Toward Scalable Management of Multiple Service Levels in IP Networks

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

Abstract

This paper analyzes and discusses the role of a distributed and simple admission control (AC) model in achieving scalable management of multiple network service levels. The model design, covering explicit and implicit AC, exhibits relevant properties which allow managing QoS and SLSs in multiservice IP networks in a flexible and scalable manner. These properties stem from the way service-dependent AC and on-line service performance monitoring are proposed and articulated in the model's architecture and operation. The scalability debate, carried out at these two levels, highlights key steps toward self-adaptive service-oriented AC and low overhead multiservice monitoring. The performance evaluation results, illustrating the role and relevance of the defined AC rules, show that QoS and SLSs requirements can be efficiently satisfied or bounded, proving that the simplicity, flexibility and self-adaptability of the model can be explored to manage multiple service guarantees successfully.

1. Introduction

Managing multiservice networks is a complex and multidimensional problem involving heterogeneous media, protocols and technologies. Achieving an uniform and ubiquitous management solution is even a more intricate issue attending to the plethora of service providers with their own business, management and technological strategies.

The provision of adequate network services facing the negotiated Service Level Specifications (SLSs), on the one side, and the effective control of incoming traffic aggregates facing the available network resources, on the other side, motivates the use of Admission Control (AC) mechanisms to keep service classes under controlled load and assure the required QoS levels. This can be particularly useful for services such as IP telephony and video conferencing [1]. The complexity introduced in the network control plane should however be kept as low as possible in order to allow for scalable service deployment and management.

To face this challenge, the distributed AC model introduced in [12, 13] aims to: (i) support multiple service assurance levels; (ii) control QoS levels inside each domain and existing SLSs between domains; (iii) operate intra and interdomain providing an unified end-to-end solution; (iv) be simple, flexible, efficient, scalable and easy to deploy in real environments. The above design goals are particularly relevant when considering the deployment of the model in a large scale, across multiple administrative domains relying, eventually, on distinct solutions regarding service offering and provisioning.

This paper is focused on examining and discussing the model properties toward a scalable multiservice management, highlighting the AC criteria effectiveness in satisfying multiple QoS and SLSs commitments simultaneously. Multiservice management scalability is here covered at two levels: first, the AC model design includes specific properties which tend to increase the model resilience to scalability problems while allowing a self-adaptive and service-oriented management of QoS and SLSs; second, as the model relies on on-line monitoring feedback, innovative approaches such as embedding the QoS control of flows and SLSs at class level and using multipurpose active monitoring also brings potential advantages toward scalability. Additionally, the model's architecture building blocks are presented and the AC criteria effectiveness is assessed under high demanding traffic conditions in order to evince the relevance and applicability of the defined AC rules in satisfying multiple service guarantees efficiently.

The remaining of this paper is organized as follows: the related work and motivation for the present study are debated in Section 2; the AC model is covered in Section 3, focusing its main underlying ideas, architecture, decision rules and major features toward scalability; the test platform and model performance results are included Section 4, finally, the conclusions are presented in Section 5.

2. Related work and motivation

Existing AC approaches are either centralized or distributed, following a parameter or measurement-based decision criterion depending on the type of services supported.

The main advantage of centralized AC approaches [5,16] is that centralizing state information and control tasks allow a global vision of the domain's QoS and operation, relieving the control plane inside the network. However, central entities need to store and manage large amounts of information, which in large and highly dynamic networks with many signaling messages and information state updates needing to be processed in real-time is even hard or prohibitive. The congestion and functional dependence on a single entity is another well-known problem of centralization.

To improve reliability and scalability in large networks, several approaches consider distributed AC with variable control complexity depending on the QoS guarantees and predictability required. To provide guaranteed services (e.g., for hard real-time traffic), AC proposals tend to require significant network state information and, in many cases, changes in all network nodes [15, 17]. To provide predictive services (e.g., for soft real-time traffic) measurement-based AC (MBAC) [2, 10] and end-to-end MBAC solutions [4, 6] have deserved special attention. These solutions leads to reduced control information and overhead, but eventually to QoS degradation. To control elastic traffic, for efficient network utilization, implicit AC strategies have also been defined [7,14].

