Self-adaptive Distributed Management of QoS and SLSs in Multiservice Networks

S. Rito Lima, P. Carvalho, V. Freitas
Computer Communications Group, University of Minho
Department of Informatics, 4710-057 Braga, Portugal
{*solange*|*pmc*|*vf*}*@di.uminho.pt*

Abstract

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 way, we describe and specify a lightweight distributed admission control (AC) model which provides an uniform solution for managing QoS and SLSs in multiclass and multidomain environments. Taking advantage of the consensual need of on-line service monitoring and traffic control at the network edges, AC decisions are driven by feedback from systematic edge-to-edge measurements of relevant QoS parameters for each service type and SLS utilization. This allows self-adaptive service and resource management, while abstracting from network core complexity and heterogeneity. In this paper, introducing an expressive notation, we specify the high-level entities for multiservice provisioning in a domain and formalize service-dependent AC equations to assure both intra and interdomain model operation. A proof-of-concept of the AC criteria effectiveness in satisfying each service class commitments while achieving high network utilization is provided through simulation.

Keywords

Network and service management, admission control, QoS and SLS monitoring

1. Introduction

The support of multiconstrained applications and services in the Internet is a multilevel problem involving enhanced protocols, devices and media, contributing ideally for a seamless end-to-end quality-of-service (QoS) solution. From the network perspective, providing QoS brings an additional burden to the IP level. As new policies, rules and traffic control mechanisms have to be deployed, a major principle to preserve should be *keep it simple*. The design of service-oriented networks based on the class-of-service (CoS) paradigm [3], where traffic is aggregated in a limited number of classes according to its QoS requirements, pursuits this principle. 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. In this way, taking simplicity, flexibility and easy deployment as initial goals, in [15, 13] we propose a distributed AC model based on edge-to-edge on-line QoS and SLS monitoring for managing the quality of multiple services in CoS IP networks.

In this paper, we extend and fully specify both the components and the operation of the proposed model for a multiclass and multidomain environment. Although AC has been

extensively studied in the literature (see Sec. 2), few studies deal with the simultaneous management of domain QoS levels and interdomain SLSs, falling short in covering and formalizing concrete AC equations to be applied to multiservice networks. The present study covers extensively these aspects, contributing with new insights on how to manage intra and interdomain operation aiming at an uniform and ubiquitous end-to-end QoS solution, irrespective of each domain inherent heterogeneity. A proof-of-concept of the proposed management scheme is also provided, illustrating its self-adaptive ability in controlling QoS and SLS parameters in a multiservice domain.

The remaining of this document is organized as follows: a debate of current AC approaches and an overview of the proposed AC model are carried out in Sec. 2; the main network domain entities concerning multiservice AC, SLS and QoS management are formalized in Sec. 3, where an intuitive and expressive notation is introduced; this notation supports the intra and interdomain AC criteria specification provided in Sec. 4; the model evaluation results are discussed in Sec. 5; finally, the conclusions are drawn in Sec. 6.

2. Multiservice AC

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. Some proposals suggest the use of central entities for AC and resource management [7, 11, 21], however, the well-known problems of centralization led to several decentralized AC approaches. In this context, measurement-based AC solutions involving only edges nodes (EMBAC), using either active or passive measurement strategies of network load and/or QoS parameters [4, 5, 8], have deserved special attention. These strategies are suitable for the provisioning of soft service guarantees, leading to reduced control information and overhead, but eventually to QoS degradation. Despite not requiring changes in the network, active EMBAC increases the initial latency and network load as probing is carried out on a per application basis. To provide hard service guarantees current AC proposals need to control the state and the load of traffic aggregates in the core nodes [7, 20], or even perform AC in these nodes. These solutions tend to require significant network state information and, in many cases, changes in all network nodes. The need to control elastic traffic, for more efficient network utilization, has also been discussed and implicit AC strategies have been defined [17, 2].

In our view, to achieve a simple and manageable multiservice AC solution a certain degree of overprovisioning is recommended in order to simplify AC while improving QoS guarantees. Attending to this aspect and to the related work mentioned earlier, the AC strategy described in the following sections is a step forward in performing distributed and lightweight AC in multiservice environments. Our approach avoids the use of per application intrusive traffic and the initial latency of edge-to-edge measurement-based solutions, while benefiting from their inherent simplicity for managing SLS and QoS

both intra and interdomain. A brief overview of the model operation principles [15, 13] is provided next so that the model formalization is better understood and sustained.

