Title: **Distributed Admission Control for QoS and SLS Management**

Authors: Solange Lima[1], Paulo Carvalho, Vasco Freitas

Affiliation University of Minho
Department of Informatics
4710-057 Braga
Portugal

Tel: +351253604432

Fax: +351253604471

E-mail addresses: {solange,pmc,vf}@di.uminho.pt

[1] Corresponding Author: Solange Lima

Distributed Admission Control for QoS and SLS Management

This article proposes a distributed admission control (AC) model based on on-line monitoring to manage the quality of Internet multimedia services and Service Level Specifications (SLSs). The AC strategy covers intra and inter-domain operation, controls quality-of-service (QoS) without adding significant complexity to the network control plane and involves only edge nodes. While ingress nodes perform implicit or explicit AC resorting to service-oriented rules for SLS and QoS parameters control, egress nodes collect service metrics providing them as inputs for AC. The end-to-end approach is viewed as a cumulative and repetitive process of AC and available service computation. We evaluate the AC criterion as regards its ability to ensure service commitments while achieving high network utilization. The results show that the proposed model provides a good compromise between simplicity, service guarantee levels and network usage, even for services with strict QoS requirements. Crucial aspects of the model interrelated areas and implementation key points are also discussed and evaluated.

Keywords: Traffic Control; Measurement-based Admission Control; Quality of Service; Service Level Agreement; Service Level Specification; QoS Monitoring; Class-based Networks; Diffserv Architecture

Running head: Distributed Admission Control for QoS and SLS Management

1 INTRODUCTION

The support of heterogeneous applications and services in the Internet with distinct quality-of-service (QoS) requirements is behind the class-of-service (CoS) network paradigm. In CoS networks, where Diffserv architecture is a reference model[1, 2], traffic flows are aggregated in a limited number of service classes according to the QoS objectives to fulfill. Controlling the admission of flows sharing a class allows to: (i) avoid over-allocation of existing network resources; (ii) avoid new flows from impairing flows already accepted; (iii) fulfill service level agreements and specifications (SLA/SLS); and (iv) prevent instability and assure QoS. However, the complexity introduced by admission control (AC) in the network control plane has to be carefully assessed as Internet traffic is highly dynamic and not every application has strict QoS requirements.

Despite the existing proposals (discussed in Section 2.2), achieving a generic, yet feasible and light, AC model for multi-service CoS networks, able to operate both intra-domain and end-to-end, is still an open issue. A further step in pursuing this objective is proposed in [3] where a new and encompassing service-oriented distributed AC model is presented. The model resorts to edge-to-edge on-line monitoring for the control of both QoS levels in a domain and the sharing of existing Service Level Specifications (SLSs) between domains. The AC decision process is distributed among the domain ingress nodes, being affected by feedback from measurements performed at egress nodes. The model end-to-end operation is treated as a per domain repetitive process of AC and available service computation.

This paper details the main components of this AC model focusing on its practical implementation. Topics such as the definition of service classes and the specification of the rules which drive AC decisions for each class are devised taking into account the service and traffic characteristics, the network resources and the QoS guarantee levels to be provided. On-line monitoring is used for SLS auditing and QoS control in the domain and for providing the necessary inputs for AC. For this purpose, several QoS and performance metrics are defined and evaluated resorting to proper estimation methodologies and mechanisms. The tuning and performance of the AC model is evaluated as regards the underlying measurement process and effectiveness of AC rules.

The remaining of this document is organized as follows: the AC problem statement focusing on its perspectives and current approaches is reviewed in Section 2; a description of the AC model, including its components and operational details are summarized in Section 3; the model implementation is detailed in Section 4; the simulation platform which supports the different test scenarios is described in Section 5; Section 6 is devoted to the discussion of the results, and the conclusions of the present work are summarized in Section 7.

2 THE ADMISSION CONTROL PROBLEM

2.1 AC Perspectives

According to [3], two AC perspectives can be considered when an SLS is taken as reference: (i) *flow AC* ensures that the admitted flows from a customer are within the capacity of the contracted SLS; or (ii) *SLS AC* ensures that the accepted SLSs for a service type can be honored through proper configuration and provisioning (see Fig. 1).

Although these are distinct AC perspectives, they follow similar principles. Whereas flow AC is based on the traffic profile and QoS objectives of a flow, SLS AC is based on the aggregate traffic profile and QoS objectives of the SLS. In fact, the semantic of the process is equivalent, only changing the granularity upon which the decision is made. Therefore, the proposed model, described here for flow AC, can be applied both to flow AC and SLS AC, with minor changes.

2.2 AC Approaches: A Service Oriented Overview

When defining an AC strategy a trade-off between the service assurance level and network control complexity needs to be carefully established. Depending on the QoS guarantees and predictability required, more or less complex AC strategies can be used, with strict or relaxed control of network resources and QoS parameters. The type and number of network nodes (e.g. edge, core, central entities) involved directly or controlled by the AC process can also vary, affecting the solution complexity.

Overprovisioning is currently a common solution to provide QoS guarantees in network backbones, avoiding or reducing the network control complexity. Although for some ISPs overprovisioning is an attainable solution, it leads to poor resource utilization and sometimes is not available or too expensive. So, further control has to be in place in order to honor QoS requirements[1].

Associated with CoS-based architectures, such as Diffserv, several AC approaches have been defined, with the

[1] In our view, despite having additional control mechanisms, a certain degree of overprovisioning is convenient and recommended in order to relax and simplify the AC process.

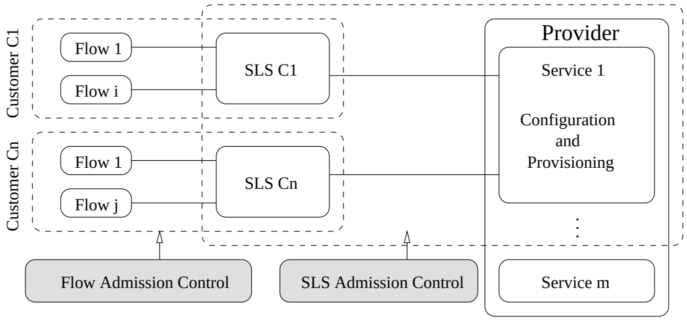

Figure 1. Flow AC and SLS AC

common aim of avoiding per-flow state information in the core nodes due to scalability reasons. Some proposals suggest the use of central entities for AC and resource management (bandwidth brokers) [4, 5, 6, 7]. However, the well-known problems of centralization led to several decentralized AC approaches. To provide quantitative service guarantees (e.g. for hard real-time traffic) current AC proposals need to control the state and the load of traffic aggregates in the core nodes [4, 8, 6, 9], or even perform AC in these nodes [8, 9]. These solutions tend to require significant network state information and, in many cases, changes in all network nodes. Furthermore, as they are closely tied to network topology and routing, their complexity increases with the network dynamics.

Providing qualitative service guarantees (e.g. for soft real-time traffic) leads to reduced control information and overhead, but eventually to QoS degradation. Obtaining a good compromise between efficient resources utilization and QoS guarantee is a major challenge. In this context, measurement-based AC (MBAC) solutions have deserved special attention. Initially performed in all network nodes, recent studies suggest that AC decisions should be carried out only at the edges (end-systems or edge routers), using either active or passive measurement strategies of network load and/or QoS parameters [10, 11, 12]. Despite not requiring changes in the network, end-to-end MBAC (EMBAC) increases the initial latency and network load as probing is carried out on a per application basis.

The need to control elastic traffic, for more efficient network utilization, has also been discussed and implicit AC strategies (without explicit signaling between the application and the network) have been defined [13, 14]. Conversely, AC approaches for streaming applications commonly use signaling between the application and the network where, upon a traffic profile and QoS objectives description, the network sends an explicit acceptance/rejection message.

A complete survey comparing the main features and limitations of current AC strategies is available in [15].

Facing the related work mentioned earlier, the proposed AC strategy, described in the following section, brings new insights on how to perform a distributed lightweight AC in multi-service environments. Section 3.3 highlights the AC model strengths, while Section **??** debates the model major hurdles and possible solutions.

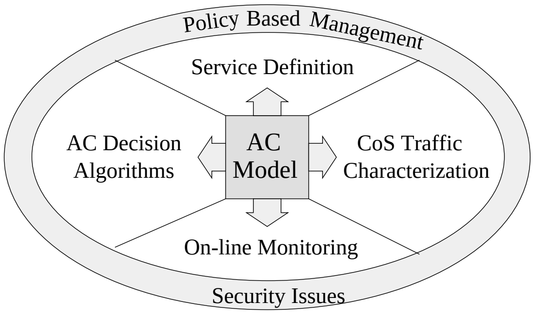

Figure 2. Model Areas

3 THE ADMISSION CONTROL MODEL

The proposed AC model considers: (i) the control of distinct network services and assurance levels, to handle application with different QoS requirements and traffic profiles; (ii) the intra-domain and end-to-end operation, controlling both the QoS levels in a domain and the sharing of the existing SLS between domains to fulfill the applications end-to-end QoS requirements. The model design is driven by simplicity, and easiness of deployment and integration in the Internet. The scalability and flexibility of the proposal as regards technological, service and application evolution goals have also been considered. These goals are relevant when deploying the model in a large scale across multiple administrative domains with distinct QoS solutions.