In these studies, aspects such as the trade-off between service assurance level and network control complexity for a scalable and flexible support of distinct service types and corresponding SLSs, intra and interdomain, are not covered or balanced as a whole. The AC model discussed in this paper extends the former studies by focusing on achieving a flexible and encompassing solution toward a scalable management of multiservice networks able to deal with the management of multiple intradomain QoS levels and interdomain SLSs simultaneously.

3. Managing service levels through AC

3.1. The AC model underlying ideas

An underlying idea of the proposed AC model is to take advantage of the need for on-line QoS and SLS monitoring in today's class-based networks and use the resulting monitoring information to perform distributed AC. This monitoring process carried out on a per-class and edge-to-edge basis allows a systematic view of each service class load, QoS levels and SLSs utilization in each domain, while facilitating SLSs auditing tasks. Performing AC at edge nodes using this feedback simplifies the network control plane, allows to make decisions on new flows' admittance with minimum latency and, generically, allows to manage QoS and SLSs. Moreover, when monitoring is carried out edge-to-edge, internal network control, topology and technologies are hidden from AC. Although such approach allows high abstraction from network core complexity and heterogeneity, network traffic dynamics and QoS can yet be sensed and updated continuously through proper service metrics. In this model, QoS control is only performed at class level instead of SLS or flow level, and SLSs control is reduced to SLS utilization control. This aspect is fundamental to alleviate the amount of state information and control overhead, increasing the model scalability.

Another important underlying idea toward model simplicity and flexibility is to consider a service-dependent degree of overprovisioning controlled by the AC rules. This is a relevant aspect to achieve a simple and manageable multiservice AC solution as it allows to relax the AC process, while widening the range of service types covered by a monitoring-based AC solution.

In brief, in the model's operation illustrated in Figure 1, while ingress nodes perform implicit or explicit AC resorting to service-dependent rules for QoS and SLS control (see Section 3.3), egress nodes collect service metrics providing them as inputs for AC. When spanning multiple domains, collecting and accumulating the QoS measures available at each domain edge nodes will allow to compute the expected end-to-end QoS[1].

3.2. AC model architecture

The AC model architecture lays on service definition, AC criteria, QoS/SLS monitoring and traffic characterization building blocks, interrelated as shown in Figure 2.

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

---

[1] The functionality and versatility of signaling protocols within NSIS framework [9], in special, the sender-initiated path-coupled case where signaling messages are routed and processed only at specific nodes in the data path is particularly suitable to support the AC model operation.

In addition, it should be noticed that a cumulative process for end-to-end QoS computation is consistent with the cascade approach for the support of interoperator IP-based services, which has the merit of being more realistic, i.e., in conformance with the Internet structure and operation, and more scalable than the source-based approach [8].