2.1 AC 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. 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.

3. Multiservice Domain Specification

Taking into account the overview of the AC model operation described above, in this section, we specify the main components of a generic network domain comprising multiple ingress and egress routers, as regards the provision of multiservices to customers (individuals or other domains). In a domain, we consider and specify the following entities: (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. 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.

3.1 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*. For this domain D_x, we define the set of service classes supported within D_x as $SC^{D_x} = \{SC_1, SC_2, ..., SC_Y\}$. From a service management

* 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 node E_m. Therefore, I^{D_x} and E^{D_x} include all ingress and egress interfaces to other domains.

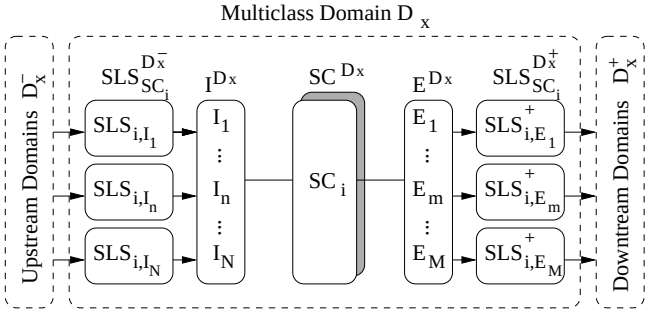

Figure 1: Domain Main Elements and Notation

and provisioning point-of-view, the definition of a service class includes identifying a set of QoS parameters under control and corresponding safety margins. Each parameter target value affected by this safety margin will allow to establish a threshold for the parameter inside the domain. These QoS thresholds are used for triggering traffic control mechanisms such as AC, reducing the change of QoS violation in the service class. Thus, 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. Then, the parameter upper bound or threshold is given by $T_{i,p} = \beta_{i,p} P_{i,p}$.

In practice, the service classes to be supported in domain D_x are closely related to the service levels negotiated with both upstream and downstream customers. 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 $SLS_{SC_i}^{D_x^+} = \{SLS_{i,E_m}^+ | E_m \in E^{D_x}\}$. Therefore, D_x is a service provider for D_x^- and a customer of D_x^+. Note that, SLS_{i,I_n} identifies a specific SLS accepted for SC_i with upstream domain D_x^-, connecting D_x through I_n, and SLS_{i,E_m}^+ identifies a specific SLS negotiated for SC_i with downstream domain D_x^+, accessible from D_x through E_m (see Fig. 1).

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. The global rate share of these users is controlled by $R_{i,I_n}^{\notin SLS}$. Therefore, $R_{i,I_n}^{\notin SLS}$ is a rate-based parameter defined to limit traffic not sustained by a specific SLS, i.e. for flows $F_j \notin SLS_{i,I_n}$. This allows 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 $F_j \in SLS_{i,I_n}$.

3.2 Upstream SLSs Specification

The definition of SLSs, apart from being a key aspect for QoS provisioning, provides a valuable input for AC, in special, when admission spans multiple domains. Therefore, 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 [16, 9, 19, 18] and management [16, 10, 6, 22].

Taking these inputs into account, an SLS template including relevant parameters and their typical contents was defined in [15]. Following that template, from an AC perspective, an upstream SLS for service class SC_i, SLS_{i,I_n}, should consider elements such as:

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 $E^{'} \subseteq E^{D_x}$ represents all possible egress nodes E_m providing access from D_x to D_x^+ for this SLS. At this point, the scope of SLS_{i,I_n} is limited to a single domain D_x, which is responsible for identifying $E^{'}$ 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. Whenever a domain uses proprietary service identifiers, appropriate mapping is needed at domain boundaries.

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}. Considering R_{i,I_n} as the aggregate rate established for SLS_{i,I_n} in the scope region, R_{i,I_n} can be expressed either as a global value or as a vector of rates $\vec{R}_{i,I_n} =< R_{i,(I_n,E_1)}, ..., R_{i,(I_n,E_m)} >$ depending on the scope of the SLS, i.e. taking all $E_m \in E'$. Handling a vector of rates may be useful for transit domains which want to pre-allocate resources for more demanding or high priority services. When the traffic profile is a vector of rates $\vec{R}_{i,I_n}$, an upstream SLS for SC_i can be defined as a matrix,

$$SLS_{i,I_n} = [SLS_{i,(I_n,E_1)}, ..., SLS_{i,(I_n,E_M)}] \tag{1}$$

where each $SLS_{i,(I_n,E_m)}$ element is null if $E_m \notin E'$, i.e. when E_m is outside the defined SLS scope. Considering this SLS_{i,I_n} definition and the set of upstream SLSs, i.e. $SLS_{SC_i}^{D_x^-}$, an Ingress-to-Egress Matrix of accepted and active SLS for a generic service class SC_i can be defined as

$$\Phi_{SLS}^{SC_i} = (\phi_{n,m}^i) \qquad \phi_{n,m}^i = SLS_{i,(I_n,E_m)} \tag{2}$$