3.1 Model Interrelated Areas

Taking advantage of the consensual need for on-line QoS and SLS monitoring in CoS networks, the AC model makes admission decisions resorting to edge-to-edge measures of relevant QoS parameters for each service type and to SLS utilization. Thus, in this process, four main areas are interrelated (see Fig. 2) [3]: (i) *service definition*, which involves the definition of the parameters and semantic of SLSs and of basic services adapted to different application types; (ii) *on-line monitoring*, which keeps track of QoS and SLS status in the domain; (iii) *AC decision criteria*, which involves the establishment of service dependent AC equations; and (iv) *CoS traffic characterization*, which provides the knowledge of the statistical properties of the classes in the domain as a result of aggregation [16]. The use of policy-based network management and security considerations were left for further

Table I. Service Level Agreement (SLA) Template

Administrative Information		
Administrative entities		Contractual parties involved
Description of service		Description of service behavior
Validity		Contract validity period
Pricing/Tariffs		Pricing and tariffs of the service
Helpdesk info/Trouble tickets		Customer support actions
Monitoring/Accounting reports		Monitoring/accounting rules
Response time to changes		Time for enforcement of changes
Other		Other rules: e.g. provisioning
SLS		
Scope of the service	- Ingress interfaces - Egress interfaces	Boundaries of the region over which the service will be enforced
Traffic classifying rules	- MultiField criterion - DSCP or ToS Precedence	Packet fields used to identify a traffic flow or aggregate
Traffic Cond. rules	- Conformance algorithm - Conformance parameters - Treatment on excess	Information used to identify in-profile and out-of-profile traffic and corresponding treatment
Expected QoS parameters	- Delay, jitter, loss,... - Qualitative objectives - Quantitative objectives	Expected QoS of the conforming traffic stream in the Scope region
Service Reliability	- Mean downtime - Time to repair,...	Expected service reliability
Service scheduling	- Start/End time	Service time availability
Others	- Route, security, ...	Left for future study

study. Before describing the model operation, relevant aspects of the model areas will be discussed.

3.1.1 *Service Definition*

A Service Level Agreement (SLA) is defined as a contract between a customer and a service provider or between service providers, specifying administrative and technical service information. The technical part of an SLA, called Service-Level Specification (SLS), defines the expected service level, QoS-related parameters and traffic control issues[2].

The definition of SLSs, apart from being a key aspect for QoS provisioning, provides a valuable input for AC, in special, when admission spans multi-domains. Therefore, defining a standard set of SLS parameters and semantics is crucial for ensuring end-to-end QoS delivery and for simplifying negotiations. Several working groups are committed to SLS definition [17, 18, 19, 20, 21] and management [17, 18, 22, 23, 24, 25, 26, 27, 28]. Taking these inputs into account, a possible SLA template including relevant parameters and their typical contents is

[2] An SLA may include multiple SLSs, however, an SLS is usually related to a service class usage. In the context of AC, as the relevant part of an SLA is the SLS, from now on we refer uniquely to SLSs and assume that each SLS is encompassed in the corresponding service class.

Table II. Upper Bounds of QoS Parameters for some Applications

ITU-T Classes	Class 0	Class 1	Class 2	Class U	
Applications	Real-time	VoIP/Interact.	Non-Interact.	WWW/Free Serv.	Video (VHS)
IPTD	150 ms	400 ms	1 s	Undefined	400 ms
IPDV	50 ms	50 ms	1 s	Undefined	17 ms
IPLR	10^{-3}	10^{-3}	10^{-3}	Undefined	10^{-5}
IPER	10^{-4}	10^{-4}	10^{-4}	Undefined	10^{-4}

defined in Table I. Although a large combination of QoS, performance and reliability parameters is possible, service providers will offer a limited number of services. To instantiate the SLS template in quantitative and qualitative standard services adapted to different application types is, in fact, the major objective. To fulfill this, substantial work has been done to identify the relevant QoS parameters and the perceived quantitative quality of applications [29, 30]. Table II summarizes commonly acceptable upper bounds of QoS parameters for common applications and services. These inputs and Diffserv Per-Domain Behavior (PDB) definitions are used to identify the services and corresponding QoS parameters to test the AC model.

3.1.2 *Monitoring Issues*

The problematic of monitoring involves the definition of metrics, measurement methodologies and timing decisions. ITU-T work on QoS in IP networks and particularly the IP performance metrics (IPPM) working group within IETF have defined a set of standard QoS and performance metrics and have proposed measuring methodologies for them [31, 32, 30]. Several tools useful for measuring the SLS metrics have also been developed and tested [29, 33].

Metric definition issues - The definition of metrics requires identifying relevant parameters for each service type and corresponding statistics. As mentioned earlier, IPPM aims at developing a set of standard metrics providing unbiased quantitative measures of quality, performance and reliability of operational Internet data delivery services. Defining a metric, identifying its type (analytical or empirical), its composition (in spatial and temporal terms) and its corresponding instances (singleton or sample metric) are topics to be addressed concerning the defined parameters [32].

Measurement methodology issues - A measurement methodology can be either passive, active or combination thereof. Passive measurements are carried out on existing traffic and are particularly suitable for troubleshooting; active measurements inject extra traffic (probing) in the network for measurement purposes, allowing to check QoS and SLS objectives in a more straightforward way. Probing brings an additional advantage when measuring edge-to-edge performance and QoS. As specific packets are injected in the network containing timestamping and sequencing data, delay and loss estimations are simplified. Obtaining these estimates combining link-by-link measures is not an efficient and easy solution. As probing is an intrusive process, its impact on the network load needs to be minimized. However, small traffic volumes may be enough to obtain meaningful measures [34, 35] and, in the proposed model, in-band probing is used per class and not per application, which reduces the overhead, being a clear advantage over other EMBAC approaches.

Timing issues - Timing decisions deal with the synchronization between measurement points and the periodicity of measurements. For synchronization purposes well-known solutions based on Network Time Protocol (NTP) and Global Positioning System (GPS) are usually used. Periodicity decisions should consider that a small time granularity increases the metric computation and dissemination overhead, and eventually leads to an excessive reactivity to short-time traffic fluctuations, whereas a sparse granularity may lead to out-of-date network state information. Depending on the measured parameter and metric purposes, timing definitions can vary significantly. The operating timescales for AC processes, running from few seconds to minutes, are not the most critical.

3.1.3 *AC Decision Algorithms*

In any AC strategy the admission criterion plays a crucial role as regards service guarantees and network efficiency. Usually AC criteria are parameter-based, measurement-based or follow an hybrid scheme combining both. Parameter-based AC algorithms take into account the network resources already in use by accepted flows and the resources the new flow will consume, according to its explicit traffic descriptor. These descriptors allow to establish upper bounds on the traffic generated by a source. When AC just takes this information, QoS conformance can be easily achieved, leading to acceptable network utilization when the flows are smooth, however, utilization is low for bursty traffic. Thus, parameter-based AC algorithms tend to be conservative and oriented for flows re-

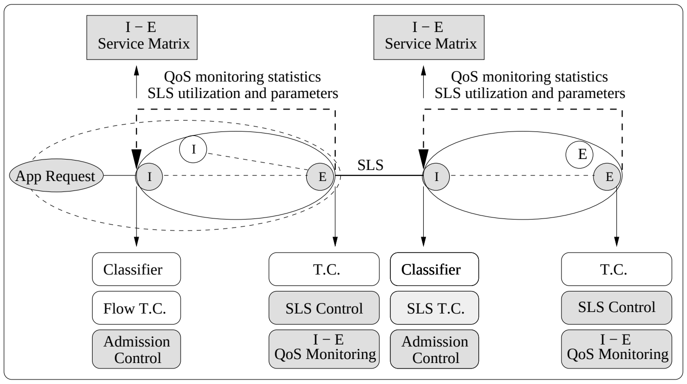

Figure 3. Domain Activities Location

quiring a guaranteed service. Measurement-based AC (MBAC) algorithms take into account measures reflecting the existing flows impact on the network load and/or QoS before deciding over a new admission. While rate-based AC rules are the most common [36], estimates of delay, loss, Explicit Congestion Notification (ECN) marks are also used. These algorithms are less conservative, taking advantage of statistical multiplexing of traffic to increase network utilization, at an eventual cost in QoS degradation. In this way, MBAC is more suitable for flows requiring a predictive service. Examples of parameter-based and measurement-based algorithms are defined and compared in [37, 10].

3.1.4 *CoS Traffic Characterization*

The statistical properties of traffic aggregated into classes [16] need to be considered so that proper thresholds or safety margins to AC can be established. For instance, classes which exhibit long-range dependence may need large safety margins as this property has a significant impact on queuing behavior and on the nature of congestion, leading to unexpected QoS degradation. Knowing the usual per class traffic volumes is also relevant for traffic forecasting and provisioning.

3.2 Model Operation Description

Before describing the model operation some concepts have to be defined to clarify its description. In a transit domain, our view of an SLS varies according to whether a client or service provider perspective is taken. In upstream SLSs, the domain acts as a service provider to the previous domain; in downstream SLSs the domain acts as a client for the next domain. A set of upstreams SLSs with identical requirements share a service class in the domain. Whenever an upstream SLS requires a distinct specific service, in case of acceptance, a new service class has to be configured in the domain.

The main tasks involved in the proposed AC model and their location are illustrated in Fig. 3. Apart from the usual classification and traffic conditioning (TC) tasks (in white boxes) present in CoS networks, ingress routers perform explicit or implicit AC (see Section 3.2.1), depending on the application type and corresponding traffic class. Egress routers perform on-line QoS monitoring and SLS control. *Ingress-Egress QoS Monitoring* measures relevant parameters for each service (service metrics) using appropriate time-scales and methodologies (see Section 4.4). The resulting measures reflect the service available from each ingress. *SLS Control* 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. QoS monitoring statistics, SLS utilization and associated parameters 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 is carried out periodically or when a metric value or its variation exceeds a limit or the SLS utilization exceeds a safety threshold.