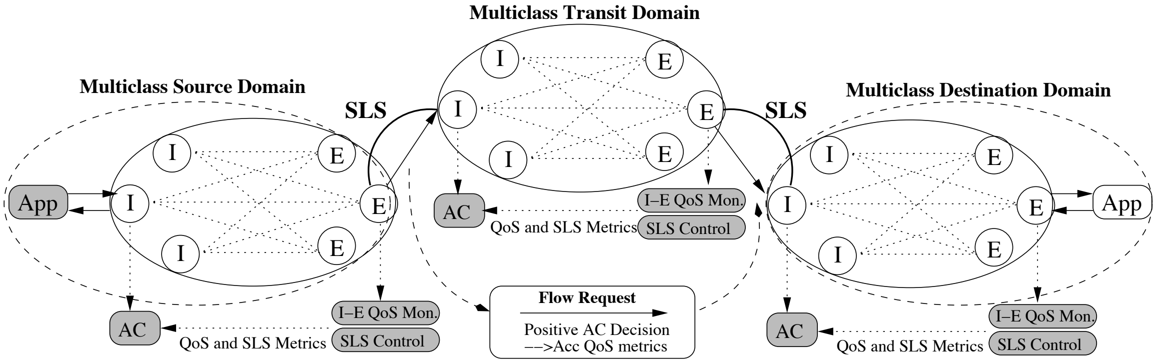

Figure 1. Distributed monitoring-based AC approach

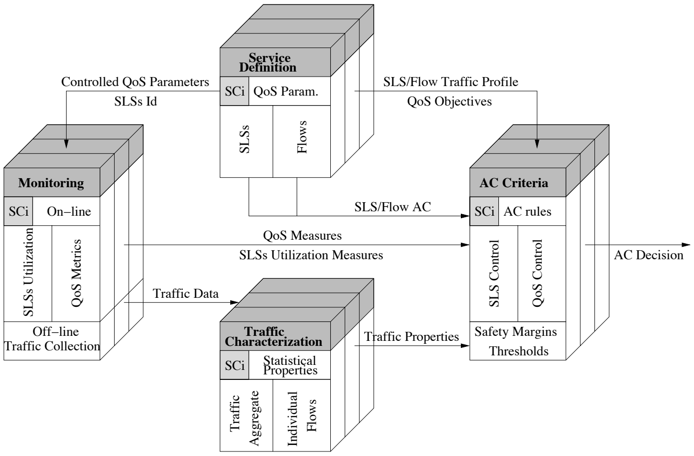

Figure 2. AC model architecture

hybrid AC algorithms provides the basics for defining a multiservice *AC decision criteria.*

3.3. AC criteria specification summary

Following the specification provided in [13] and the notation in Table 1, the AC criteria comprises: (i) rate-based SLS control rules; (ii) QoS parameters control rules.

Rate-based SLS control rules For a service class SC_i under *explicit AC*, verifying if a new flow $F_j \in SLS_{i,I_n}$ can be admitted at each ingress node I_n involves testing if the negotiated rate R_{i,I_n} for SLS_{i,I_n} can accommodate the new flow traffic profile, i.e.

$$\tilde{R}_{i,(I_n,*)} + r_j \le \beta_{i,I_n} R_{i,I_n} \qquad (1)$$

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

$$\tilde{R}^{+}_{i,(*,E_m)} + r_j \le \beta^{+}_{i,E_m} R^{+}_{i,E_m} \qquad (2)$$

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

For a service class SC_i under *implicit AC*, as flows are unable to describe r_j, traffic flows are accepted or rejected implicitly according to the value of a variable $AC\_Status_{\Delta t_i}$ computed once for the measurement interval Δt_i (see Eq. (3)).

QoS parameters control rules At each ingress node I_n, the admission of new flows in Δt_i is determined by

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

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

Table 1. Model notation summary

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

restores the QoS target values. This will only be checked at Δt_{i+1}. The end-to-end case is specified in [13].

3.4. Model key points toward scalability

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

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

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

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

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

(iv) the AC model provides enough flexibility to accommodate the evolution of technologies, services and applications. Important aspects contributing to the model's flexibility are the service-dependent nature of AC rules and the conceptual modular independence between AC and monitoring tasks.

Scalable QoS and SLS monitoring - the major key points identified are the following:

(i) the control of the SLSs' negotiated QoS parameters is embedded in the QoS control of the corresponding service classes, reducing the amount of SLSs' dynamic state information and control overhead. At SLS level, the traffic load is the only parameter measured locally at I_n or E_m nodes. In more detail, considering the set of the expected QoS parameters of each SC_i, SLS_{i,I_n} and F_j respectively, accepting SLSs and flows based on the subset inclusion rule $P_{F_j} \subseteq P_{SLS_{i,I_n}} \subseteq P_{SC_i}$ is of crucial importance regarding the scalability of the control strategy. This means that each $P_{j,p}$ value must be bounded by the corresponding $P_{i,I_n,p}$ value (if applicable, i.e. if $F_j \in SLS_{i,I_n}$) which, in turn, must be bounded by the corresponding class target value $P_{i,p}$. 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.

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

(iii) the use of multipurpose active monitoring, i.e., the use of light probing patterns able to capture simultaneously the behavior of multiple QoS metrics of each class, also brings potential advantages to scalability [11].

The model properties defined above tend to increase the model's resilience to scalability problems. A summary identifying the impact that large-scale environments may have on the proposed AC solution is highlighted in Table 2.

Table 2. Issues on the AC model scalability

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

4. Evaluating the model performance

To evaluate the AC model's ability to manage multiple service commitments efficiently in a multiclass environment, a simulation prototype was set up using NS-2.

