Improving QoS Guarantees through Implicit AC

Solange Rito Lima, Pedro Sousa, Paulo Carvalho

University of Minho, Department of Informatics

4710-057 Braga, Portugal

Email:{solange,pns,pmc}@di.uminho.pt

Abstract—In multiservice networks, admission control (AC) is a convenient means of assuring high quality communications by safeguarding enough availability for customer traffic. This can be particularly useful to preserve the quality of services such as IP telephony and video conferencing, and to ensure acceptable throughput to elastic flows. This paper tackles the problematic of performing implicit AC in multiservice networks, pointing out a flexible yet simple to deploy solution for controlling flows which do not explicitly send signaling admission requests. This allows to complement the explicit AC case, widening the ability to integrate services and applications in a transparent way. The versatility and self-adaptability of the proposed distributed AC criteria in ensuring the quality of multiple services is also proved.

I. INTRODUCTION

It is commonly accepted that Admission Control (AC) plays a key role in preventing instability and congestion, assuring QoS and Service Level Specification (SLS) fulfillment. Depending on the application or service characteristics, explicit and implicit AC can be in place. While explicit flow AC is oriented to applications able to signal the network with their traffic profile and QoS objectives, implicit flow AC is required to control applications which are unable to provide that signaling information and, in particular, to elastic applications. Although performing AC for real-time traffic, such as IP telephony and video conferencing services, is generically consensual [1], the need to control the admission of elastic TCP traffic is more arguable, dividing opinions. While some argue that once TCP is adaptive controlling the number of flows sharing the available bandwidth is unnecessary, we subscribe the argument that may be required to control network overload in order to preserve an acceptable throughput per active flow, and thus, the QoS offered to users [2]–[5].

Despite implicit AC has been matter of study in the literature (see Section II), tackling the problematic of implicit AC in multiservice IP networks, where distinct service guarantees need to be provided, is a topic requiring further study. This paper, proposing a flexible AC solution to handle implicit and explicit AC in multiclass environments, contributes to demystify the problematic of implicit AC, and generically, shedding light on multiservice AC tasks as a whole.

This paper contents is organized as follows: current related work on implicit AC is debated in Section II; an overview of the AC model architecture and operation is given in Section III; the AC criteria specification, including the proposed implicit AC algorithms is included in Section IV; the proof-of-concept and performance results are provided in Section V; finally, the main conclusions are summarized in Section VI.

II. RELATED WORK ON IMPLICIT AC

The implicit AC concept can be applied either to TCP or UDP flows that do not use AC signaling. Considering the case of real-time UDP applications and services requiring per-domain and end-to-end service guarantees but without generating explicit admission requests, the flow detection process may trigger internally an explicit signaling process. In this scenario, the flow profile and QoS requirements for this explicit AC process have to be inferred from the flow type or high-level signaling protocols [6], [7].

As regards TCP, a minimum TCP bandwidth is required to achieve a minimal session level user utility [3] and the use of AC will assure that purpose, avoiding wasting network resources on retransmissions and incomplete transfers [2]. Due to the large number of TCP flow arrivals and their eventual small duration, controlling individual flows using explicit signaling and reservations is impracticable, therefore, a measurement-based AC approach for elastic traffic is more appropriate to assure that the solution is able to react and scale properly. Without per-flow signaling, the detection and acceptance/rejection of a new flow is made implicitly. Within implicit AC the simple discard of initial flow packets is usually enough to inform the source of a rejection decision, otherwise those packets will proceed. In more detail, possible solutions to support detection and corresponding AC decision are: (i) to detect and drop packets initializing the TCP connection (TCP SYN and/or SYN ACK) [3]; (ii) to maintain a list of accepted and active flows based on the corresponding flow identifiers [4]. While the former solution is easy to implement, the latter may be more flexible but critical for high-speed interfaces due to its potential overhead.

Following this discussion, this paper is focused on proposing and tuning a lightweight AC criterion able to regulate the implicit admission of incoming TCP flows requiring minimum throughput guarantees and real-time UDP flows requiring distinct QoS assurance levels. These signaled or inferred QoS levels guide an explicit AC process.

III. AC MODEL ARCHITECTURE OVERVIEW

In order to pursuit design goals such as flexibility, scalability and easy deployment, the AC model architecture proposed in [9] comprises: (i) distributed control between edge nodes; (ii) no control tasks within the network core; (iii) reduced state information and control overhead; (iv) measurement-based self-adaptive behavior regarding network dynamics.

This model, oriented to accommodate multiple services, intends to allow AC irrespectively of the applications' ability to signal the network. In the model's operation, while ingress nodes perform implicit or explicit AC resorting to service-dependent rules for QoS and SLS control (see Section IV), 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. This cumulative process for end-to-end QoS computation is consistent with the cascade approach for the support of interoperator IP-based services, which is in conformance with the Internet structure and operation, and is more scalable than the source-based approach [8].

IV. SPECIFYING THE MULTISERVICE AC CRITERIA

For controlling both the QoS levels in the domain and the utilization of existing SLSs, the following rules have been defined: (i) rate-based SLS control rules; (ii) QoS parameters control rules; (iii) end-to-end QoS control Rules. The specification of these rules, following the notation in [9], is presented in Table I. The conformance of the defined rules determine the acceptance of a new flow F_j. Note that Eq. (3) is not flow dependent, i.e. it is checked once during Δt_i to determine $AC\_Status_{\Delta t_i}$. An $AC\_Status_{\Delta t_i} =$ accept indicates that the measured QoS levels for SC_i are in conformance with the QoS objectives and, therefore, new flows can be accepted. An $AC\_Status_{\Delta t_i} =$ reject indicates that no more flows should be accepted until the class recovers and restores the QoS target values, which will only be checked at Δt_{i+1}. 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 $AC\_Status_{\Delta t_i}$.