An $SLS_{i,(I_n,E_m)}$ is effectively active whenever its negotiated scheduling period is valid, i.e. $t_{actual} \in [t_{i,I_n,0}, t_{i,I_n,f}]$. Therefore, when an element of $\Phi_{SLS}^{SC_i}$ is zero, the corresponding SLS does not exist, is not yet active or has expired. $\Phi_{SLS}^{SC_i}$ allows to infer the expected traffic matrix for service class SC_i, which may then be used for current service provisioning in domain D_x. When R_{i,I_n} is defined as a single and global rate value independently of each egress $E_m \in E'$, $\Phi_{SLS}^{SC_i} = (\phi_n^i)$.

4. $SLS_{i,I_n} \rightarrow ExpectedQoS$ specifies the expected QoS parameters for SLS_{i,I_n}, 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, each QoS parameter $P_{i,I_n,p}$ value is bounded by the corresponding service class $P_{i,p}$, regardless the incoming I_n and accepted SLS_{i,I_n}. In other words, it is the QoS parameter target value for the class that bounds the corresponding SLS's expected QoS value. Depending on each parameter semantics, $P_{i,p}$ can either be an upper or lower bound. 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 $IPTD_{i,I_n}, ipdv_{i,I_n}, IPLR_{i,I_n}$.

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 [19], this interval is recommended to be month-range.

3.3 Downstream SLSs Specification

In a domain D_x, when defining and negotiating an SLS with a downstream domain D_x^+, i.e. an SLS_{i,E_m}^+, the contracted service from a particular egress node E_m should foresee the provision of adequate service levels taking into account all active SLSs going through E_m. From an E_m perspective, specifying 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. In the same way, collecting $SLS_{SC_i}^{D_x^+}$ information, an egress-based matrix can be defined,

$$\Phi_{SLS^+}^{SC_i} = (\phi_m^{i+}) \qquad 1 \leq m \leq M \tag{3}$$

representing accepted and active downstream SLSs. If an egress node E_m does not have a defined SLS^+ for class SC_i, ϕ_m^{i+} is null. The negotiated traffic profile for $\forall SLS_{i,E_m}^+ \in SLS_{SC_i}^{D_x^+}$ is given by $SLS_{i,E_m}^+ \rightarrow TrafficProfile = \bigoplus_{k=1}^{N} \phi_{k,m}^i$, with the operator $\bigoplus$ denoting the aggregation of all accepted SLS_{i,I_n} for SC_i that may use E_m to leave D_x.

3.4 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 $TrafficProfile$, the required QoS parameters $ReqQoS$ and an optional $QoSTolerance$ factor. 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, as for the SLS_{i,I_n} definition,

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

2. $F_j \rightarrow ReqQoS$, identifying the flow's QoS requirements (if any), can be defined associating a tolerance to each flow's parameter, i.e. defining a set of parameters $P_{F_j} = \{(P_{j,1}, \gamma_{j,1}), ..., (P_{j,P''}, \gamma_{j,P''})\}$, 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 $P_{i,p}$. The tolerance to $P_{j,p}$ degradation, expressed by $\gamma_{j,p}$, may be considered by the AC criteria;

3. $F_j \rightarrow AccQoS$ is used to accumulate QoS metric values in a multidomain end-to-end AC operation (see Sec. 4.3);

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.

Table 1 Controlled QoS Parameters

Throughput r (bps)	$r_{i,\Delta t_i} = (\sum bits\_recv_i)_{\Delta t_i}/\Delta t_i$
Utilization U (%)	$U_{i,\Delta t_i} = r_{i,\Delta t_i}/C$
IP Transfer Delay ($IPTD$)	$IPTD_{i,pkt} = (t_{E_m,pkt} - t_{I_n,pkt})$
Mean IPTD	$\overline{IPTD}_{i,\Delta t_i} = (\sum IPTD_{i,pkt}/\sum pkts\_recv_i)_{\Delta t_i}$
Inst. Packet Delay Var. ($ipdv$)	$ipdv_{i,2pkt} = (IPTD_{i,pkt} - IPTD_{i,pkt-1})$
Mean ipdv	$\overline{ipdv}_{i,\Delta t_i} = (\sum \lvert ipdv_{i,2pkt}\rvert/\sum pkts\_recv_i)_{\Delta t_i}$
IP Loss Ratio (IPLR)	$IPLR_{i,tot} = tot\_pkts\_lost_i/tot\_pkts\_sent_i$
Mean IPLR	$\overline{IPLR}_{i,\Delta t_i} = (\sum pkts\_lost_i/\sum pkts\_sent_i)_{\Delta t_i}$

