p}$, i.e.,

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

where $T_{i,p}$, as explained in Section 4.1, reflects a safety margin $\beta_{i,p}$ to the QoS parameter target value, i.e.,$T_{i,p} = \beta_{i,p} P_{i,p}$. Eq. (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}. In practice, the QoS control rules are applied on an Ingress-Egress basis using information stored in the QoS matrix available at each I_n.

5.3 End-to-End Admission Control

Assuming a consistent mapping between the service classes in domains D_x^-, D_x and D_x^+, making an AC decision at ingress node I_n of domain D_x, should consider the rule:

$$\forall P_{j,p} \in P_{F_j} : (\texttt{op}_1(P_{j,p}^{acc^-}, P_{i,p}))\ \texttt{op}_2\ (\gamma_{j,p} P_{j,p}) \tag{4}$$

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 when crossing previous domains, $P_{j,p}^{acc^-}$, affected by the corresponding target value of $P_{i,p}$ in the present domain D_x. Depending on each parameter semantics, $\texttt{op}_1$ and $\texttt{op}_2$ may express different operations. For instance, when $P_{j,p}$ is a delay parameter, a positive AC decision occurs when $add(P_{j,p}^{acc^-}, P_{i,p}) \leq \gamma_{j,p} P_{j,p}$. If the flow can be accepted in D_x, the new available service computation to be included in the flow request is given by $P_{j,p}^{acc} = \texttt{op}_1(P_{j,p}^{acc^-}, P_{i,p})$.

6 Model key points

This section highlights the most important features of the model concerning a scalable self-adaptive management of QoS and SLSs in multiservice networks. These features stem from two important management tasks covered and interrelated in the model, which are service-dependent AC and on-line service performance monitoring.

Scalable service-dependent AC - the major key points identified are as follows:

(i) different service types, QoS parameters and SLSs can be controlled simultaneously in a distributed and simple fashion, involving only edge nodes, i.e., the network core is kept unchanged and treated as a black box. This provides a convenient level of abstraction and independence from network core complexity and heterogeneity;

(ii) the state information is service and (I_n, E_m) based which, apart from leading to reduced state information, is particularly suitable for SLS auditing. Per-flow state information is only kept at the source domain ingress routers for TC (when applicable), while other downstream domains may maintain TC based on the SLS aggregated traffic profile, as usual;

Table 2 Control Rules Summary

TYPE OF RULE	DESCRIPTION
SLS Rate Control Rules	Verify upstream and downstream SLSs utilization
$\tilde{R}_{i,(I_n,*)} + r_j \leq \beta_{i,I_n} R_{i,I_n}$	$\tilde{R}_{i,(I_n,*)}$ - current measured rate of flows using SLS_{i,I_n} independently of the egress nodes E_m involved; r_j - rate of the new flow F_j; $0 < \beta_{i,I_n} \leq 1$ - service-dependent safety margin defined for the negotiated rate R_{i,I_n} of SLS_{i,I_n}.
$\tilde{R}^{+}_{i,(*,E_m)} + r_j \leq \beta^{+}_{i,E_m} R^{+}_{i,E_m}$	$\tilde{R}^{+}_{i,(*,E_m)}$ - current measured rate of flows using SLS^{+}_{i,E_m}, considering all ingress-to-E_m estimated rates of flows going through E_m; r_j - rate of the new flow F_j; $0 < \beta^{+}_{i,E_m} \leq 1$ - service-dependent safety margin for the rate R^{+}_{i,E_m} defined in SLS^{+}_{i,E_m}.
QoS Control Rules	Verify the conformance of QoS levels in the domain
$\forall (P_{i,p}, \beta_{i,p}) \in P_{SC_i} : \tilde{P}_{i,p} \leq T_{i,p}$	$\tilde{P}_{i,p}$ - ingress-to-egress measured QoS parameter; $\beta_{i,p}$ - corresponding safety margin; $T_{i,p}$ - parameter's upper bound or threshold, given by $T_{i,p} = \beta_{i,p} P_{i,p}$, used to set the acceptance status for Δt_i.
End-to-end Control Rules	Cumulative computation and verification of e2e QoS
$\forall P_{j,p} \in P_{F_j}$: $(\mathtt{op}_1\ (P^{acc^-}_{j,p}, P_{i,p}))\ \mathtt{op}_2\ (\gamma_{j,p} P_{j,p})$	$P_{j,p}$ - flow QoS parameter, allowing a tolerance factor $\gamma_{j,p}$; $P^{acc^-}_{j,p}$ - cumulative value for $P_{j,p}$ when crossing upstream domains; $P_{i,p}$ - corresponding target value in present domain.

(iii) the signaling process for intra and interdomain operation is simple, horizontal and fluid. The flow AC request is used both for per-domain AC and for end-to-end available service computation along the data path, and no soft/hard state behavior and symmetric routing paths are imposed;

(iv) the AC model provides enough flexibility to accommodate technological, service and application evolution. Important aspects contributing to the model's flexibility are: (a) the service-dependent nature of AC rules and adjustable parameterization; (b) the ability to be decoupled between ingress and egress nodes; (c) the conceptual modular independence between AC and monitoring tasks, which increases their ability to integrate new developments and improvements.