Defined service classes Considering current service configuration guidelines [1], three service classes are defined. As basic policy, TCP and UDP traffic are treated separately; UDP traffic is further divided according to its QoS requirements. Table 3 summarizes the service classes implemented, highlighting AC and QoS monitoring decisions and parameters used to configure the AC rules controlling both SLS utilization and domain QoS levels. The negotiated rates (R^{+}_{i,E_m}) of downstream SLSs have been defined according to the traffic load share intended for the corresponding class in the domain. As shown, the parameterization of the AC rules is service-dependent and larger β^{+}_{i,E_m} and tighter $T_{i,p}$ are defined for more demanding classes. For instance, a $\beta^{+}_{i,E_m} = 0.85$ corresponds to impose a safety margin or degree of overprovisioning of 15% to absorb load fluctuations and optimistic measures. The AC thresholds $T_{i,p}$ considers domain's characteristics and perceived QoS upper bounds for common applications and services.

Domain topology The network domain consists of ingress routers I_1, I_2, a multiclass network core and an egress router E_1. The service classes SC1, SC2 and SC3 are implemented in all the domain nodes. I_2 is used to inject concurrent or cross traffic (referred as CT-I2), allowing to evaluate concurrency effects on distributed AC and assess the impact of cross traffic on the AC model performance. The scenarios with cross traffic allow to contemplate the presence of unmeasured traffic within the core, having an impact on the domain's QoS and load but without being explicitly measured by E_1 SLS rate control rules (see Figure 3). The domain routers implement the service classes according to a hybrid Priority Queuing - Weighted Round Robin (PQ-WRR(2,1)) scheduling discipline, with RIO-C as AQM mechanism. Each class queue is 150 packets long. The domain internodal links capacity is 34Mbps, with a 15ms propagation delay.

4.1. Performance results

An initial assessment of the explicit and implicit AC criteria has demonstrated that in presence of concurrent traffic, the self-adaptive behavior inherent to on-line monitoring combined with the established AC rules was effective in controlling each class QoS and SLS commitments [13].

This paper explores the behavior of the AC criteria under more demanding traffic conditions such as the presence of cross-traffic. This aspect is of major relevance as, in large scale environments, due to the internal traffic dynamics and topology characteristics, some traffic may constitute an additional load just in parts of an (I_n, E_m) path without being accounted for in the corresponding SLS^{+}_{i,E_m} (see Figure 3). The present tests consider that the traffic injected into I_2 is cross-traffic, hence, E_1 is not aware of it apart from the impact it may have on QoS estimation. The AC criteria ability to self-adapt to new QoS thresholds is also explored, identifying the most critical QoS parameters to control.

An additional set of tests regards studying other important aspects which may impact on the model's behavior, namely, testing the influence of network/service traffic characteristics and of measurement time interval Δt_i dimension. Thus, to complement the study with a default Δt_i of $5s$, larger intervals are tested.

The performance evaluation tests focus on: (i) verifying if QoS parameters are in conformance with the established QoS levels; (ii) quantifying QoS violations, at class and packet level; (iii) evaluating each class blocking probabilities; (iv) measuring the utilization level of each class individually and of the network domain globally, verifying the conformance of each SLS rate share (R^{+}_{i,E_m}).

Table 3. Service Classes SC_i

SC_i	Serv. Type	AC Type	R^{+}_{i,E_m}	β^{+}_{i,E_m}	$P_{i,p}$	$T_{i,p}$	Example	Traffic Src	i.a.t.	hold.t
SC1	guaranteed (hard-RT)	explicit and conservative	3.4Mbps (10% share)	0.85	IPTD ipdv IPLR	35ms 1ms 10^{-4}	VoIP Cir.Emulation Conv. UMTS	Exp. or Pareto on/off (64kbps, pkt=120B on/off = 0.96/1.69ms)	Exp. 0.3s	Exp. 90s
SC2	predictive (soft-RT)	explicit and flexible	17Mbps (50% share)	0.90	IPTD IPLR	50ms 10^{-3}	audio/video streaming	(256kbps, pkt=512B on/off = 500/500ms)	0.5s	120s
SC3	best-effort	implicit	13.6Mbps	1.0	IPLR	10^{-1}	elastic apps.	FTP traffic (pkt=512B)	0.5s	180s