3.5 Monitoring and Controlling per-Class QoS Metrics

For service class SC_i and ingress node I_n, a dynamic Ingress-Egress Service matrix used to control QoS levels and support AC decisions is defined as

$$\Psi_{QoS}^{SC_i} = (\psi_m^i), \qquad 1 \leq m \leq M \tag{4}$$

Service data in the matrix $\Psi_{QoS}^{SC_i}$ 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. $\psi_m^i \rightarrow P_p = \tilde{P}_{i,(I_n,E_m),p}$. Using this measured data and corresponding class thresholds, a QoS status indicator, defined as $\psi_m^i \rightarrow AC\_Status$, 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 Sec. 4.2).

Examples of relevant edge-to-edge QoS parameters to be measured and the respective equations are defined in Table 1.

4. AC Criteria Specification

Following the generic AC model description provided in Sec. 2.1, the AC criterion resorts to (i) rate-based SLS control rules and (ii) QoS parameters control rules. These rules follow the notation introduced in Sec. 3.

4.1 Rate-based SLS Control Rules

For each ingress node $I_n \in I^{D_x}$ and each egress node $E_m \in E^{D_x}$ one or more SLSs can be in place. At this point, a single mapping between a service class SC_i within D_x and an SLS for the corresponding service type with an upstream D_x^- or downstream D_x^+ domain is assumed. 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.

$$\tilde{R}_{i,(I_n,*)} + r_j \leq \beta_{i,I_n} R_{i,I_n} \tag{5}$$

In Eq. (5), $\tilde{R}_{i,(I_n,*)}$ 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 (mean or peak) rate R_{i,I_n} for SLS_{i,I_n}. When R_{i,I_n} is viewed as a vector depending on a subset of egress nodes ($E' \subseteq E^{D_x}$) in the domain, $\tilde{R}_{i,(I_n,E_m)} + r_j \leq \beta_{i,(I_n,E_m)} R_{i,(I_n,E_m)}$.

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.

$$\tilde{R}^+_{i,(*,E_m)} + r_j \leq \beta_{i,E_m} R^+_{i,E_m} \tag{6}$$

In Eq. (6), $\tilde{R}^+_{i,(*,E_m)}$ 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.

$$\tilde{R}^+_{i,(*,Em)} = \sum_{k=1}^{N} \tilde{r}_{i,(I_k,E_m)} \tag{7}$$

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. 5) 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. (6), 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. (5) and (6) 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 Eq. (5) using the measured rate $R^{\notin SLS}_{i,I_n}$, i.e.

$$\tilde{R}^{\notin SLS}_{i,(I_n,*)} + r_j \leq \beta_{i,I_n} R^{\notin SLS}_{i,I_n} \tag{8}$$

Implicit AC - The equations above, which take into account both the rate estimates and the flow traffic profile, can be easily applied to 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. (9)), 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.

4.2 QoS Parameters Control Rules

When controlling the QoS levels in a domain, the QoS parameters and corresponding thresholds may vary depending on each service class SC_i commitments, the statistical properties of the traffic and degree of overprovisioning.

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.

$$\forall (P_{i,p}, \beta_{i,p}) \in P_{SC_i} : \tilde{P}_{i,p} \leq T_{i,p} \tag{9}$$

where $T_{i,p}$, as explained in Sec. 3.1, 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. (9) 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 Ψ^i_{QoS} available at each I_n.