Scalable QoS and SLS Monitoring - the major key points identified are as follows:

(i) the control of the SLSs' negotiated QoS parameters is embedded in the QoS control of the corresponding service classes, reducing the amount of SLSs' dynamic state information and control overhead. At SLS level, the traffic load is the only parameter measured locally at I_n or E_m nodes. In more detail, considering the set of the expected QoS parameters of each SC_i, SLS_{i,I_n} and F_j respectively, accepting SLSs and flows based on the subset inclusion rule $P_{F_j} \subseteq P_{SLS_{i,I_n}} \subseteq P_{SC_i}$ is of crucial importance regarding the scalability of the control strategy;

(ii) the systematic use of on-line monitoring for traffic load and QoS metrics' estimation in a per-class basis, while allowing an adaptive service management, avoids per-application intrusive traffic to obtain measures and reduces AC latency as measures are available on-line. Furthermore, systematic measurements have an intrinsic auto-corrective nature, allowing to detect short or long-term traffic fluctuations depending on the measurement time interval, and implicitly take into account the effect of cross-traffic and other internally generated traffic (e.g., routing, management, multicast);

(iii) the use of multipurpose active monitoring, i.e., the use of light probing patterns able to capture simultaneously the behavior of multiple QoS metrics of each class, also brings potential advantages to scalability [Lima et al.(2007b)].

The model properties defined above tend to increase the model's resilience to scalability problems. A summary identifying the impact that large-scale environments may have on the proposed AC solution is highlighted in Table 3.

Table 3 Issues on the AC model scalability

Main variables	Scalability issues
network dimension	Number of edge nodes involved: - impacts on edge state information and monitoring overhead - may increase the need for handling concurrent AC Core complexity: - no impact on model overhead - no significant impact expected on AC criteria efficiency
number of SCi	SCi state information at edge nodes QoS monitoring overhead Probing intrusion (if applicable)
number of SLSs	SLS state information at involved edge nodes SLS utilization monitoring overhead No impact on QoS monitoring overhead
number of flows	Number of AC decisions No impact on domain state information Traffic Conditioning at source domain I_n (if applicable)

7 Self-adaptive QoS and SLS Management

Having defined the AC model conceptually, this section provides a proof-of-concept of the proposed solution, illustrating its self-adaptive ability in controlling QoS and SLSs in a multiclass domain. For this purpose, a prototype was set using NS-2. This prototype implements three functional interrelated modules - Automatic Source Generation Module, AC Decision Module, and QoS and SLS Monitoring Module. Figure 4 presents a simplified diagram of the simulation model architecture, including the relation between these modules and the main underlying functions and variables. The two recursive modules represented in gray are responsible for the dynamic behavior of traffic source generation and monitoring.

7.1 Test environment

Defined service classes Considering current differentiated service configuration guidelines [Babiarz et al.(2006)], three service classes are defined. As basic policy, TCP and UDP traffic are treated separately; UDP traffic is further divided according to its QoS requirements. Table 4 summarizes the service classes implemented, highlighting AC and QoS monitoring decisions and parameters used to configure the AC rules controlling both SLS utilization and domain QoS levels. The negotiated rates (R^+_{i,E_m}) of downstream SLSs have been defined according to the traffic load share intended for the corresponding class in the domain. As shown, the parameterization of the AC rules is service-dependent and larger safety margins β^+_{i,E_m} and tighter thresholds $T_{i,p}$ are defined for more demanding classes. For instance, a $\beta^+_{i,E_m} = 0.85$ corresponds to impose a safety margin or degree of overprovisioning of 15% to absorb load fluctuations and optimistic measures. The AC thresholds $T_{i,p}$ consider

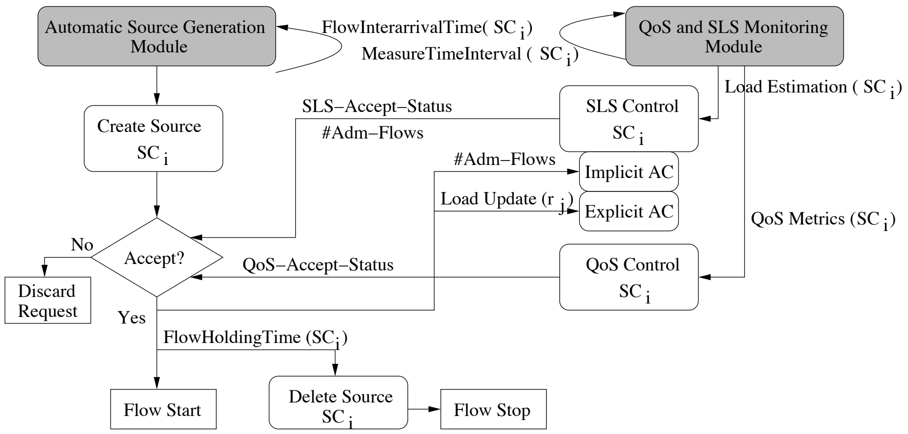

Fig. 4 Simulation model diagram

domain's characteristics and perceived QoS upper bounds for common applications and services [Seitz(2003), D. Miras et al.(2002)]. Table 5 illustrates the type of traffic sources in use and the corresponding parameters.