3.2.1 *Explicit and Implicit AC*

As the proposed model is multi-service, explicit and implicit AC can be in place depending on the application or service characteristics. Explicit flow AC is oriented to applications able to signal the network with their traffic profile and QoS objectives, e.g. streaming applications. In this case, the AC decision requires two initial verifications (see Fig. 4): (i) *SLS Utilization Control* checks if the downstream SLS can accommodate the traffic profile of the new flow. Verifying if the upstream SLS can accommodate the new flow profile is optional. As the previous do-

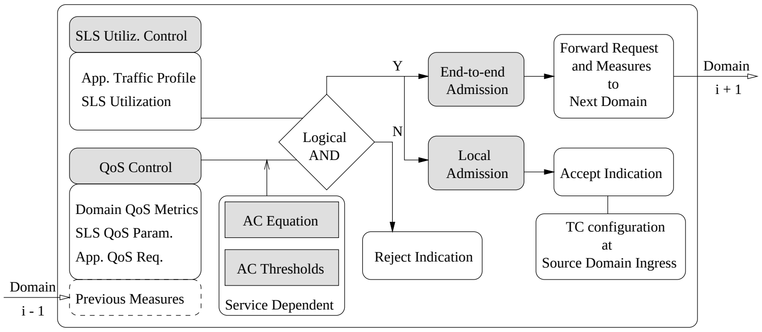

Figure 4. Admission Control Criterion

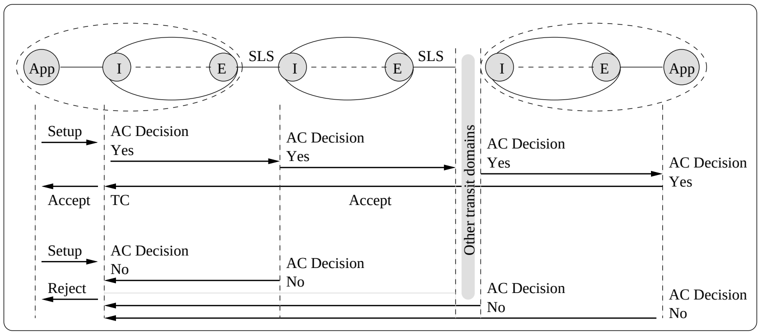

Figure 5. End-to-end Admission Control Procedure

main has controlled the corresponding downstream SLS traffic load, the current domain can control the upstream SLS using a simple TC mechanism. (ii) *QoS Control* checks if, for the corresponding egress node and service, the domain QoS metrics, the SLS QoS parameters agreed with the downstream domain[3] and the previous measures (if any) fulfill the application QoS requirements.

Each AC decision is based on a service dependent AC equation and thresholds defined to ensure specific service guarantees (explained in Section 4.2). For each class, admission thresholds must be stricter than the class QoS objectives, which in turn, must be stricter than the QoS requirements of all accepted flows, if specified.

When a flow is accepted in the domain, the notification may be generated either locally (local admission) or remotely (end-to-end admission). The *end-to-end case* is viewed as a repetitive and cumulative process of admission control and available service computation, performed at ingress nodes (see Fig. 5). At each domain the 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. The last AC decision is taken by the receiver in its own domain, which is viewed as a local admission. When a rejection occurs, the source is notified directly from the rejection point. Acceptance notifications can also be used to configure TC at the source domain ingress router. This solution leads to a generic AC model, which can be applied both to source and transit domains.

Implicit flow AC, oriented to applications which do not use signaling and in particular to elastic applications, use implicit detection of flows [13]. This type of AC, likely to be implemented only in the source domain, will be restricted to SLS information and QoS monitoring. Two possible implicit reject actions are (i) SYN packets discarding or (ii) simply packet discarding based on flow accept/reject tables [13].

3.3 Model Strengths

The proposed model has important features that should be highlighted, such as: (i) only edge nodes are involved, i.e., the network core is treated as a black-box; (ii) the state information is service/SLS and Ingress-Egress based,

[3] While domain QoS metrics are always checked, when the destination of the flow request is inside the domain, SLS verification is not mandatory. However defining intra-domain SLSs will turn the AC process generic and independent of the destination's location.

which is particularly suitable for SLS auditing; (iii) per-flow state information is only kept at the source domain ingress router for TC, while other downstream domains maintain the TC based on the SLS traffic profile, as usual; (iv) the signaling process for intra and inter domain operation is simple and does not imply soft/hard state behavior and symmetric routing paths[4]; (v) the systematic use of on-line monitoring for traffic load and QoS estimation, while allowing an active service management avoids the common per application intrusive traffic and the initial latency of edge-to-edge measurement-based AC approaches. Furthermore, the effect of cross-traffic and other internally generated traffic (e.g. routing, management, multicast traffic) is implicitly taken into account. The AC model also allows controlling simultaneously different services types, QoS parameters and SLSs, while being distributed. Usually, this type of control is only covered in centralized AC approaches.

3.4 Major Hurdles and Solutions

A distributed and edge-to-edge monitoring AC model, although simplifying the network control plane, poses additional challenges to parameter estimation and control. For instance, controlling the available bandwidth or capacity in a single link or node-by-node is a fairly simple process. However, controlling it edge-to-edge is not straightforward as the available bandwidth, available capacity or network bottlenecks vary dynamically over time. Methodologies and tools for estimating the available path capacity and available bandwidth are described in [29, 38]. An additional aspect of concern is how to foresee the impact a new flow acceptance will have on the existing network resources and current QoS parameters. This can be accommodated defining and tuning proper parameters' safety margins in order to keep AC model simple.

AC, being distributed, may involve multiple ingress routers taking concurrent AC decisions. Therefore, dealing with concurrency is a key aspect as otherwise AC may lead to over or false acceptance. This problem can be reduced resorting to the definition of per service safety margins to absorb load fluctuations, complemented by a token system to limit simultaneous AC decisions.

[4] There is no guarantee that the path used for the flow data is the same used for the flow request. This may not be problematic providing that the new path is established maintaining the same QoS characteristics. In fact, the new metrics will reflect the load variation and AC will act accordingly.

4 MODEL IMPLEMENTATION

This section details the model implementation aspects [39] regarding the main areas described in Section 3.1.

4.1 Definition of Service Classes and SLS

Apart from a wide range of mechanisms allowing to handle traffic aggregates according to a specific behavior, a differentiated services architecture needs to be supported by an adequate strategy of traffic classification. Due to economical and technical reasons the definition of a traffic classification criterion is a subjective task. For instance, client A may be willing to pay more than client B to obtain a better service quality, for identical traffic types. Moreover, when a criterion is based on TCP-UDP/IP packet headers, both packet fragmentation, packet encryption and the use of negotiated or unregistered application ports difficult classification. Most of the criteria suggest distinct classes for UDP and TCP traffic so that non-reactive and reactive applications do not compete for the same resources. Some go further suggesting that the duration of flows, the transmission rate and packet size characteristics should also be considered [40]. A classification method based on QoS application requirements such as delay or loss sensitivity is also common.

Taking these inputs into consideration, and as initial policy, TCP and UDP traffic are treated separately, being UDP traffic further divided according to the QoS requirements stringiness. As a result, three initial service classes were defined. Service Class 1 (SC1), being supported by the Expedited Forwarding (EF) Per-Hop Behavior (PHB) [41], provides a very high QoS performance service guarantee. This service is oriented to streaming applications imposing hard real-time constraints. Due to the high priority treatment this class requires in each network node, which may starve low priority classes, the access to the corresponding service is tightly controlled. In our service model, SC1 takes a reference value of 10% of the (bottleneck) link capacity as an upper bound for the admissible traffic load. For this service, the AC criterion will follow a conservative schema, with TC giving a severe treatment on excess traffic (see Section 4.2). Service Class 2 (SC2), being supported by the Assured Forwarding (AF) PHB [42, 2], provides a predictive type of service with low delay, low loss and minimum bandwidth guarantee. This service is oriented to a range of streaming applications with soft real-time constrains. For this

Table III. Definition of Service Classes

Service Class	Service Level	Traffic type	PHB	AC	Policing	Scheduling
SC1	guaranteed	UDP (hard RT)	EF	explicit & conservative	drop on excess	strict priority
SC2	predictive	UDP (soft RT)	AF	explicit & flexible	3 color marker	WRR
SC3	best-effort	TCP	BE	implicit & relaxed	3 color marker	WRR

service, the AC criterion will be less conservative, taking more advantage of statistical multiplexing. TC will act on non-conformable traffic following a three-color marking scheme [43]. In a first set of experiments, only AF1x is considered. In SC1 and SC2 the AC criteria, apart from considering the flow traffic profile description, take into account measures of network loss and delay of those classes given their relevance for the supported streaming applications. Service Class 3 (SC3) provides the common best-effort service. Adaptive TCP applications are generically included in this class. Detailed classification rules for TCP differentiation, e.g. handling short and long lived connections separately, will be considered in the future, taking the remaining AF classes. For this service, the AC criterion will be implicit and relaxed. Giving the nature of TCP traffic and associated congestion control, loss will be considered as the most significant parameter under control. As for SC2, TC uses a three-color marker for policing. The service classes described above, and summarized in Table III, are implemented resorting to a class-based queuing differentiation mechanism, where each queue is served according to a work-conserving priority weighted round-robin scheduling mechanism.