4.3 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:

$$\forall P_{j,p} \in P_{F_j} : (\mathtt{op_1}(P_{j,p}^{acc^-}, P_{i,p}))\ \mathtt{op_2}\ (\gamma_{j,p} P_{j,p}) \tag{10}$$

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, $\mathtt{op_1}$ and $\mathtt{op_2}$ may express different operations, i.e. $\mathtt{op_1} \in \{add|sub|max|min|mul|f^{spec}\}$ and $\mathtt{op_2} \in \{\leq \mid < \mid \geq \mid > \mid =\}$. If the flow can be accepted in D_x, the new available service computation to be included in the flow request is given by

$$P_{j,p}^{acc} = \mathtt{op_1}(P_{j,p}^{acc^-}, P_{i,p}) \tag{11}$$

In order to clarify the use of Eq. (10) and Eq. (11), lets consider the following examples: (i) when $P_{j,p}$ is a delay parameter, a positive AC decision occurs when $add(P_{j,p}^{acc^-}, P_{i,p}) \leq \gamma_{j,p} P_{j,p}$; (ii) when $P_{j,p}$ is a loss ratio parameter, $op_1 = f^{spec}$, i.e. the cumulative computation of the loss in domain D_x is given by $P_{j,p}^{acc} = 1 - ((1 - P_{j,p}^{acc^-})(1 - P_{i,p}))$. For very low values of $P_{j,p}^{acc^-}$ and $P_{i,p}$, $op_1 = f^{spec}$ can be changed to an additive function ($op_1 = add$); (iii) when $P_{j,p}$ is a rate parameter, $op_1 = min$ may be used to carry the minimum available rate for SC_i across all the involved domains up to the destination.

5. Self-adaptive QoS and SLS Management

Before providing a proof-of-concept of the proposed AC model regarding its ability to self-adapt and manage QoS and SLS parameters in multiservice IP networks, we first describe the simulation test platform and the configuration of its main entities.

5.1 Implementing and configuring the test platform

A diffserv domain with AC based on the model described above was developed and implemented in the Network Simulator (NS-2) according to the topology illustrated in Fig. 2. In

* During Δt_i, only the rate measures $R_{i,I_n}^{\notin SLS}$, R_{i,I_n} and R_{i,E_m}^{+} can change at each I_n according to r_j of new admitted flows (if using explicit AC). Updating rate estimations leads to a more conservative AC as the rates of new flows are considered but the compensation effect of terminating flows is not taken into account. Keeping rate estimation unchanged during Δt_i explores this compensation effect but may increase overacceptance levels.

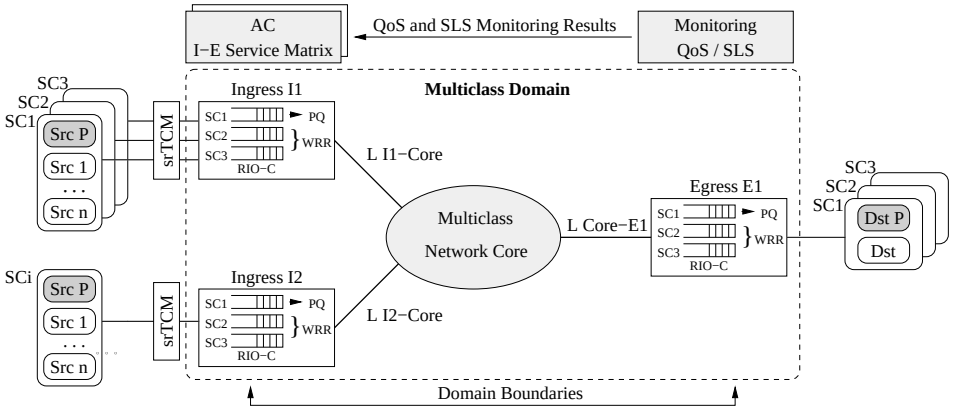

Figure 2: Network simulation topology

this simulation prototype, following current IETF service class guidelines [1], three fundamental service classes were defined. SC1, oriented to conversational services, provides a high quality service guarantee and is supported by EF PHB. This class may comprise traffic with hard real-time constraints such as VoIP, circuit emulation over IP or conversational UMTS traffic. SC2, oriented to a range of streaming applications with soft real-time constraints, provides a predictive service with low delay, low loss and minimum bandwidth guarantee and is supported by AF PHB. This class may comprise broadcast TV, audio and video streaming, webcasting or UMTS streaming traffic. SC3, oriented to elastic applications, generically, supports TCP adaptive traffic. Depending on the nature of TCP flows (e.g. high throughput vs. undifferentiated traffic), this class can be implemented using a AF or DF PHB. There is also the possibility of injecting concurrent traffic (CT-I2) of any of the above classes, allowing to test the effect of cross-traffic on probing.