Domain Topology The network domain consists of ingress routers I_1, I_2, a multiclass network core and an edge router E_1. The service classes SC1, SC2 and SC3 are implemented in all the domain nodes. I_2 is used to inject concurrent or cross traffic (referred as CT-I2), allowing to evaluate concurrency effects on distributed AC and assess the impact of cross traffic on the AC model performance. The scenarios with cross traffic (see Figure 5) 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 rule (Eq. (2)). 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. The domain routers implement the service classes according to a

Table 4 Service Classes SC_i

SC_i	Serv. Type	AC Type	R^+_{i,E_m}	β^+_{i,E_m}	$P_{i,p}$	$T_{i,p}$	Example
SC1	guaranteed (hard-RT)	explicit and conservative	3.4Mbps (10% share)	0.85	IPTD ipdv IPLR	35ms 1ms 10^{-4}	VoIP Circuit Emulation Conversational UMTS
SC2	predictive (soft-RT)	explicit and flexible	17Mbps (50% share)	0.90	IPTD IPLR	50ms 10^{-3}	audio/video streaming
SC3	best-effort	implicit	13.6Mbps	1.0	IPLR	10^{-1}	elastic applications

Table 5 Traffic Sources

SC_i	Traffic Sources	interarrival time	holding time
SC1	Exponential or Pareto on/off (64kbps, pkt=120B,on/off = 0.96/1.69ms)	Exponential 0.3s	Exponential 90s
SC2	Exponential or Pareto on/off (256kbps, pkt=512B, on/off = 500/500ms)	Exponential 0.5s	Exponential 120s
SC3	FTP traffic (pkt=512B)	Exponential 0.5s	Exponential 180s

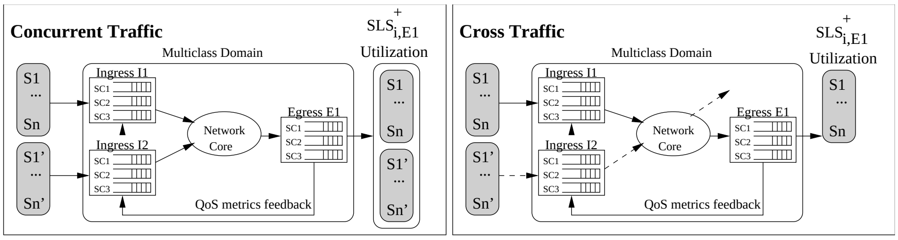

Fig. 5 Concurrent vs. cross traffic for SCi

hybrid Priority Queuing - Weighted Round Robin (PQ-WRR(2,1)) scheduling discipline, with RIO-C as AQM mechanism. The domain internodal links capacity is 34Mbps, with a 15ms propagation delay.

7.2 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: (i) Test1 and Test2 are devoted to an initial assessment and tuning of the explicit and implicit AC criteria; (ii) 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; (iii) Test4 explores the model's ability to fulfill new thresholds to the most relevant QoS parameters under control; (iv) Test5 and Test6 study other important aspects which may impact on the model's behavior, namely, the influence of traffic characteristics and the impact of the measurement time interval Δt_i dimension. Thus, to complement the study with a default Δt_i of $5s$, larger intervals 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.

7.2.1 Test1 - Generic model operation

A detailed view of some of the controlled metrics for each class is shown in Figure 6. 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

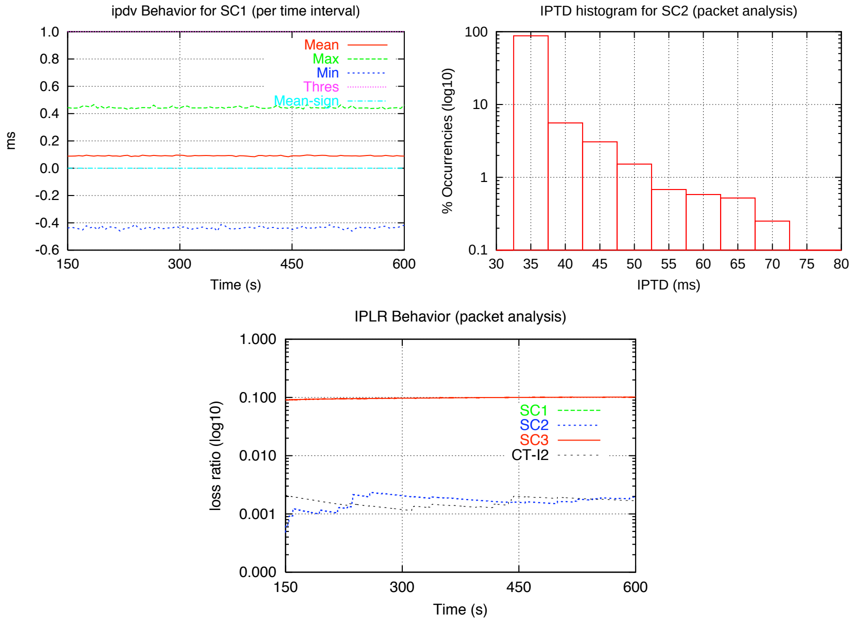

Fig. 6 Results in Δt_i: ipdv for SC1; IPTD for SC2. IPLR for all classes

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 6;

(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 6 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 4); (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.