4.2 Admission Control Criteria

The definition of an admission control criterion involves establishing the rules which determine flows acceptance or rejection. In the proposed model, the AC criteria are generically measurement-based controlling both the QoS in the domain and the downstream SLS utilization, which leads to the specification of two types of rules: (i) rate-based SLS control rules; and (ii) QoS parameters control rules.

4.2.1 *Rate-based SLS Control Rules*

Assuming $I = \{I_1, I_2, ..., I_n\}$ as the set of ingress nodes and $E = \{E_1, E_2, ..., E_m\}$ as the set of egress nodes in a domain, for each egress $E_r \in E$ with $1 \leq r \leq m$, one or more SLSs can be in place, one per service type and per downstream domain. At this point, it is assumed a single mapping between a class of service within the domain and a downstream SLS for the corresponding service type. As each SLS has a specified negotiated rate, a rate based Measure-Sum (MS) algorithm is applied to control SLS utilization. For each ingress $I_r \in I$ with $1 \leq r \leq n$, Eq. (1), which takes both rate estimates and flow traffic description, is used to verify if a new flow can be admitted,

$$\rho_s + r_j \leq \beta_s R_s \quad (1) \qquad\qquad \rho_s = \sum_{k=1}^{n} \rho_{i,k} \quad (2)$$

In Eq. (1) ρ_s is the current measured load or estimated rate of flows using SLS_s, r_j is the rate specified for flow j, $0 < \beta_s \leq 1$ is the utilization target for the SLS and R_s is the rate defined in SLS_s. For the corresponding egress router $E_r \in E$, the rate estimation process considers all the ingress-egress rate estimates for class i going through E_r, as expressed by Eq.(2).

4.2.2 *QoS Parameters Control Rules*

For the domain QoS control, depending on the traffic class, both the QoS parameters under control and their corresponding targets can vary. Therefore, this emphasizes the need for specific per class AC criterion. Depending on each service class commitments, the statistical properties of traffic and levels of overprovisioning, each AC criterion uses pre-defined QoS thresholds. A new flow is accepted if

$$\{\forall p \in P_i : \tilde{p}_p \leq t_p\} \quad (3) \qquad\qquad t_p = \beta_p p_p \quad (4)$$

i.e. the controlled parameters $P_i = \{p_1, p_2, ..., p_p\}$ of class i checked against the corresponding pre-defined threshold t_p determine an acceptance status for Δt_i, which remains unchanged during this interval. Each t_p is also affected by a safety margin to the parameter bound $0 < \beta_p \leq 1$ as shown in Eq.(4). $\tilde{p}_p$ represents the measured value of p_p for Δt_i. Tuning these limits, making them useful and realistic indicators of the overall QoS status is a fundamental aspect for AC, as shown in Section 6.2.

According to the service definition provided in Section 4.1, AC for services SC1 and SC2 uses the two types of rules defined above. For SC1, a more conservative criterion is taken, considering the worst-case scenario (flow peak rates, concurrent AC taking place at other ingress nodes and optimistic measures) larger safety margins and tighter thresholds. Accordingly to Table IV, the controlled QoS parameters for SC1 are IPTD (similar to One Way Delay), ipdv and IPLR, whose definitions are provided in Table V. AC for SC2 takes the flow mean rate for SLS

Table IV. Admission Control Criteria

	Flow Inputs		Network Inputs	SLS Util. Control		QoS Parameter Control	
Class	T.Desc.	QoS	Measures	Parameter	Method	Parameter	Method
SC1	peak rate	if any	load, IPTD, ipdv, IPLR	rate	MS	IPTD, ipdv, IPLR	thresh.
SC2	mean rate	if any	load, IPTD, IPLR	rate	MS	IPTD, IPLR	thresh.
SC3	n.a.	n.a.	load, IPLR	rate	thresh.	IPLR	thresh.

control and IPTD and IPLR for QoS control. For SC3, oriented to TCP traffic, AC is implicit and decisions are made based essentially on IPLR control. Recall that, due to the nature of TCP traffic, where a flow has not a pre-defined rate, Eq. (1) is applied as a threshold for the estimate rate. The estimation mechanisms for the parameters under control and the time granularity used in the estimation are discussed in Section 4.4.

4.3 Extending AC to End-to-End

Apart from the AC decision process carried out at each domain ingress router based on the domain QoS and SLS status (see Figs. 4 and 5), whenever a flow AC request specifies its end-to-end QoS requirements, QoS parameter control rules need to be extended to accommodate an additional verification. More precisely, verifying if each flow QoS parameter target can be satisfied involves considering the corresponding QoS parameter bound in the domain and the cumulative value computed so far. The way these values are handled depend on the type of parameter being controlled. For instance, delay parameters are addictive whereas loss ratio parameters are multiplicative.

4.4 On-line Monitoring

In order to achieve flexibility and portability, the AC and the monitoring modules are independent. Although being decoupled from the AC process itself, monitoring is a critical component of the model as it is used for active QoS and SLS control. For active QoS control, monitoring has to be on-line [34, 24] in order to provide feedback reflecting the current network conditions so that proper management decisions can be made in useful time. In our study, the objective of on-line monitoring is twofold. First, it allows SLS auditing in the domain [44]. Second, it provides inputs for the AC decision module. Apart from flow traffic profile and/or QoS requirements information, MBAC algorithms require a realistic view of network status and performance, therefore, several parameters need

Table V. Controlled QoS Parameters

Rate Parameters	
Throughput ρ (bps)	$\rho_{i,\Delta t_i} = (\sum bits\_recv_i)_{\Delta t_i} / \Delta t_i$
Utilization U (%)	$U_{i,\Delta t_i} = \rho_{i,\Delta t_i} / C$
Delay Parameters (ms)	
IP Transfer Delay ($IPTD$)	$IPTD_{i,pkt} = (t_{E_m,pkt} - t_{I_n,pkt})$
Mean IPTD ($\overline{IPTD}$)	$\overline{IPTD}_{i,\Delta t_i} = (\sum IPTD_{i,pkt} / \sum pkts\_recv_i)_{\Delta t_i}$
Inst. Packet Delay Var. ($ipdv$)	$ipdv_{i,2pkt} = (IPTD_{i,pkt} - IPTD_{i,pkt-1})$
Mean ipdv ($\overline{ipdv}$)	$\overline{ipdv}_{i,\Delta t_i} = (\sum \lvert ipdv_{i,2pkt} \rvert / \sum pkts\_recv_i)_{\Delta t_i}$
Maximum IPTD ($IPTD^{max}$)	$IPTD^{max}_{i,\Delta t_i} = max(IPTD_{i,pkt})_{\Delta t_i}$
Minimum IPTD ($IPTD^{min}$)	$IPTD^{min}_{i,\Delta t_i} = min(IPTD_{i,pkt})_{\Delta t_i}$
IPTD Alarm ($IPTD^{alarm}$)	$IPTD^{alarm}_{i,\Delta t_i} = \overline{IPTD}_{i,\Delta t_i} - \overline{IPTD}_{i,\Delta t_{i,w}}$
Loss parameters (%)	
IP Loss Ratio (IPLR)	$IPLR_{i,tot} = tot\_pkts\_lost_i / tot\_pkts\_sent_i$
Mean IPLR ($\overline{IPLR}$)	$\overline{IPLR}_{i,\Delta t_i} = (\sum pkts\_lost_i / \sum pkts\_sent_i)_{\Delta t_i}$

to be defined and estimated in order to drive AC decisions. The on-line monitoring implementation options are discussed below.

4.4.1 *QoS and Performance Metrics*

Several edge-to-edge QoS and performance parameters have been identified to be controlled at each egress monitoring module. Generically, these parameters are used for both SLS auditing and QoS control in the domain. Depending on each service, some of them are also taken as inputs for AC, as defined in Table IV. Considering [30, 32] definitions, the following parameters are specified for a given ingress-to-egress pair (I_r, E_r), class i and time interval Δt (see Table V).

4.4.2 *Measurement Methodology*

For each class, the parameters in Table V are estimated and controlled, resorting to passive and active measurements. Comparing the outcome of both approaches allows to assess the usability and tuning of the probing process as regards probing periodicity and pattern (as discussed in Section 6). In more detail, passive measurements consist of evaluating the QoS parameters[5] for each measurement interval Δt_i, using the traffic aggregate (comprising active accepted flows) of service class i. A similar method is followed for active measurements but it only considers

[5] Note that passive measurements involving delay and loss related parameters are simplified when using a simulation scenario where timestamping and sequencing data can be easily included in each packet.

the probing traffic embedded in each service class.

4.4.3 *Parameters Estimation*

Apart from the measurement methodology itself, there are several measurement mechanisms which can be used for parameter estimation [37, 45]. In particular, *Time-Window*, *Point Sample* and *Exponential Averaging* mechanisms are commonly an option due to their simplicity. In brief, *Time-window* (TW) computes an average for the parameter under control for every sampling period S. After a window consisting of T samples S, the highest average is taken as the estimation for the next T window. At any time, the estimate is immediately increased when a measured sample is higher that the current estimation or a new flow is admitted. In the latter case, the estimate is increased by the corresponding flow's parameter value according to the parameter's semantics. For instance, when estimating network load, the advertised flow rate is added. At this point, the window T can be restarted so that a full new window is used to capture the new flow impact. This avoids a too optimistic AC view of the network whenever the new flow traffic is not immediately sensed by the measurements. *Point Sample* (PS) simply takes a sample of the parameter in each sampling period S as the average [37]. *Exponential Averaging* (EA) takes a sample of the parameter in each sampling period S, however, the average v' is computed as a function of previous measurements v and the current one v_t, i.e. $v' = (1-\gamma)v + \gamma v_t$. The parameter γ determines the weight the new measurement has in the estimated average. Similarly to the time-window mechanism, when a new flow is admitted, the estimation is artificially increased.