SC1 and SC2 traffic is generated resorting to exponential on-off sources: SC1 comprises low rate UDP traffic sources (64kbps) with on/off periods of 0.96ms/1.69ms and small packet sizes (120bytes), reflecting voice-like traffic; SC2 also comprises UDP traffic with higher peak rates (256kbps), on/off periods of 500ms/500ms and larger packet sizes (512bytes). SC3 comprises long-lived high throughput TCP traffic, resulting from an FTP application generating packets of 512bytes. The flow arrival process is Poisson with exponentially distributed interarrival (0.4-2s) and holding times (90, 120, 180s for SC1, SC2 and SC3, respectively). The type of concurrent traffic injected at Ingress I2 is mapped to SC2. For each (I_n, E_m) pair, a single probing source is embedded in each service class. Depending on each class characteristics and QoS measuring requirements, probing sources should be selected and parameterized following the findings in [14].

The domain routers implement the three service classes according to a hybrid Priority Queuing - Weighted Round Robin (PQ-WRR(2,1)) scheduling discipline, with RIO-C for AQM. Each class queue is 150 packets long. The domain internodal link capacity is 34Mbps, with a 15ms propagation delay. At network entrance, SC1 is policed and marked using a token bucket which controls both rate and burst size, whereas SC2 and SC3 are

Table 2 SLS and QoS control

SLS Control Rule (Eq.(6))				
Class	Monitoring Inputs $\tilde{R}^{+}_{i,(*,E_m)}$	Flow Inputs r_j	SLS Rate R^{+}_{i,E_m} (share %)	Safety Margin β_{i,E_m}
SC1	Traffic load	peak rate	3.4Mbps (10%)	0.85
SC2	Traffic load	mean rate	17.0Mbps (50%)	0.90
SC3	Traffic load	n.a.	13.6Mbps (40%)	1.0
QoS Control Rule (Eq.(9))				
Class	Monitoring Inputs $\tilde{p}_{i,p}$	Flow Inputs	QoS Param. Thresh. $t_{i,p}$	
SC1	IPTD, ipdv, IPLR	if available	$35ms; 1ms; 10^{-4}$	
SC2	IPTD, IPLR	if available	$50ms; n.a.; 10^{-3}$	
SC3	IPLR	n.a.	$n.a.; n.a.; 10^{-1}$	

policed and marked using a single-rate Three Color Market (srTCM). The access links to domain boundaries are configured so that intradomain measurements are not affected.

The service-dependent AC rules are parameterized as specified in Table 2. We have considered three downstream SLS, one per service class. The choice of SLS rate shares, safety margins and QoS parameters thresholds are defined in the right hand side of Table 2. As shown, larger safety margins and tighter thresholds are defined for more demanding classes. The AC thresholds are set taking into account the domain topology dimensioning, queuing and propagation delays and perceived QoS upper bounds for common applications and services [12]. As shown in Table 2, SC1 traffic is blocked when the sum of the rate estimate $\tilde{R}^{+}_{i,(*,E_m)}$ and the flow's peak rate r_j is above 85% of the class rate share defined in SLS^{+}_{i,E_m}, i.e. R^{+}_{i,E_m}, or any of the controlled QoS parameters exceeds the pre-defined thresholds. For SC2, a safety margin of 90% was defined and the flow mean rate is now used. SC3 does not include any safety margin and the controlled parameter is IPLR.

5.2 Evaluation Results

Generically, 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, even for high network utilization levels. Fig. 3 illustrates the obtained mean IPTD, ipdv and IPLR of the service classes defined above for a measuring interval Δt_i of 5s. As shown, SC1, SC2 and SC3 exhibit a very stable behavior regarding the pre-defined QoS levels, in special, for delay related parameters. IPLR is more difficult to keep tightly controlled, however, IPLR deviations still stay well-bounded around the threshold defined. Note that, these results are particularly encouraging attending to the high overall network utilization obtained (see Fig. 4(b)). This Figure also illustrates that each class share is accomplished, with SC3 exceeding its share slightly. This occurs due to the adaptive nature of TCP traffic, combined with the traffic fluctuations of SC2. This aspect along with the inherent properties of the scheduling mechanism in use are responsible for most of QoS violations in SC2.