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

Table 6 Test results and statistics at packet level

Class	#active_flows (avg)	%utilization(avg)	%violations:(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

(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 7[2]. To minimize this, more conservative estimates, larger safety margins and/or specific approaches to control concurrency may be required. Exploring new safety margins to avoid eventual QoS violations has resulted in consistent blocking probabilities while keeping high global utilization levels (see Figure 8). Note that, enlarging the default SC1 and SC2 safety margins (see Table 4) in 10% is enough to avoid the QoS packet violations presented in Table 6. For all test situations, $Total\ IPLR$ for SC3 remains very stable around 10^{-1}.

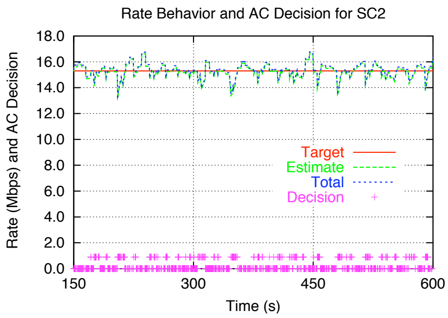

Fig. 7 Rate estimate evolution and AC decision for SC2

7.2.2 Test2 - Redefining the implicit AC criterion

The experiments assessing the implicit AC criteria effectiveness show that: (i) when rate variables determine the $AC\_status_{\Delta t_i}$ admittance value, this AC rule is clearly dominant, causing long rejection periods cyclically. In these periods, whose length depends on the

[2] In Figure 7, *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., $\tilde{R}^{+}_{i,(*,E_m)}$, 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.

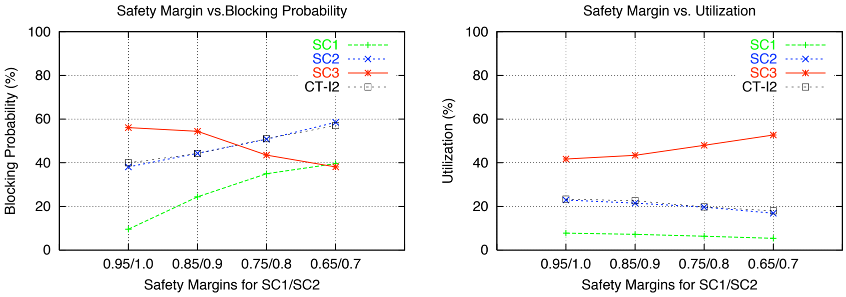

Fig. 8 Influence of varying the safety margins on the blocking probabilities and utilization

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 without an evident QoS degradation of SC1 and SC2.

7.2.3 Test3 - Impact of cross traffic

The way cross-traffic impacts on the system performance varies with the service class considered as cross-traffic.

(i) In the presence of SC2 cross traffic, the main rule determining AC decisions in this class is the QoS control rule, with $AC\_status_{\Delta t_i} = reject$ activated by IPLR violations. This rule by itself maintains the QoS levels controlled, as shown in Figure 9. The SLS rate control rule and the corresponding safety margins are now less relevant and restrictive. The global utilization of SC2 (I_1+ CrossTraffic) decreases slightly comparing to the concurrent case, with the amount of traffic accepted at I_1 being adjusted according to the amount of cross traffic. This decrease is a consequence of the effect of cross traffic on $C1$ queue occu-

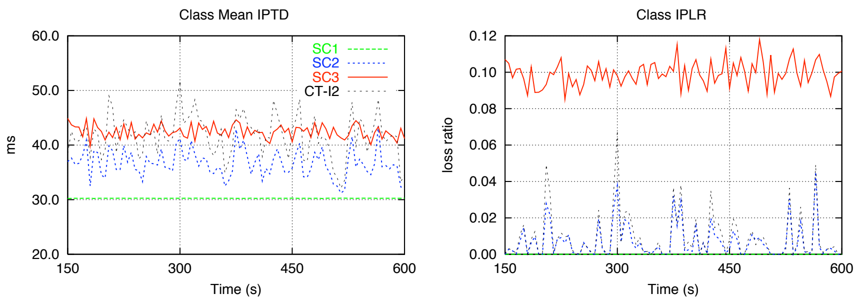

Fig. 9 Class mean IPTD and IPLR 20% of SC2 cross traffic

pancy increasing loss events and triggering the QoS control rule more frequently. However, as shown in Figure 10, the rate share of each class is well accomplished and the global utilization very high. SC3 exceeds slightly its defined rate share, taking advantage of SC1/SC2 unused bandwidth resources, due to the work conserving nature of the traffic scheduler. The packet level analysis reveals that $\%pkt\_violations$ on IPTD is 0.05 and 12.8, for 10% and 20% of cross traffic (i.e., up to 40% of the SC2 class share) respectively, and $Total\ IPLR$ is 0.005 and 0.008 (of the same order of magnitude of the defined threshold).