Tuning these mechanisms configuration, i.e. finding appropriate values for S and T is obviously a key point as regards the realism of the estimation. While S controls the measurement sensitivity, T controls the mechanism adaptability. A lower value of S leads to a higher sampling frequency. This means that the mechanism is more reactive to bursts, leading to high measures, ending up in a more conservative AC. In opposition, the larger S, the smoother the estimation process reacts to flow traffic variability. In this case, the number of accepted flows tends to increase. The value of T rules the window size. An higher T leads to less frequent estimate updates and more stability due to the memory effect of past estimates. This reduces the capacity of the method to react, leading to a more conservative estimation approach. The window size T also needs to be balanced with the flow arrival rate and flow duration. While the former impacts on how the time window is reinitialized, the latter may lead to over-estimates in case of short-lived flows. The most visible effect on parameter estimation, and indirectly in AC, results from tuning T. In order to obtain a statistically meaningful number of samples, following [37] guidelines, $T/S \geq 10$ with $S > 100 * L/C$, for a L bits packet size transmitted at rate C.

Although the described measurement mechanisms are usually applied to a single node, we have applied the same concepts to edge-to-edge measurements. For SLS utilization control, the class traffic load is the estimated parameter. This value, which corresponds to an aggregate rate for the class, is obtained resorting to the three estimation mechanisms described above in order to assess which of them more closely reflects the real network behavior (see Section 6.2). For QoS control, the point sample method is used. However, as in [46], for each sampling period S, independently of which estimation method is in use, the parameter average is taken instead of an instantaneous value. Considering the passive measurement methodology, this corresponds to use all the packets in a class[6]. As regards the active measurement methodology, only the probing packets are used. Additionally, as our estimates are edge-to-edge, the dimensioning of S considers the one-way delay and not the packet transmission time.

5 SIMULATION SCENARIO

The main objectives of testing the proposed AC model in a multi-class domain are threefold [39]. First, we intend to assess the active measurement methodology as a whole. Both probing patterns, probing periodicity and probing ability to capture each class behavior are studied. Second, a comparison of the estimation mechanisms TW, Avg_PS and EA is carried out in order to evaluate which one provides the closest estimate to reality, tuning S and T timing parameters. Third, the proposed AC criteria are evaluated as regards their ability to ensure service commitments are not violated, while assessing both network utilization and QoS safety margins. As argued before, some degree of overprovisioning is considered for stringent QoS classes so that simplicity and flexibility of MBAC can be useful to control these classes too. In order to pursuit the objectives defined above, a simulation prototype based on the

[6] This is obviously too demanding, specially in high-speed networks due to large traffic volume and reduced packet processing time. In the present context, the only aim is to tune the probing process.

Network Simulator (NS-2) platform [47] was developed, being its main characteristics detailed below [39].

5.1 Simulation topology

The simulation topology is illustrated in Figure 6. We have adopted a simple initial configuration which allows to pursue the objectives defined above. Obviously to assess the model scalability and end-to-end behavior a more complex scenario will be considered. The network domain consists of ingress routers I_1, I_2, a core router C_1 and an edge router E_1. The service classes SC1, SC2 and SC3 are implemented in 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 traffic. This allows to evaluate concurrency in distributed AC and assess cross traffic impact.

The domain internodal links capacity is 34Mbps, with a 15ms propagation delay. $L_{C1,E1}$ link works as a bottleneck in this network topology. Access links have been configured not to influence the intra-domain measurement results.

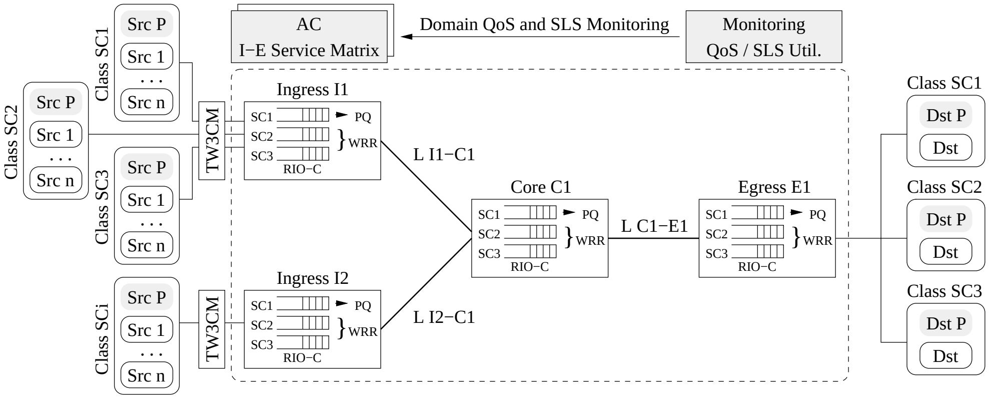

Figure 6. Simulation Topology

In each node, each class queue is 150 packets long. The scheduling discipline follows an hybrid Priority Queuing - Weighted Round Robin (PQ-WRR) and the active queue management (AQM) mechanism is RIO-C. 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 remaining class (SC2, SC3) queues are served with a 2 to 1 proportionality. At the network entrance, each traffic class is policed using a TSW3CM [43]. Table VI summarizes the parameters used in the topology configuration.

Table VI. Network Topology Configuration

Links	Delay	Capacity	Queues	AQM	Qsize	Scheduler	Policer
$L_{I1,C1}$	15ms	34Mbps	$Q_{I1,C1}$	rio-c	150 pkts	PQ-WRR(2,1)	n.a.
$L_{I2,C1}$	15ms	34Mbps	$Q_{I2,C1}$	rio-c	150 pkts	PQ-WRR(2,1)	n.a.
$L_{C1,E1}$	15ms	34Mbps	$Q_{C1,E1}$	rio-c	150 pkts	PQ-WRR(2,1)	n.a.
L_{access}	0ms	100Mbps	n.a.	n.a	n.a.	n.a.	TSW3CM

5.2 Source models

Generically, three source models have been considered: Constant Bit Rate (CBR) sources, Exponential on-off (EXP) and Pareto on-off (PAR) sources. EXP sources have exponential distributed sojourn on-off times, where during the on period an exponential random number of packets is generated at fixed rate $r$kbps. PAR sources follow the same behavior but now ruled by a Pareto distribution with a shape factor α. PAR sources with $1 < \alpha < 2$ under aggregation will allow to generate traffic exhibiting long-range dependence.

SC1 comprises UDP traffic with small to medium peak rate and packet sizes, as usually generated by some real-time streaming applications, such as voice over IP (VoIP). SC2 comprises UDP traffic with higher peak rate and packet sizes, as generated by other real-time streaming applications with high variability. SC3 supports generic TCP traffic. However, initial tests consider UDP traffic in this class.

The flow arrival process is Poisson with exponentially distributed interarrival and holding times. The choice of parameters for each source type is compiled in Table VII.

As regards probing, three in-band source types have been defined. In [35], two packets per second according to a Poisson distribution are used as a probing scheme to assess loss and delay between any two network measurement points. Here, the adequacy of this pattern is compared with both CBR and Exponential on-off probing. The use of EXP-like sources prevents possible synchronization among probing and other events in the IP network [35].

Table VII. Source Parameter Configuration

Class	Protocol	Src Type	Src Parameters	EXP Inter. Time	EXP Hold. Time
SC1	UDP	CBR_{SC1}	(r=100kbps, l=128bytes)	400ms-2s	60s
	UDP	EXP_{SC1}	(r=200kbps, l=128bytes, on=off=500ms)	400ms-2s	60s
	UDP	PAR_{SC1}	(r=200kbps, l=128bytes, on=off=500ms, α=1.5)	400ms-2s	60s
SC2	UDP	CBR_{SC2}	(r=0.5Mbps, l=512bytes)	400ms-2s	120s
	UDP	EXP_{SC2}	(r=1Mbps, l=512bytes, on=off=500ms)	400ms-2s	120s
	UDP	PAR_{SC2}	(r=1Mbps, l=512bytes, on=off=500ms,α=1.5)	400ms-2s	120s
SC3	TCP	FTP App.	(r=unspecified, l=512bytes)	400ms-2s	120s
SC3	UDP	CBR_{SC3}	(r=0.5Mbps, l=512bytes)	400ms-2s	120s
	UDP	EXP_{SC3}	(r=1Mbps, l=512bytes, on=off=500ms)	400ms-2s	120s
	UDP	PAR_{SC3}	(r=1Mbps, l=512bytes, on=off=500ms, α=1.5)	400ms-2s	120s
Probing	UDP	CBR_P	(r=1.6kbps (2pkts/s), l=100bytes)	1 src	sim. duration
	UDP	POI_P	(r=1.6kbps, l=100bytes, Poisson)	1 src	sim. duration
	UDP	EXP_P	(r=3.2kbps, l=100bytes, on=off=250ms)	1 src	sim. duration

Table VIII. Service Parameter Configuration

Service Class	SLS Rate R_s (% share)	Utilization Target β_s	QoS Parameter	Threshold
SC1	3.4Mbps (10%)	0.75	(IPTD,ipdv,IPLR)	(35ms,1ms,10^{-4})
SC2	17.0Mbps (50%)	0.90	(IPTD,IPLR)	(50ms,10^{-3})
SC3	13.6Mbps (40%)	1.00	(IPLR)	(10^{-1})

5.3 Service and AC Configuration

Table VIII illustrates the main parameters used to configure the AC rules, for controlling both SLS utilization and domain QoS levels. Three downstream SLSs have been considered, one per service class, with a negotiated rate (R_s) defined according to the traffic load share intended for the corresponding class in the domain. The MS algorithm that rules SLS utilization has specific utilization target (β_s) values depending on how conservative the AC decisions must be. For instance, a β_s=0.75 corresponds to impose a safety margin of 25% to absorb load fluctuations and optimistic measures. This value can be viewed as a degree of overprovisioning. The AC thresholds which rule the control of the 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 [48].