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

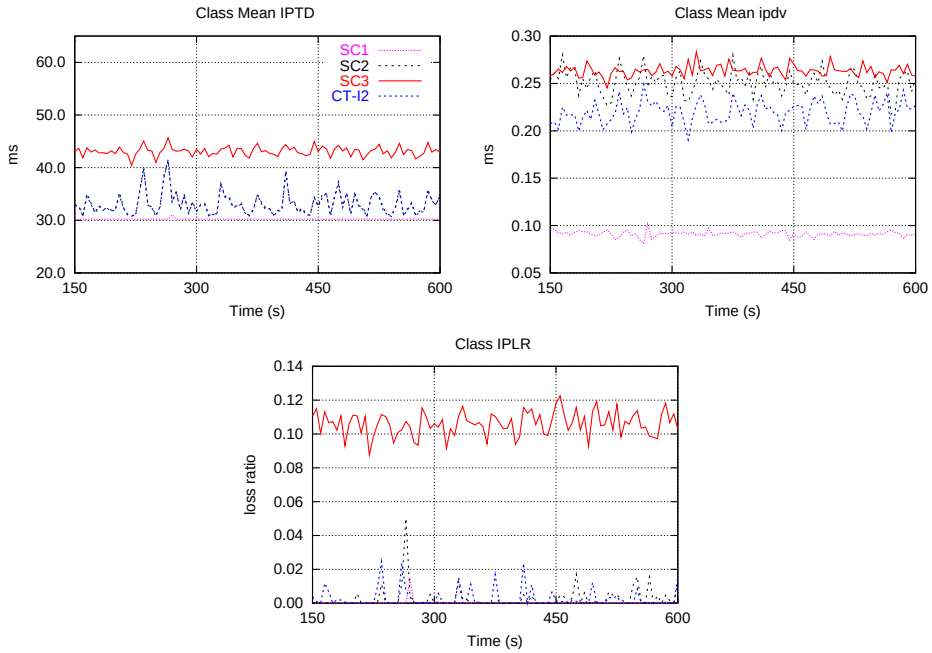

Figure 3: Class Mean IPTD, ipdv, IPLR for $\Delta t_i = 5s$

several measurement time intervals, while those flows last. To minimize this, more conservative estimates or larger safety margins might be required. We have explored the latter option, where new safety margins aiming at a service provisioning without QoS violations were established. In this way, Table 3 summarizes the results initially obtained for each class, extending them for the new defined safety margins. As shown, it includes: (i) the average number of concurrent active flows (# a.f); (ii) the corresponding utilization (U); (iii) the percentage of packets exceeding the pre-defined IPTD and ipdv delay bounds; (iv) the total loss ratio; (v) a new safety margin for which no QoS violations occur; (vi) the corresponding new number of active flows (# a.f.') and utilization (U'). In addition to the comments above, Table 3 results show that the percentage of QoS violations at packet level is very small, specially for class SC1, and the total IPLR is below the pre-defined thresholds. Under the new defined safety margins, the AC criteria, despite being more conservative, still achieve good network utilization.

In this evaluation process, we have also noticed that for implicit AC, the control of rate variables bring negative effects to SC3 stability and should be avoided. In fact, a criterion resorting to a rate-based $AC\_status_{\Delta t_i}$ and to $Adm\_flows_{\Delta t_i}$, which limits the number of active flows, lead to long AC blocking periods and to a resource take over by long-lived flows. Conversely, considering an $AC\_status_{\Delta t_i}$ determined by QoS control in addition to $Adm\_flows_{\Delta t_i}$ have proved to be mandatory in order to keep a "lively" number of active flows (see Fig. 4(a)), while satisfying the classes' QoS requirements.

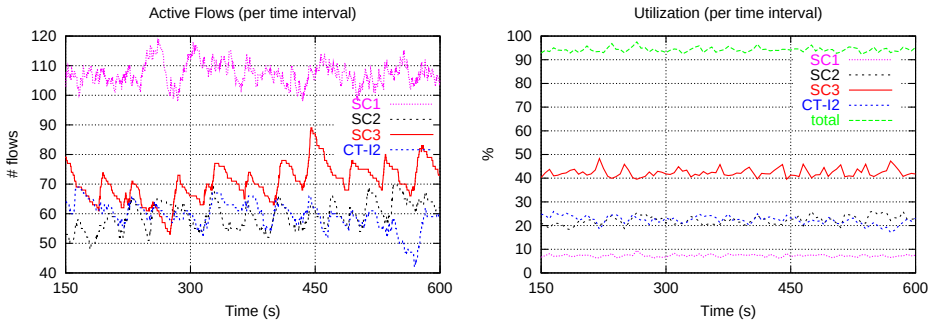

Figure 4: (a) Number of active flows for $\Delta t_i = 5s$; (b) Utilization

Table 3 Test Results

SC_i	# a.f.	U(%)	%viol(IPTD;ipdv)	total IPLR	$\beta'_{i,p}$	# a.f.'	U'(%)
SC1	107	7.4	(0.013 ; 0.001)	0.00017	0.80	100	6.9
SC2	59	22.4	(3.3 ; n.a.)	0.003	0.78	49	18.7
SC3	70	42.8	(n.a.; n.a.)	0.106	1.0	84	48.4
CT-I2	59	22.6	(3.1 ; n.a.)	0.003	0.78	53	20.0