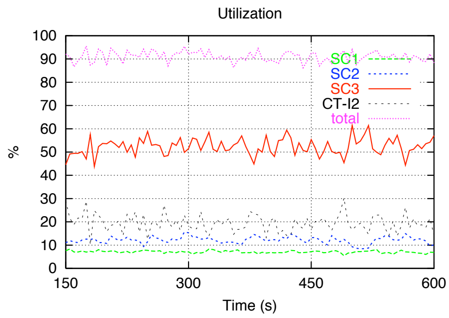

Fig. 10 (a) Utilization for 20% of SC2 cross traffic

(ii) In the presence of cross traffic from class SC1, numerous QoS violations in IPTD, ipdv and IPLR become evident and difficult to control despite the rejection indication provided by the QoS control rule. This is due to high traffic fluctuations and to the nature of the scheduling mechanism, which has defined a Max-EF-Rate for PQ treatment. In the presence of an excessive rate at C_1, unmeasured and uncontrolled by E_1, several blocking events may occur at the scheduler affecting SC1 traffic. The QoS control rule, detecting these violations, sets $AC\_Status_{\Delta t_i}$ to rejection mode. However, the effect of flows already accepted within the previous acceptance period (bounded by the rate control rule with $\beta^+_{i,E_m} = 0.85$), along with the cross traffic load, leads to QoS degradation that may span more than one Δt_i. Defining larger safety margins, as an example, for 2.5% of SC1 cross traffic (i.e., 25% of the class share), $\beta^+_{i,E_m} = 0.5$ allows an SC1 behavior without QoS violations[3]. A simple ISPs design rule for tight delay, jitter and loss control is provisioning twice the capacity of the expected aggregate peak load [C. Filsfils(2005)].

(iii) When cross traffic is from class SC3, the model behaves similarly to the concurrent traffic case. In fact, as AC for this class is not based on the rate control rule, the presence of cross and concurrent traffic is only reflected in the measured QoS. This means that SC3 IPLR is kept controlled by the QoS control rule, preserving the QoS behavior. The same occurs for the remaining service classes.

From these set of experiments, the relevance of the defined AC rules becomes evident for 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 the domain QoS levels in presence of unmeasured cross traffic. In real environ-

[3] For $\beta^+_{i,E_m} = 0.85$, the packet level analysis reveals that the $\%pkt\_violations$ for IPTD is 3.3, for ipdv is 0.36 and $Total\ IPLR$ is 0.012 (two orders of magnitude above IPLR threshold).

ments, 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. Although being encouraging on this aspect, the obtained results might be even more satisfactory when considering that a significant amount of the involved cross traffic will be sensed and controlled by other egress nodes.

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

7.2.4 Test4 - Adapting to new thresholds

This test scenario intents to illustrate the model's ability to self-adapt to distinct QoS thresholds, in particular, to control new delay and loss bounds. The traffic submitted to ingress I2 is cross traffic. As an example, Figure 11 shows that when a tighter IPTD threshold of 35ms is set for SC2, AC is effective in bringing and maintaining IPTD controlled around that value. Simultaneously, considering a new IPLR threshold of 0.05 for SC2 and SC3 (more relaxed and tight than the previous one of 10^{-3} for SC2 and 10^{-1} for SC3), it is notorious that the control strategy has been able to bring IPLR to the new value. From the figure is also evident that IPLR is more difficult to keep tightly controlled, however, a consistent behaviour around 0.05 is achieved.

7.2.5 Test5 - Testing the impact of traffic characteristics

From the analysis carried out so far, it is clear that controlling QoS and SLS utilization in a multiservice domain involves configuring and handling multiple and interrelated variables. The difficulty and complexity of such control cannot be dissociated from the statistical properties of traffic entering the network domain.

On the one hand, the choice and parameterization of a source model determine the intrinsic characteristics of each traffic flow, reflecting the way it behaves during its lifetime. On the other hand, at aggregate level, i.e., when considering multiple flows, they also determine the statistical properties of the traffic within each service class, and consequently, the challenges posed to traffic control mechanisms. For instance, low or high load estimates resulting from short-term traffic fluctuations may mislead AC decisions, while long-term properties such as LRD have proved to impact on the nature of congestion and on some AC algorithms.

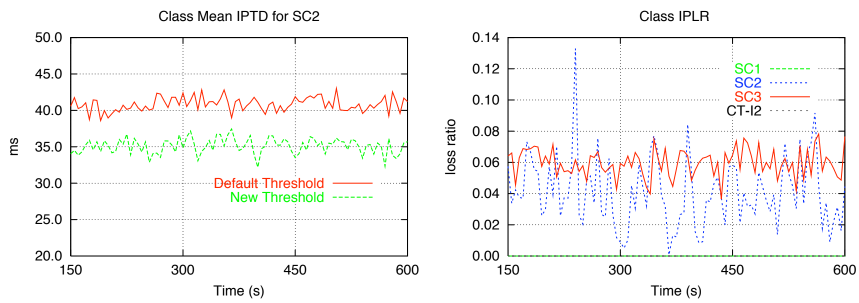

Fig. 11 IPTD and IPLR adaptation to new thresholds (20% of SC2 cross traffic);

In the present context, maintaining an AC parameterization similar to Test1 (i.e., the safety margins and thresholds), several experiments were carried out to evaluate the impact of different types of sources on the performance of the AC proposal. In this way, in addition to $EXPOO$ sources, CBR and $PAROO$ sources were included in the tests, as illustrated in Table 7. Pareto sources with a shape parameter $1 < \alpha < 2$ under aggregation allow to generate traffic exhibiting LRD.