6 SIMULATION RESULTS

In this section, the results are discussed following three related vectors evaluating: (i) the probing process (ii) the estimation mechanism and (iii) the AC criteria. The results were obtained running multiple simulations of about eight minutes after discarding the results from an initial convergence period. The definitions and values included in Tables V, VI, VII, and VIII were generically used throughout the experiments.

6.1 Evaluation of the Probing Process

Generically, the probing process is assessed in terms of finding to which extent it can be used to monitor network status. In other words, we verify how realistically the probing traffic can capture or follow a particular class behavior. In this way, three distinct probing sources (CBR, POI, EXP) were considered to measure the parameters defined in Table V. For each class, the probing measuring outcome was cross-checked against the corresponding measures using the traffic in the class.

Despite the traffic type in each class, the results obtained are consistent and similar for the probing sources considered. Our major findings are: (i) probing can be successfully used to evaluate both IPTD and mean IPTD. This is true both for capturing the shape and scale of these QoS metrics. Figure 7 shows an almost perfect match for the mean IPTD. This is valid even for probing rates as low as two packets per second, (ii) for the test conditions, probing is inappropriate for measuring both ipdv and IPLR, as Figure 7 shows. As ipdv is a consecutive packet measure, probing gaps lead to higher measures as a consequence of queue occupancy variations. For IPLR, as loss is typically seen as a rare event and probing packets are marked as high priority (green) packets, the probing method is again inadequate. The exception to this behavior occurs when traffic in a class suffers heavy loss (over 10%). This mis-behavior can be reduced resorting to higher probing rates and using red-marked probe traffic. However, the overhead introduced may be prohibitive, in particular, if it is performed in-band. Due to its particular rate characteristics probing rate cannot be directly compared to class throughput and load estimation. For bandwidth measurements, specific probing techniques must be used [49, 38].

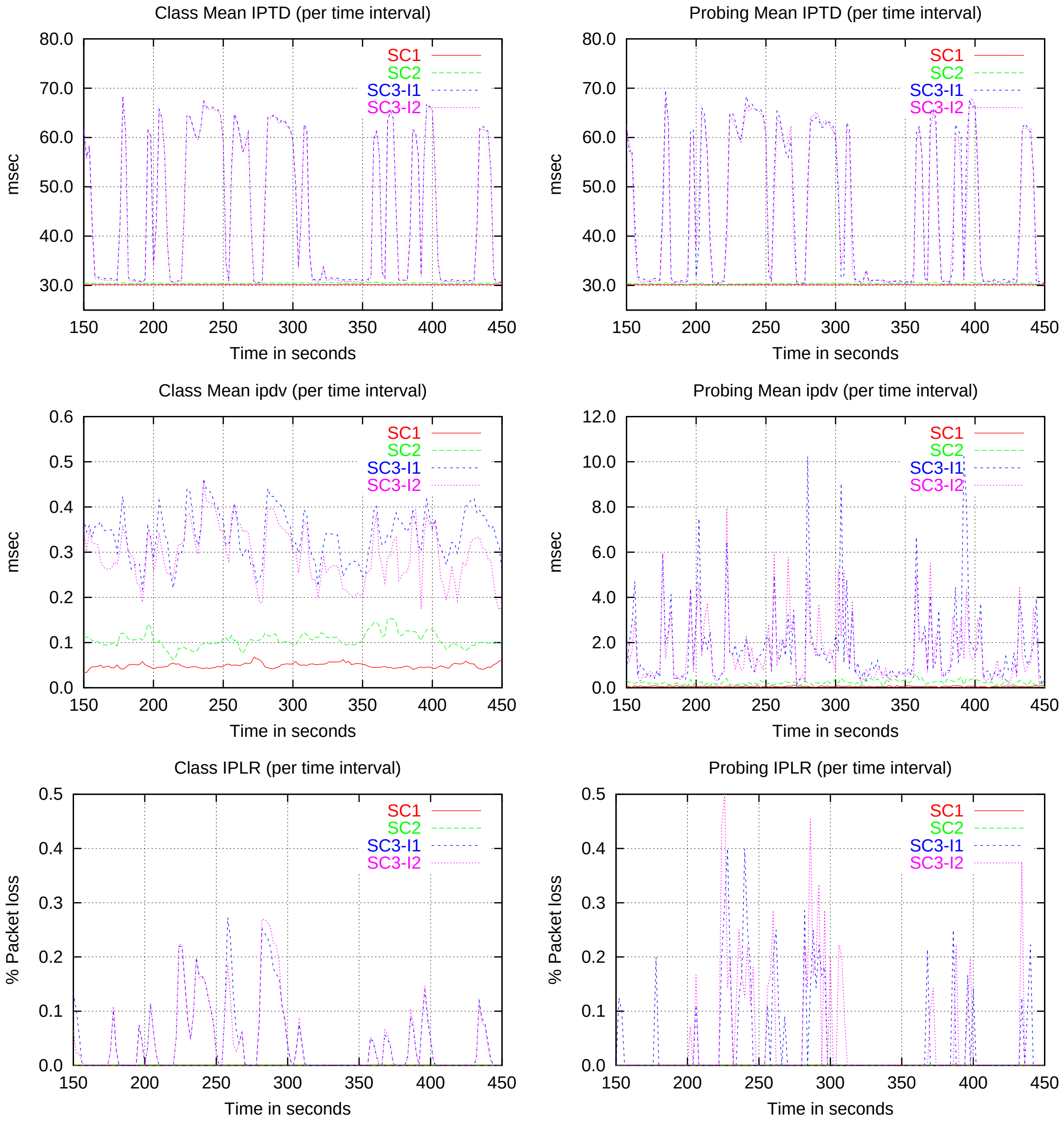

Figure 7. Class (left) and Probing (right) Mean IPTD, Mean ipdv and IPLR

6.2 Evaluation of the Estimation Mechanism

In a first instance, the evaluation method consists of determining on how close an estimate is to the real traffic load. In this way, the rate estimation of each service class using the estimation mechanisms TW, Avg_PS and EA (described in Section 4.4.3) is compared (see Figure 8), taking Avg_PS estimates as reference. This is because Avg_PS represents the real aggregate mean rate in a pre-defined interval S.

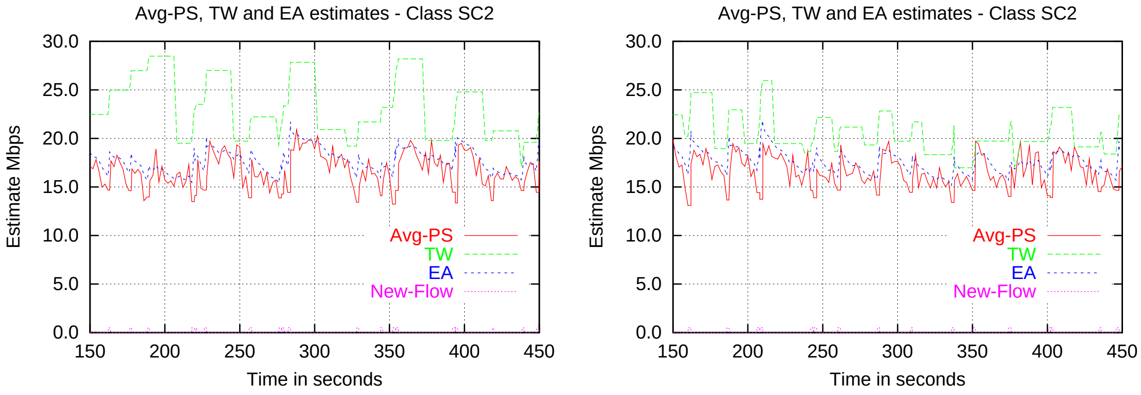

Figure 8. Estimation Mechanisms for (T=10; S=2; $\gamma = 0.25$; flow(EXP,500kbps,i.a.t.=0.4s,h.t.=120s) (a) Reseting T (b) without Reseting T on Flow Admittance

In most of the cases, TW leads clearly to overestimation of the metrics and, in practice, it can be a very conservative method. In special, when the window T is reinitialized upon a new flow admission and for short flow interarrivals, the estimate increases steadily as the departure of flows is not taken into account. The importance of the ratio between flow duration and T is studied in detail in [37]. However, this method allows to consider in advance the weight the new admission might have. This also occurs with EA, where the estimation is artificially increased when a new flow is admitted. This estimation method is controlled by the parameter γ. As shown in Fig. 8, using $\gamma = 0.25$ a closer match is achieved.

As far as AC is concerned, there are other aspects to consider: (i) the estimate is due to be used during a time interval T and (ii) the estimate needs to reflect, and somehow foresee, the network behavior trends. The tests on AC criteria, using all the above estimation mechanisms, show that Avg_PS allows to achieve high network utilization without service violations, for CBR traffic. However, for EXP and PAR traffic, all services have suffered disruption. EXP/PAR traffic fluctuations and a particularly low estimate leads to over acceptance. When this happens, in the following estimation period, the AC rate and QoS control rules will stop the new flows' entrance, however, degradation occurs during the lifetime of the existing ones. This effect can be reduced with TW and EA due to their initial accounting on the flow rate. In fact, an immediate increase on the estimate, reflecting the impact the new flow will have, allows a more adaptive and conservative AC.