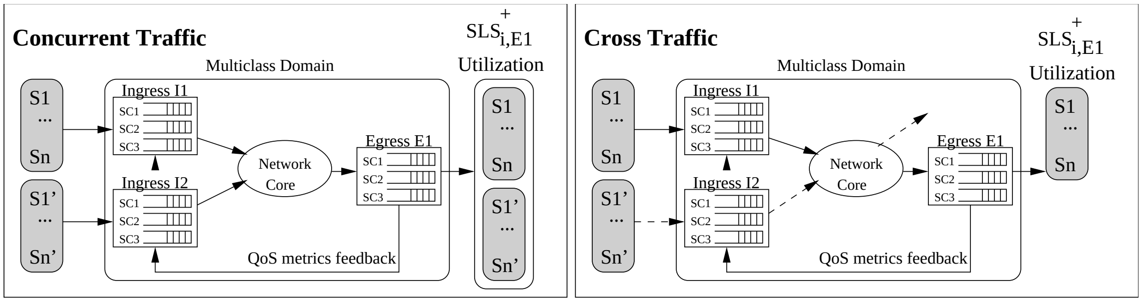

Figure 3. Concurrent vs. cross traffic for SCi

Overview of results with concurrent traffic - When examining in detail which AC rules determine the generic behavior of the model discussed in [13], the following has been identified: (i) SC1 flows are controlled essentially by the SLS rate control rule (Eq. 2) as a result of a stable QoS behavior associated with this high priority class; (ii) AC for SC2 flows is triggered by SLS and/or QoS control rules (Eq. 2 and Eq. 3); (iii) SC3 flows are mostly controlled by the QoS control rule; (iv) according to the results, IPLR violations assume a predominant role in setting the variable $AC\_status_{\Delta t_i}$ to a rejection mode in the QoS control rule. Nevertheless, the percentage of QoS violations at packet level for the controlled QoS metrics is very small and the total IPLR is kept within the pre-defined thresholds. These are very encouraging results attending to the high global utilization (U) achieved, where each class rate share is well accomplished (see Table 4).

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

(i) In the presence of SC2 cross traffic, the main rule determining AC decisions in this class is the QoS control rule, with $AC\_status_{\Delta t_i} = reject$ activated by IPLR violations. This rule by itself maintains the QoS levels controlled, as shown in Figure 4.

Table 4. Results and statistics at packet level

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

The SLS rate control rule and the corresponding safety margins are now less relevant and restrictive. The global utilization of SC2 (I_1+ CrossTraffic) decreases slightly comparing to the concurrent case, with the amount of traffic accepted at I_1 being adjusted according to the amount of cross traffic. This decrease is a consequence of the effect of cross traffic on $C1$ queue occupancy increasing loss events and triggering the QoS control rule more frequently. However, as shown in Figure 5(a), the rate share of each class is well accomplished and the global utilization very high. SC3 exceeds slightly its defined rate share, taking advantage of SC1/SC2 unused bandwidth resources, due to the work conserving nature of the traffic scheduler. The packet level analysis reveals that $\%pkt\_violations$ on IPTD is 0.05 and 12.8, for 10% and 20% of cross traffic (i.e., up to 40% of the SC2 class share) respectively, and $Total\ IPLR$ is 0.005 and 0.008 (of the same order of magnitude of the defined threshold).

This test has also included experiments with distinct QoS thresholds. As an example, Figure 5 (b) shows the model's ability to control delay and loss bounds. When a tighter IPTD threshold of 35ms is set for SC2, AC is effective in bringing and maintaining IPTD controlled around that value. The same occurs when an IPLR threshold of 0.05 is set for SC2 and SC3 (more relaxed and tight than the previous one, respectively).