Table 7 AC results for distinct source models

Src Type	#active_flows	%utilization	IPTD: mean;	max;	%pkts_violations	Total IPLR
CBR_{SC1}	107.3	7.3	30.2	30.6	0.0	0.0
CBR_{SC2}	116.3	44.0	31.2	38.0	0.0	0.0
FTP_{SC3}	61.6	43.0	42.7	74.9	n.a.	0.102
$EXPOO_{SC1}$	105.9	7.2	30.2	30.6	0.0	0.0
$EXPOO_{SC2}$	116.6	44.2	32.4	69.9	1.58	0.0015
FTP_{SC3}	65.9	43.4	41.7	77.2	n.a.	0.102
$PAROO_{SC1}$	104.3	7.2	30.2	30.6	0.0	0.0
$PAROO_{SC2}$	115.5	44.1	32.3	70.3	1.62	0.002
FTP_{SC3}	66.9	43.3	42.8	79.0	n.a.	0.103

The results obtained with these new source models similarly parameterized (in terms of rate, flow interarrival/holding times, and on/off periods when applicable) show that the utilization levels achieved for the distinct service classes are maintained. However, $IPTD^{max}$, *% pkt_ violations* on IPTD threshold and $Total\ IPLR$ tend to increase with traffic variability. While for CBR sources there are no packets exceeding the IPTD threshold and there is no packet loss, for $EXPOO$ and $PAROO$ sources the delay and loss behavior mentioned above is verified, in particular for SC2. Nevertheless, each class QoS commitments are generically met. In this context, the obtained results indicate that the proposed AC model exhibits good performance in handling traffic with different characteristics and burstiness.

AC fairness on concurrent flows - When analyzing the model behavior in presence of concurrent traffic with distinct flow characteristics within the same service class, the results show the model ability to adapt consistently to different conditions in the concurrent classes, adjusting the number of admitted flows according to the flows' defined rate and maintaining the global and per-class utilization levels similar to the ones obtained previously.

Table 8 AC results on fairness

Class	Source Type	#active_flows	%utilization	%pkts_violations: (IPTD ; ipdv)	Total IPLR
SC1	$EXPOO^1_{SC1}$	56.9	3.9	(0 ; 0)	0.0
	$EXPOO^1_{SC1}$	58.4	4.0	(0.16 ; 0.035)	0.0010
	$EXPOO^1_{SC1}$	52.4	3.6	(0.21 ; 0.052)	0.0012
CT-I2	$EXPOO^1_{SC1}$	58.2	3.9	(0 ; 0)	0.0
	$EXPOO^2_{SC1}$	10.4	3.9	(0.17 ; 0.026)	0.0011
	$EXPOO^3_{SC1}$	11.5	4.2	(0.24 ; 0.039)	0.0014
SC2	$EXPOO_{SC2}$	111.1	42.1	(0.19 ; n.a.)	0.0043
	$EXPOO_{SC2}$	111.3	42.0	(0.17 ; n.a.)	0.0040
	$EXPOO_{SC2}$	110.7	41.8	(0.09 ; n.a.)	0.0032
SC3	$EXPOO_{SC3}$	99.4	49.3	(n.a. ; n.a.)	0.093
	$EXPOO_{SC3}$	99.4	49.1	(n.a. ; n.a.)	0.094
	$EXPOO_{SC3}$	99.1	49.1	(n.a. ; n.a.)	0.096

The results in Table 8 illustrate this fair behavior when the concurrent class is SC1 with more demanding flow peak rates, burstiness and flow interarrival/holding times[4]. Under the new traffic conditions, the QoS behavior of SC1 shows a slight degradation. However, the *% pkt_ violations* is very low and $Total\ IPLR$ is kept well bounded within one order of magnitude above the established QoS thresholds. IPLR behavior in Δt_i is illustrated in Figure 12. The cause of QoS degradation is the higher fluctuations in the rate estimations when SC1 flows' rate is increased, irrespectively of the concurrent traffic having or not similar characteristics. The QoS degradation noticed can be avoided resorting to a higher safety margin in the SLS rate control rule for SC1. As illustrated in Table 8, the remaining service classes are not particularly affected by the new test conditions.

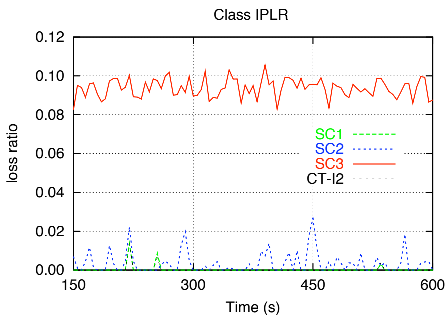

Fig. 12 IPLR behavior (CT-I2 = $EXPOO^2_{SC1}$)

7.2.6 Test6 - Testing the impact of the measurement time interval

These experiments aim at evaluating the impact of larger Δt_i on the AC model's performance, considering new measurement time intervals of 30s and 60s. This test also explore the effect of updating or not $\tilde{R}^+_{i,E_m}$ at each I_n when a new flow is admitted.