6.3 Evaluation of the AC Criteria

While the active monitoring process is being tuned, passive measurements are used to evaluate the proposed AC criteria so that results are not misleading. The Avg_PS mechanism (with $S = 2s$) which represents the real average values for the parameters under control is used. In these experiments, concurrent SC3 traffic is injected through ingress I2. As expressed in Table VIII, SC1 traffic is blocked whether the sum of the rate estimate ρ_s and the flow's peak rate r_j is above 75% of the class share (see Equation (1)), or any of the QoS controlled parameters exceed the pre-defined thresholds. For SC2, a safety margin of 10%, which corresponds to an utilization target β_s=0.9, was defined and the flow mean rate is used instead. SC3 does not use safety margin but controls IPLR. The tests are performed under high demanding conditions with a flow interarrival of 400ms.

Table IX summarizes the results obtained for each class and each source type as regards: (i) the average of concurrent active flows; (ii) the percentage of packets exceeding the pre-defined delay bounds; (iii) the total loss ratio; (iv) the new utilization target proposal for which no QoS violations occur; and (v) the new average of concurrent active flows. The utilization considering the bottleneck capacity (34Mbps) is depicted in Fig. 9(a).

The results obtained show that while for CBR traffic there is no loss and reduced delay violations, for EXP and PAR sources an increasing packet loss is clearly noticed. 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 persists over the next intervals while they last. This over acceptance is caused by traffic fluctuations combined with the type of estimation mechanism used (without prevision). To minimize this, more conservative estimates or larger safety margins are needed. We have explored this last option for EXP traffic, where new safety margins leading

Table IX. AC Test Results

Class	Src Type	#act_flows	%pkts_viol:(IPTD;ipdv)	%IPLR	New Util.Target	new #act_flows
SC1	CBR_{SC1}	26.0	(0.41 ; 0.14)	0.0	0.55	19
	EXP_{SC1}	27.0	(0.33 ; 0.08)	2.2		
	PAR_{SC1}	26.5	(0.30 ; 0.08)	2.9		
SC2	CBR_{SC2}	31.5	(0.05 ; n.a)	0.0	0.75	27.5
	EXP_{SC2}	34.5	(0.07 ; n.a)	1.5		
	PAR_{SC2}	34.0	(0.08 ; n.a)	1.1		
SC3	CBR_{SC3}	16.0+16.0	(n.a. ; n.a)	0.7	0.80	13.5+14.5
	EXP_{SC3}	15.5+17.5	(n.a. ; n.a)	17.4		
	PAR_{SC3}	16.5+19.0	(n.a. ; n.a)	20.2		

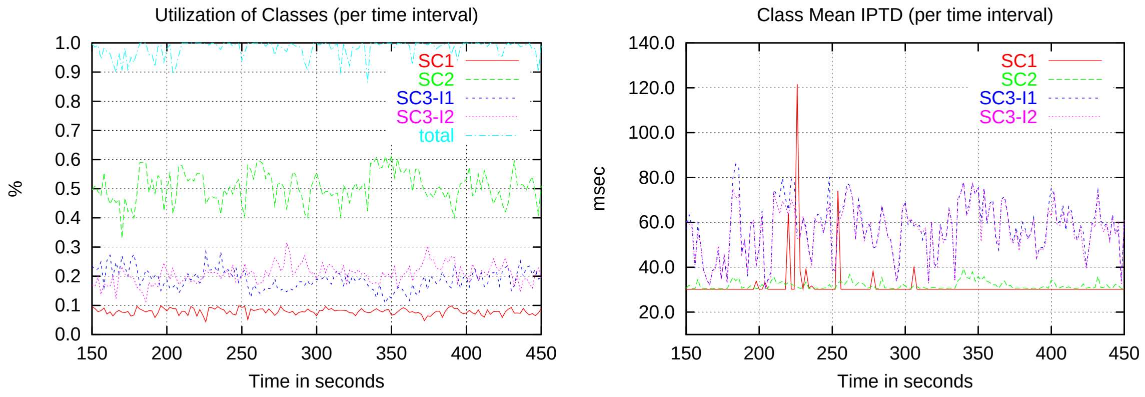

Figure 9. Initial Utilization Target: (a) Utilization (b) Mean IPTD (for EXP sources)

to no QoS violations were established. Fig. 10 shows that, under the new defined margins, the AC criterion achieves good network utilization. In this tuning process, we found that, in SC3, IPLR is a difficult parameter to control. Even when it is below the defined upper bound, the Mean IPLR per interval may exceed the corresponding threshold. This may be due to the low weight in the scheduling discipline which serves the corresponding queue. In addition, the results show that SC1 can be particularly affected by the scheduling mechanism. While PQ is suitable and commonly used to handle in-profile high priority aggregates (EF traffic), if the aggregate rate exceeds the maximum rate allowed by the scheduler, the class is severely punished. This is evident through an increase of IPTD and IPLR (see Fig. 9 (b)), which stems from a head-of-line blocking while waiting for the scheduling cycle to be completed. This stresses the need of having a tight control on this class and a wider safety margin.

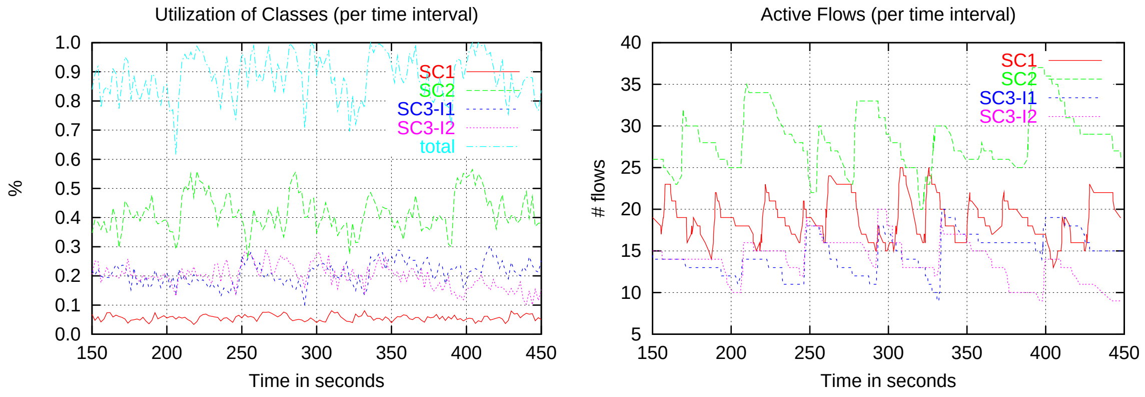

Figure 10. New Utilization Target: (a) Utilization (b) Number of Accepted Flows

6.4 Scalability issues

The scalability of the proposed AC model is certainly an important topic to discuss and evaluate as it may limit or compromise the solution overall performance and the deployment in real environments. In our view, the network dimension, the number of service classes SCi and the number of AC flow requests and acceptances are generically the main variables to consider when studying the scalability of an AC proposal. These aspects have proved to be serious constraints when the network keeps per flow state information and/or AC decisions involve all network nodes along the data path [50, 51]. When AC takes an SLS as reference, the number of existing SLSs should also be considered as a variable with impact on scalability.

The properties of the AC proposed model, where QoS control is carried out in an edge-to-edge basis and SLSs control is embedded in the corresponding service class, increase the model resilience to scalability problems. Although scalability issues are currently under test using a more complex simulation scenario, a summary of the impact they may have on the AC solution is highlighted in Table X. On going work intents to sustain and extend this initial considerations providing quantitative results on this topic.

7 CONCLUSIONS

This paper discusses and evaluates a lightweight, distributed AC model for managing network services quality and SLSs in class-based networks. The proposed AC model, based on on-line QoS and SLS monitoring, is simple,

Table X. Scalability Issues

Main variables	Possible impact on the proposed AC model
network dimension	independent of network core complexity (topology and control mechanisms) dependent on the number of edge nodes ($I_n \times E_m$, worst case) edge state information and monitoring overhead no significant impact expected on AC criteria efficiency may increase the need for handling concurrent AC
number of SCi	SCi state information at I_n nodes (Ingress-Egress Matrix) QoS monitoring overhead at E_m nodes monitoring intrusion (active methodology)
number of SLSs	SLS information at involved edge nodes utilization monitoring overhead no impact on QoS monitoring
number of flows	number of AC decisions no impact on domain state information TC at source domain I_n (optional)

easy to deploy and provides enough flexibility to accommodate distinct network services, both intra-domain and end-to-end. The AC strategy involves only the edge nodes of a domain and avoids complex AC signaling. The intra-domain operation controls both the QoS levels in the domain and the utilization of the contracted SLSs with downstream domains using an AC module at ingress nodes and a monitoring module at egress nodes. The end-to-end operation uses the flow request to perform both AC at domain entrance and end-to-end available service check, avoiding extra control mechanisms.

From a practical perspective, we evaluate the two main components of the model: the monitoring process, which controls QoS and SLS parameters, providing inputs to AC; and the AC criteria which guide the AC decisions. Parameter estimation methodologies and mechanisms are compared and tuned. Both probing patterns and periodicity were assessed as regards its ability to capture the behavior of the different classes. The results show that probing is a good solution to measure IPTD, however, for the patterns used, it cannot capture ipdv and IPLR behavior properly. The evaluation of the proposed AC criteria, as regards their ability to ensure service commitments, shows that using proper AC rules and safety margins, the simplicity and flexibility of this measurement-based AC approach can be successfully used to manage service quality. Current work includes further tuning of the active monitoring process and exploring alternative approaches to deal with concurrent AC decisions.