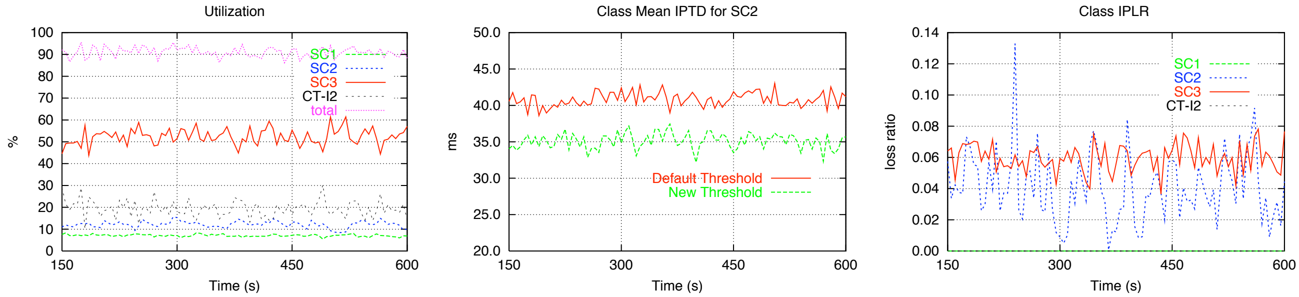

Figure 5. (a) Utilization for 20% of SC2 cross traffic; (b) IPTD and IPLR adaptation to new thresholds

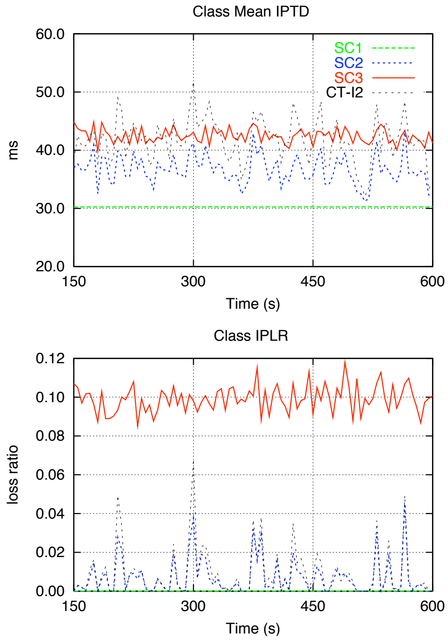

Figure 4. Class mean IPTD and IPLR for 20% of SC2 cross traffic

(ii) In the presence of cross traffic from class SC1, numerous QoS violations in IPTD, ipdv and IPLR become evident and difficult to control despite the rejection indication provided by the QoS control rule. This is due to high traffic fluctuations and to the nature of the scheduling mechanism, which has defined a Max-EF-Rate for PQ treatment. In the presence of an excessive rate at C_1, unmeasured and uncontrolled by E_1, several blocking events may occur at the scheduler affecting SC1 traffic. The QoS control rule, detecting these violations, sets $AC\_Status_{\Delta t_i}$ to rejection mode. However, the effect of flows already accepted within the previous acceptance period (bounded by the rate control rule with $\beta^{+}_{i,E_m} = 0.85$), along with the cross traffic load, leads to QoS degradation that may span more than one Δt_i. To minimize this, more conservative estimates, larger safety margins and/or specific approaches to control concurrency, may be required. As regards defining larger safety margins, as an example, for 2.5% of SC1 cross traffic (i.e., 25% of the class share), $\beta^{+}_{i,E_m} = 0.5$ allows an SC1 behavior without QoS violations[2]. A simple ISPs design rule for tight delay, jitter and loss control is provisioning twice the capacity of the expected aggregate peak load [3].

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

From these set of experiments, the relevance of the defined AC rules becomes evident for assuring service commitments in the domain. While the rate control rule assumes a preponderant role for service classes SC1 and SC2 to control the traffic load and indirectly QoS, particularly in situations involving concurrent traffic, the QoS control rule is decisive to assure the domain QoS levels in presence of unmeasured cross traffic. In real environments, where the two type of situations are likely to occur simultaneously, the two AC rules will complement each other to increase the domain capabilities to guarantee service commitments. Although being encouraging on this aspect, the obtained results might be even more satisfactory when considering that a significant amount of the involved cross traffic will be sensed and controlled by other egress nodes.

From the above reasoning, it is important to remark that, knowing which AC rule is more influent on the AC decision process can also bring relevant information and directions

---

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

for improving service configuration and provisioning both intra and interdomain.

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

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

In the present context, maintaining an AC parameterization similar to Test1 (i.e., the safety margins and thresholds), several experiments were carried out to evaluate the impact of different types of sources on the performance of the AC proposal. In this way, in addition to $EXPOO$ sources, CBR and $PAROO$ sources were included in the tests, as illustrated in Table 5. Pareto sources with a shape parameter $1 < \alpha < 2$ under aggregation allow to generate traffic exhibiting LRD.