According to the obtained results, maintaining the default test conditions, the major impact of increasing Δt_i (creating consequently a longer "blind" period regarding the real network status) is to create a cyclic AC status behavior affecting the number of active flows and utilization of each class (see Figures 13 (a) and (b), and footnote 4 for graph details). The classes' QoS commitments are easily met for higher Δt_i, as result of an utilization decrease. This means that the QoS behavior of these service classes for $\Delta t_i = 30s$ and $\Delta t_i = 60s$ is better than for $\Delta t_i = 5s$, both from a measurement interval and packet level perspectives. SC3 follows similar trends to the tests using smaller Δt_i. Despite the good QoS results achieved, for larger Δt_i the AC rejection period may be excessive. The cyclic behavior exhibited in Figure 13 is also stressed by the demanding characteristics of the flow

[4] The initial configuration of SC1 sources (see Table 5) is referred as $EXPOO^1_{SC1}$ (rate = 64kbps; On = 0.96 /Off = 1.69ms (mean rate = 23kbps); flow interarrival time = 0.3s; flow holding time = 120s). $EXPOO^2_{SC1}$ (rate = 256kbps; On/Off = 500ms (mean rate = 128kbps); flow interarrival time = 0.3s; flow holding time = 90s) corresponds to a more demanding traffic source and $EXPOO^3_{SC1}$ is equivalent to $EXPOO^2_{SC1}$ varying the flow arrival and departure processes, i.e., ($EXPOO^2_{SC1}$; flow interarrival time = 0.6s; flow holding time = 120s). As mentioned, to test more demanding traffic conditions and unbalanced loads, $EXPOO^2_{SC1}$ and $EXPOO^3_{SC1}$ peak rates are around five times $EXPOO^1_{SC1}$ peak rate.

arrival process; under more moderate flow arrival conditions, that behavior tends to smooth and the evolution of active flows and utilization become more regular.

In more detail, considering SC1 and $\Delta t_i = 30s$ as an example (see Figure 13), it is visible that after each load estimation update, the system enters into a positive AC cycle with a slope that depends on flow inter-arrival. After each flow admission, each I_n will update the load estimate until detecting that the new acceptances lead to the defined utilization target. In that moment, new incoming flows start to be refused and the last estimation is kept until Δt_{i+1}, when a new load is estimated and provided. As flow departures within a time interval are not taken into account, when the new update takes place, the rate estimation at the ingress node tends to decrease abruptly. Thus, updating the rate estimates at each I_n according to the mean or peak rate of accepted flows leads to a more conservative AC as new incoming rates are considered without pondering the compensation effect of departing flows. This effect tends to be more notorious when Δt_i increases as the I_n estimation update reflecting the real network conditions, sent by the monitoring module, is provided later. Keeping rate estimates ($\tilde{R}^+_{i,E_m}$) unchanged during Δt_i irrespective of flows acceptance, explores this compensation effect but may increase overacceptance and lead to more QoS violations in all the service classes. In summary, considering the test scenarios presented previously, a smaller Δt_i may be preferable to take advantage of the good compromise among network utilization, QoS and stability. Nevertheless, dimensioning Δt_i involves establishing a trade-off between the overhead of the metrics' update process and the accuracy of capturing the real network status.

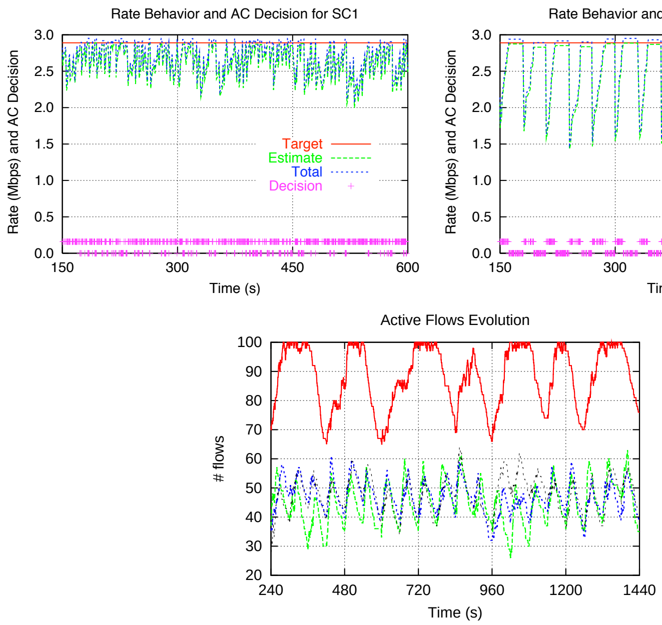

Fig. 13 (a) Influence of Δt_i on rate and AC for SC1: $\Delta t_i = 5s$ and $\Delta t_i = 30s$; (b) Active flows for $\Delta t_i = 60s$ adjusting $\tilde{R}^+_{i,E_m}$

8 Conclusions and Future Work

In this paper, we have presented and specified a service-oriented distributed AC model for managing QoS and SLSs in multiclass and multidomain environments. The model resorts to feedback from edge-to-edge on-line measurements of service-specific QoS parameters and SLS utilization to perform explicit or implicit AC. Resorting to an intuitive and expressive notation, we have specified multiservice domain entities such as service classes, upstream and downstream SLSs, and traffic flows in order to formalize generic service-dependent AC rules. These rules allow a flexible and self-adaptive control of QoS levels and SLS usage both intra and interdomain.