REFERENCES

1. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, An Architecture for Differentiated Services, IETF RFC 2475, 1998.
2. D. Grossman, New Terminology and Clarifications for Diffserv, IETF RFC3260, Apr. 2002.
3. S. Lima and P. Carvalho and A. Santos and V. Freitas, A Distributed Admission Control Model for CoS Networks using QoS and SLS Monitoring, in *IEEE International Conference on Communications - ICC'03*, May 2003.
4. 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.
5. 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.
6. Z.-L. Zhang, Z. Duan, Y. T. Hou, and L. Gao, Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services, in *ACM SIGCOMM'00*, 2000.
7. R. Neilson, J. Wheeler, F. Reichmeyer, and S. Hares, A Discussion of Bandwidth Broker Requirements for Internet2 Qbone Deployment, Internet2 Qbone BB Advisory Council, Aug. 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. L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang, Endpoint Admission Control: Architectural Issues and Performance, in *ACM SIGCOMM'00*, 2000.
11. 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.
12. V. Elek, G. Karlsson, and R. Rnngren, Admission Control Based on End-to-End Measurements, in *IEEE INFOCOM'00*, 2000.
13. R. Mortier and I. Pratt and C. Clark and S. Crosby, Implicit Admission Control, *IEEE Journal on Selected Areas in Communication*, Vol. 18, No. 12, pp. 2629–2639, Dec. 2000.
14. N. Benameur, S. Fredj, F. Delcoigne, S. Oueslati-Boulahia, and J. Roberts, Integrated Admission Control for Streaming and Elastic Traffic, in *QofIS'01*, M. Smirnov, J. Crowcroft, J. Roberts, and F. Boavida, Eds., Vol. 2156, Sept. 2001, pp. 67–81.
15. S. Lima, A Comparative Analysis of Admission Control Strategies, Technical Report TR0102l, Mar. 2002.
16. S. Lima, P. Carvalho, A. Santos, and V. Freitas, Long Range Dependence of Internet Traffic Aggregates, in *IFIP Networking 2002*, E. Gregori, M. Conti, A. Campbell, G. Omidyar, and M. Zuckerman, Eds., Vol. 2345. Springer, May 2002, pp. 1159–1164.
17. 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, Jan. 2004.
18. A. Diaconescu, S. Antonio, M. Esposito, S. Romano, and M. Potts, Cadenus D2.3 - Resource Management in SLA Networks, Cadenus Project IST-1999-11017, May 2003.
19. D. Goderis, S. Bosch, Y. T'Joens, O. Poupel, C. Jacquenet, G. Memenios, G. Pavlou, R. Egan, D. Griffin, P. Georgatsos, and P. Heuven, Service Level Specification Semantics and Parameters, IETF draft: draft-tequila-sls-02.txt (work in progress), Feb. 2002.
20. A. Sevasti and M. Campanella, Service Level Agreements Specification for IP Premium Service, Geant and Sequin Projects, Oct. 2001.
21. S. Salsano, F. Ricciato, M. Winter, G. Eichler, A. Thomas, F. Fuenfstueck, T. Ziegler, and C. Brandauer, Definition and Usage of SLSs in the Aquila consortium, IETF draft: draft-salsano-aquila-sls-00.txt, Nov. 2000.
22. M. Mellia, C. Casetti, G. Mardente, and M. Marsan, An Analytical Framework for SLA Admission Control in a Diffserv Domain, in *IEEE INFOCOM'03*, Mar. 2003.

23. J. Chen, A. McAuley, V. Sarangan, S. Baba, and Y. Ohba, Dynamic Service Negotiation Protocol (DSNP) and Wireless Diffserv, in *ICC'02*, Apr. 2002.

24. P. Bhoj, S. Singhal, and S. Chutani, SLA Management in Federated Environments, *Computer Networks*, Vol. 35, No. 1, Jan. 2001.

25. P. Trimintzios, I. Andrikopoulos, G. Pavlou, C. Cavalcanti, D. Goderis, Y. T'Joens, P. Georgatsos, L. Georgiadis, D. Griffin, C. Jacquenet, R. Egan, and G. Memenios, An Architectural Framework for Providing QoS in IP Differentiated Services Networks, in *7th IFIP/IEEE International Symposium on Integrated Network Management - IM'01*, 2001.

26. A. Prieto and M. Brunner, SLS to Diffserv Configuration Mappings, in *12th International Workshop on Distributed Systems: Operations and Management - DSOM'01*, Oct. 2001.

27. D. Lorenz and A. Orla, Optimal partition of QoS requirements on unicast paths and multicast trees, *IEEE/ACM Transactions on Networking*, Vol. 10, pp. 102–114, Feb. 2002.

28. D. Raz and Y. Shavitt, Optimal Partition of QoS requirements with Discrete Cost Functions, *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 18, No. 12, pp. 2593–2602, Dec. 2000.

29. S. Leinen and V. Reijs, Geant D9.7 - Testing of Traffic Measurement Tools, Geant Project, Sept. 2002.

30. T. Chahed, TF-NGN - IP QoS Parameters, TF-NGN, Nov. 2000.

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

32. V. Paxson, G. Almes, J. Mahadavi, and M. Mathis, Framework for IP Performance Metrics, IETF RFC2330, 1998.

33. CAIDA Tools, 2002. [Online]. Available: http://www.caida.org/tools/index.xml

34. A. Liakopoulos, D2.1 - Monitoring and Verifying Premium IP SLAs, Sequin Project, Apr. 2002.

35. F. Georgatos and F. Gruber and D. Karrenberg and M. Santcroos and H. Uijterwaal and R. Wilhelm, Providing Active Measurements as a Regular Service for ISPs, in *PAM'01*, Apr. 2001.

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

37. S. Jamin, S. Shenker, and P. B. Danzig, Comparison of Measurement-Based Call Admission Control Algorithms for Controlled-Load Service, in *INFOCOM'97*, 1997, pp. 973–980.

38. C. Dovrolis and M. Jain, End-to-End Available Bandwidth: Measurement methodology, Dynamics, and Relation with TCP Throughput, in *ACM SIGCOMM'02*, Aug. 2002.

39. S. Lima, P. Carvalho, A. Santos, and V. Freitas, Managing Services Quality through Admission Control and Active Monitoring, in *6th IFIP/IEEE Management of Multimedia Networks and Services - MMMS'03*, A. Marshall and N. Agoulmine, Eds., Vol. 2839. Springer, 2003, pp. 142–154.

40. A. Bak, W.Burakowski, F. Ricciato, S. Salsano, and H. Tarasiuk, Traffic Handling in AQUILA QoS IP Networks, in *QofIS'01*, M. Smirnov, J. Crowcroft, J. Roberts, and F. Boavida, Eds., Vol. 2156, Sept. 2001, pp. 243–260.

41. B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, An Expedited Forwarding PHB (Per-Hop Behavior), IETF RFC 3246, Mar. 2002.

42. J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, Assured Forwarding PHB Group, IETF RFC 2597, June 1999.

43. W. Fang, N. Seddigh, and B. Nandy, A Time-Sliding Window Three Color Marker (TSWTCM), IETF RFC 2859, June 2000.

44. S. Lima and P. Carvalho and A. Santos and V. Freitas, A Distributed Admission Control Model for Class-based Networks, in *8th IEEE International Conference on Communications Systems - ICCS'02*, Nov. 2002.

45. M. Gerla, A Survey of Admission Control Algorithms, UCLA, Tech. Rep. CS215, Dec. 1998. [Online]. Available: http://www.cs.ucla.edu/t̃ang/papers/admission_control_paper.pdf

46. S. Floyd, Comments on measurement-based admissions control for controlled-load services, LBNL, Tech. Rep., 1996.

47. Network Simulator. [Online]. Available: http://www.isi.edu/nsnam/

48. V.Reijs, Perceived quantitative quality of applications, jul 2001. [Online]. Available: http://www.heanet.ie/Heanet/projects/nat_infrastruct/perceived.html

49. C. Dovrolis, P. Ramanathan, and D. Moore, What Do Packet Dispersion Techniques Measure? in *IEEE INFOCOM'01*, 2001.

50. R. Braden, D. Clark, and S. Shenker, Integrated Services in the Internet Architecture: an Overview, IETF RFC1633, 1994.

51. J. Wroclawski, The Use of RSVP with IETF Integrated Services, IETF RFC 2210, Sept. 1997.

Solange Lima graduated in 1991 and obtained her MSc degree in Computer Communications at the University of Minho, Braga, Portugal, in 1997. She is currently an active researcher of the Computer Communications Group, where she is finishing her PhD degree. She is also a Senior Lecturer of Computer Communications, Departament of Informatics, at the University of Minho, Portugal.

Paulo Carvalho graduated in 1991 and obtained his PhD degree in Computer Science at the University of Kent at Canterbury, Canterbuty, UK, in 1997. He is currently Assistant Professor of Computer Communications, Departament of Informatics, at the University of Minho, Portugal.

Vasco Freitas graduated in 1972 and obtained his MSc and PhD degrees in Control and Computer Communications at the University of Manchester, UK, in 1977 and 1980. From 1989 until 1994 he was Director of Networking at the Portuguese Foundation for Scientific Computing to establish the National University Data Network. He is currently Professor of Computer Communications at the University of Minho, Portugal.