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

AC fairness on concurrent flows - In order to analyze the model behavior in the presence of concurrent traffic with distinct flow characteristics within the same service class several tests were carried out. Initial results show that the model is able to adapt consistently to different conditions in the concurrent classes, adjusting the number of admitted flows according to the flows' defined rate and maintaining the global and per-class utilization levels similar to the ones obtained previously.

The results in Table 6 illustrate this fair behavior when the concurrent class is SC1 with more demanding flow peak rates, burstiness and flow arrival/holding times[3]. Under the new traffic conditions, the QoS behavior of SC1 shows a slight degradation. However, the *% pkt_ violations* is very low and $Total\ IPLR$ is kept well bounded within one order of magnitude above the established QoS thresholds. IPLR behavior in Δt_i is illustrated in Figure 6. The cause of QoS degradation is the higher fluctuations in the rate estimations when SC1 flows' rate is increased, irrespectively of the concurrent traffic having or not similar characteristics. The QoS degradation noticed can be avoided resorting to a higher safety margin in the SLS rate control rule for SC1. As illustrated in Table 6, the remaining service classes are not particularly affected by the new test conditions.

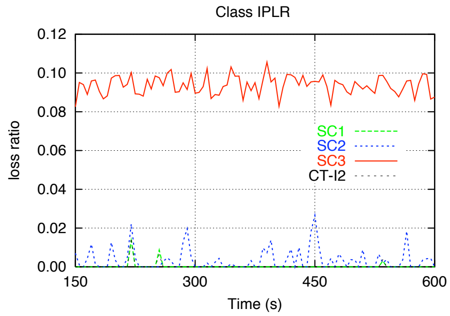

Figure 6. IPLR behavior (CT-I2 = $EXPOO^2_{SC1}$)

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

According to the obtained results, maintaining the default test conditions, the major impact of increasing Δt_i (creating consequently a longer "blind" period regarding the real network status) is to create a cyclic AC status behavior affecting the number of active flows and utilization of each class (see Figures 7 (a) and (b))[4]. The classes' QoS

---

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

[4] In Figure 7, *Target* line represents the value $\beta^+_{i,E_m} R^+_{i,E_m}$ above which AC rejection occurs, *Estimate* line represents the estimated rate or load of SLS^+_{i,E_m}, i.e., $\tilde{R}^+_{i,(*,E_m)}$, and *Total* line reports to the previous estimate by adding the new flow rate r_j. *Decision* dots represent a posi-

Table 5. AC results for distinct source models

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

Table 6. AC results on fairness

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

commitments are easily met for higher Δt_i, as result of an utilization decrease. This means that the QoS behavior of these service classes for $\Delta t_i = 30s$ and $\Delta t_i = 60s$ is better than for $\Delta t_i = 5s$, both from a measurement interval and packet level perspectives. SC3 follows similar trends to the tests using smaller Δt_i. Despite the good QoS results achieved, for larger Δt_i the AC rejection period may be excessive. The cyclic behavior exhibited in Figure **??** is also stressed by the demanding characteristics of the flow arrival process; under more moderate flow arrival conditions, that behavior tends to smooth and the evolution of active flows and utilization become more regular.

In more detail, considering SC1 and $\Delta t_i = 30s$ as an example (see Figure 7), it is visible that after each load estimation update, the system enters into a positive AC cycle with a slope that depends on flow inter-arrival. After each flow admission, each I_n will update the load estimate until detecting that the new acceptances lead to the defined utilization target. In that moment, new incoming flows start to be refused and the last estimation is kept until Δt_{i+1}, when a new load is estimated and provided. As flow departures within a time interval are not taken into account, when the new update takes place, the rate estimation at the ingress node tends to decrease abruptly. Thus, updating the rate estimates at each I_n according to the mean or peak rate of accepted flows leads to a more conservative AC as new incoming rates are considered without pondering the compensation effect of departing flows. This effect tends to be more notorious when Δt_i increases as the I_n estimation update reflecting the real network conditions, sent by the monitoring module, is provided later. Keeping rate estimates ($\tilde{R}^{+}_{i,E_m}$) unchanged during Δt_i irrespective of flows acceptance, explores this compensation effect but may increase overacceptance and lead to more QoS violations in all the service classes.