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. Under demanding cross traffic conditions, the relevance of the two defined AC rules became evident complementing each other to increase the domain capabilities to guarantee service commitments. The use of systematic edge-to-edge monitoring and a controlled degree of overprovisioning revealed essential design aspects contributing to achieve a simple and self-adaptive solution for managing multiple service levels.

Developing a light, effective and reliable process for computing and disseminating QoS metrics in real environments on a near real-time basis is an important aspect left for future work.

References

[Babiarz et al.(2006)] J. Babiarz, K. Chan, and F. Baker. Configuration Guidelines for Diffserv Service Classes. RFC 4594, August 2006. URL `http://www.ietf.org/rfc/rfc4594.txt`. Informational.

[Breslau and Jamin(2000)] L. Breslau and S. Jamin. Comments on the Performance of Measurement-Based Admission Control Algorithms. In *IEEE INFOCOM'00*, March 2000.

[Breslau et al.(2000)] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint Admission Control: Architectural Issues and Performance. In *ACM SIGCOMM'00*, 2000.

[C. Filsfils(2005)] J. Evans C. Filsfils. Deploying Diffserv in Backbone Networks for Tight SLA Control. *IEEE Internet Computing*, 9(1):58–65, Jan.-Feb. 2005. ISSN 1089-7801.

[Cetinkaya et al.(2001)] C. Cetinkaya, V. Kanodia, and E. Knightly. Scalable Services via Egress Admission Control. *IEEE Transactions on Multimedia*, 3(1):69–81, March 2001.

[D. Miras et al.(2002)] D. Miras et al. A Survey of Network QoS Needs of Advanced Internet Applications. Internet2 Working Document, November 2002.

[Duan et al.(2004)] 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*, 15(2):167–182, February 2004.

[Elek et al.(2000)] V. Elek, G. Karlsson, and R. Rnngren. Admission Control Based on End-to-End Measurements. In *IEEE INFOCOM'00*, 2000.

[Fredj et al.(2001)] S. Fredj, S. Oueslati-Boulahia, and J. Roberts. Measurement-based Admission Control for Elastic Traffic. In *17th International Teletraffic Congress*, June 2001.

[Georgatsos et al.(2004)] 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)*, September 2004.

[Goderis et al.(2003)] D. Goderis, D. Griffin, C. Jacquenet, and G. Pavlou. Attributes of a Service Level Specification (SLS) Template. IETF draft: draft-tequila-sls-03.txt (work in progress), October 2003.

[Jamin et al.(1997)] 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*, pages 56–70, February 1997.

[Lima et al.(2005)] Solange Rito Lima, Paulo Carvalho, and Vasco Freitas. Self-adaptive Distributed Management of QoS and SLSs in Multiservice Networks. In *IEEE/IFIP International Conference on Integrated Management (IM 2005)*, Nice, France, May 2005. IEEE Press.

[Lima et al.(2007a)] Solange Rito Lima, Paulo Carvalho, and Vasco Freitas. Admission Control in Multiservice IP Networks: Architectural Issues and Trends. *IEEE Computer Communications Magazine*, 45(4): 114–121, April 2007a.

[Lima et al.(2007b)] Solange Rito Lima, Pedro Sousa, and Paulo Carvalho. Enhancing QoS metrics estimation in multiclass networks. In *SAC '07: Proceedings of the 2007 ACM symposium on Applied computing*, pages 227–231, New York, NY, USA, 2007b. ACM Press. ISBN 1-59593-480-4. doi: http://doi.acm.org/10.1145/1244002.1244060.

[Morand et al.(2004)] P. Morand, M. Boucadair, P. Levis, R. Egan, H. Asgari, D. Griffin, J. Griem, J. Spencer, P. Trimintzios, M. Howarth, N. Wang, P. Flegkas, K. Ho, S. Georgoulas, G. Pavlou, P. Georgatsos, and T. Damilatis. Mescal D1.2 - Initial Specification of Protocols and Algorithms for Inter-domain SLS Management and Traffic Engineering for QoS-based IP Service Delivery and their Test Requirements. Mescal Project IST-2001-37961, January 2004.

[Mortier et al.(2000)] R. Mortier, I. Pratt, C. Clark, and S. Crosby. Implicit Admission Control. *IEEE Journal on Selected Areas in Communication*, 18(12):2629–2639, December 2000.

[Seitz(2003)] Neal Seitz. ITU-T QoS Standards for IP-Based Networks. *IEEE Communications Magazine*, 41(5), June 2003.

[Sevasti and Campanella(2001)] A. Sevasti and M. Campanella. Geant D9.1 - Service Level Agreements Specification for IP Premium Service. Geant and Sequin Projects, October 2001.

[Stoica and Zhang(1999)] I. Stoica and Hui Zhang. Providing Guaranteed Services Without Per Flow Management. In *ACM SIGCOMM'99*, October 1999.

[Teitelbaum et al.(1999)] B. Teitelbaum, S. Hares, L. Dunn, R. Neilson V. Narayan, and F. Reichmeyer. Internet2 QBone: building a testbed for differentiated services. *IEEE Network*, 13(5):8–16, "Sep./Oct." 1999.

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