6. Conclusions and Future Work

In this paper, we have specified a service-oriented distributed AC model for managing QoS and SLSs in multiclass and multidomain environments. Explicit or implicit AC decisions are made based on feedback from edge-to-edge on-line measurements of service-specific QoS parameters and SLS utilization. This allows a dynamic control of services and resources, while abstracting from network inherent complexity and heterogeneity. 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 domain. The results have shown that the proposed multiservice management scheme establishes a good compromise between simplicity and efficiency, allowing to satisfy effectively distinct service level commitments, while achieving a high network utilization. Despite properties of the AC model such as edge-to-edge QoS control, SLSs control embedded in the corresponding service class and reduced state information and signalling tend to increase the model resilience to scalability problems, future work intends to sustain an extend the obtained results to more complex scenarios involving multiple domains.

References

[1] J. Babiarz, K. Chan, and F. Baker. Configuration Guidelines for DiffServ Service Classes (work in progress). draft-baker-diffserv-basic-classes-04.txt, October 2004.

[2] N. Benameur et al. Integrated Admission Control for Streaming and Elastic Traffic. *QofIS'01*, volume 2156, pages 67–81, September 2001.

[3] S. Blake et al. An Architecture for Differentiated Services. IETF RFC 2475, 1998.
[4] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhang. Endpoint Admission Control: Architectural Issues and Performance. In *ACM SIGCOMM'00*, 2000.
[5] C. Cetinkaya, V. Kanodia, and E. Knightly. Scalable Services via Egress Admission Control. *IEEE Transactions on Multimedia*, 3(1):69–81, March 2001.
[6] J. Chen, A. McAuley, V. Sarangan, S. Baba, and Y. Ohba. Dynamic Service Negotiation Protocol (DSNP) and Wireless Diffserv. In *ICC'02*, April 2002.
[7] 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.
[8] V. Elek, G. Karlsson, and R. Rnngren. Admission Control Based on End-to-End Measurements. In *IEEE INFOCOM'00*, 2000.
[9] D. Goderis et al. Attributes of a Service Level Specification (SLS) Template IETF draft: draft-tequila-sls-03.txt (work in progress), October 2003.
[10] Alexander Keller and Heiko Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *Journal of Network and Systems Management, Special Issue on E-business Management*, 11(1), March 2003.
[11] Ibrahim Khalil and Torsten Braun. Edge Provisioning and Fairness in VPN-DiffServ Networks. *Journal of Network and Systems Management, Special Issue on Management of Converged Networks*, 10(1), March 2002.
[12] S. Leinen and V. Reijs. Geant D9.7 - Testing of Traffic Measurement Tools. Geant Project, September 2002.
[13] S. Lima, P. Carvalho, and V. Freitas. Distributed Admission Control for QoS and SLS Management. *Journal of Network and Systems Management - Special Issue on Distributed Management*, 12(3):397–426, September 2004.
[14] S. Lima, P. Carvalho, and V. Freitas. Measuring QoS in Class-based IP Networks using Multipurpose Colored Probing Patterns. In *ITCom 2004*. SPIE, 2004.
[15] S. Lima, P. Carvalho, 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.
[16] P. Morand et al. 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.
[17] 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.
[18] S. Salsano et al. Definition and Usage of SLSs in the Aquila consortium. IETF draft: draft-salsano-aquila-sls-00.txt (work in progress), November 2000.
[19] A. Sevasti and M. Campanella. Geant D9.1 - Service Level Agreements Specification for IP Premium Service. Geant and Sequin Projects, October 2001.
[20] I. Stoica and Hui Zhang. Providing Guaranteed Services Without Per Flow Management. In *ACM SIGCOMM'99*, October 1999.
[21] B. Teitelbaum et al. Internet2 QBone: building a testbed for differentiated services. *IEEE Network*, 13(5):8–16, 1999.
[22] P. Trimintzios et al. An Architectural Framework for Providing QoS in IP Differentiated Services Networks. In *7th IFIP/IEEE International Symposium on Integrated Network Management - IM'01*, 2001.