In summary, considering the test scenarios presented previously, a smaller Δt_i may be preferable to take advantage of the good compromise among network utilization, QoS and stability. Dimensioning Δt_i also involves establishing a trade-off between the overhead of the metrics' update process and the accuracy of capturing the real network status. Therefore, developing a light, effective and reliable process for computing and disseminating QoS metrics in real environments is an aspect requiring further study.

5. Conclusions

This paper discusses how scalable management of multiple network service levels can be accomplished resorting to a distributed and simple AC model. The key points

---

tive (dots above the x-axis) or negative (dots overlapping the x-axis) AC decision, considering also the QoS control rule evaluation.

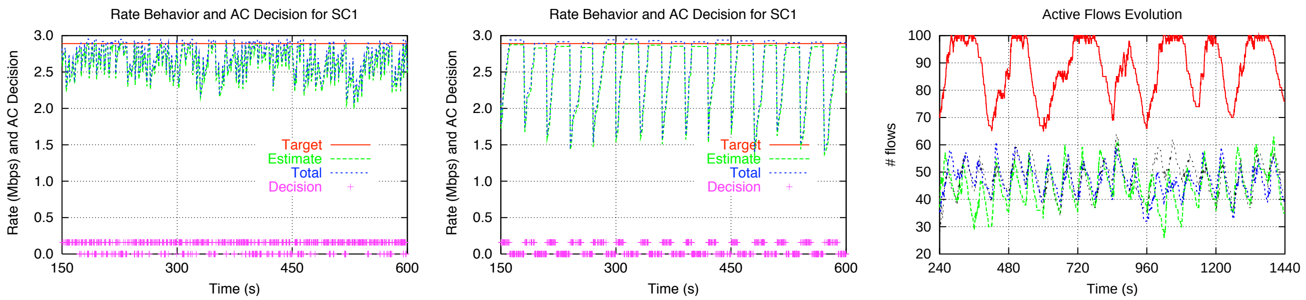

Figure 7. (a) Influence of Δt_i on rate and AC behavior for SC1: $\Delta t_i = 5s$ and $\Delta t_i = 30s$; (b) Active flows for $\Delta t_i = 60s$ adjusting $\tilde{R}^{+}_{i,E_m}$

of this model toward a scalable control of QoS and SLSs both intra and interdomain have been debated. In particular, distributing control between edge nodes, relieving network core from control tasks, reducing state information and control overhead, sensing and adapting to network dynamics through measurements, supporting AC irrespectively of applications' ability to explicit QoS requirements and signaling the network, are relevant aspects for deploying the strategy in heterogeneous large scale environments. To improve service monitoring scalability, QoS control is only performed at class level instead of SLS or flow level.

The evaluation of the model's performance has showed that the proposed AC model, using a two-rule AC criterion defined on a service class basis, has been able to assure service level guarantees and achieve high network utilization, without adding significant complexity to the network elements. The use of systematic edge-to-edge monitoring and a controlled degree of overprovisioning revealed essential design aspects contributing to achieve a simple and self-adaptive solution for managing multiple service levels.

References

[1] J. Babiarz, K. Chan, and F. Baker. Configuration Guidelines for Diffserv Service Classes. RFC 4594, Aug. 2006.

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

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

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

[5] 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, Feb. 2004.

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

[7] S. Fredj, S. Oueslati-Boulahia, and J. Roberts. Measurement-based Admission Control for Elastic Traffic. In *17th Int. Teletraffic Congress*, June 2001.

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

[9] R. Hancock, G. Karagiannis, J. Loughney, and S. V. den Bosch. Next Steps in Signaling (NSIS): Framework. RFC 4080 (Informational), June 2005.

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

[11] S. Lima, P. Carvalho, and V. Freitas. Measuring QoS in Class-based IP Networks using Multipurpose Colored Probing Patterns. In *SPIE ITCom 2004*, volume 5598, pages 171–182, Philadelphia, U.S.A., October 2004.

[12] S. R. Lima, P. Carvalho, and V. Freitas. Distributed Admission Control for QoS and SLS Management. *Journal of Network and Systems Management - Special Issue on Distributed Management*, 12(3):397–426, Sept. 2004.

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

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

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

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

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