Algorithm 1 allows to handle implicit AC requests, i.e. after detecting a new flow F_j implicitly, an AC decision is made at ingress node I_n. For a domain D_x, service class SC_i and (I_n, E_m) pair, algorithm 2 determines the QoS acceptance status for Δt_i according to the QoS measures provided by egress node E_m (Eq. 3). This acceptance status is updated once each Δt_i interval, but other scenarios may be devised.

V. TESTING IMPLICIT AC

To evaluate the AC model's ability to manage multiple service commitments efficiently in a multiclass environment, a simulation prototype was devised and set up using NS-2. This prototype implements three functional interrelated modules - Automatic Source Generation Module, AC Decision Module, and QoS and SLS Monitoring Module. Considering an (I_n, E_m) pair, Figure 1 presents a simplified diagram of the simulation model architecture, including the relation between these modules and main underlying functions and variables. The two modules represented in gray are recursive being responsible for the dynamic behavior of traffic source generation and monitoring.

Taking into consideration current service configuration guidelines [1], three initial service classes are defined and

TABLE I
CONTROL RULES SUMMARY

TYPE OF RULE	DESCRIPTION
SLS Rate Control Rules	Verify upstream and downstream SLSs utilization
$\check{R}_{i,(I_n,*)} + r_j \leq \beta_{i,I_n} R_{i,I_n}$ (1)	$\check{R}_{i,(I_n,*)}$ - measured rate of flows using SLS_{i,I_n}, independently of E_m; r_j - rate of the new flow F_j; $0 < \beta_{i,I_n} \leq 1$ - service-dependent safety margin defined for the negotiated rate R_{i,I_n} of SLS_{i,I_n}.
$\check{R}^+_{i,(*,E_m)} + r_j \leq \beta^+_{i,E_m} R^+_{i,E_m}$ (2)	$\check{R}^+_{i,(*,E_m)}$ - measured rate of flows using SLS^+_{i,E_m}, considering all ingress-to-E_m estimated rates of flows; r_j - rate of the new flow F_j; $0 < \beta^+_{i,E_m} \leq 1$ - service-dependent safety margin for the rate R^+_{i,E_m} defined in SLS^+_{i,E_m}.
QoS Control Rules	Verify the conformance of QoS levels in the domain
$\forall (P_{i,p}, \beta_{i,p}) \in P_{SC_i} : \check{P}_{i,p} \leq T_{i,p}$ (3)	$\check{P}_{i,p}$ - ingress-to-egress measured QoS parameter; $\beta_{i,p}$ - corresponding safety margin; $T_{i,p}$ - parameter's upper bound or threshold, given by $T_{i,p} = \beta_{i,p} P_{i,p}$, used to set the acceptance status for Δt_i.

TABLE II
MODEL NOTATION SUMMARY

Notation	Description
Domain Notation	
D_x, D_x^-, D_x^+	Current, upstream and downstream domains
I_n	Ingress node n in domain D_x
E_m	Egress node m in domain D_x
Service Class Notation	
SC_i	Service class i supported in D_x
$P_{i,p}, \beta_{i,p}$	Target value and Safety Margin of QoS parameter p for SC_i
$T_{i,p}$	QoS threshold for $P_{i,p}$ given by $P_{i,p} * \beta_{i,p}$
SLS_{i,I_n}	Accepted upstream SLS for SC_i connecting D_x through I_n
$P_{i,I_n,p'}$	Target value of QoS parameter p' of SLS_{i,I_n}
$\phi_{i,n}$	Entry of matrix of accepted/active upstream SLSs for SC_i
SLS^+_{i,E_m}	Negotiated downstream SLS for SC_i leaving D_x through E_m
$\phi^+_{i,m}$	Entry of matrix of accepted/active downstream SLSs for SC_i
Flow Notation	
$F_j \in SLS_{i,I_n}$	Flow j belonging to an upstream SLS requiring AC
$P_{j,p''}, \gamma_{j,p''}$	Target value and tolerance to QoS parameter p'' of F_j
Monitoring Information	
Δt_i	Measurement time interval for SC_i
$\psi_{i,(n,m)}$	(I_n, E_m) entry of QoS monitoring matrix for SC_i in Δt_i
$\psi_{i,(n,m)} \rightarrow P_p$	Measured QoS parameter p of P_{SC_i} for (I_n, E_m) pair

---

Algorithm 1: Implicit AC Decision

```
(...)
F_j ⟵ NewFlowDetection()
(...)
ImplicitAC(I_n, F_j)
    (...)
    SC_i ⟵ Classify(F_j)
    SLS_{i,I_n} ⟵ IdentifySLS(SC_i, I_n, F_j)
    E_m ⟵ GetEgress(SC_i, F_j → Dst_id)
    SLS^+_{i,E_m} ⟵ IdentifySLS^+(SC_i, E_m, F_j)
    [CheckPermission(SC_i, SLS_{i,I_n}, SLS^+_{i,E_m}, F_j)]

    /* At beginning of Δt_i perform CheckQoSStatus(ψ_{i,(n,m)}) */
    ACDecision ⟵ (ψ_{i,(n,m)} → AC_Status)
    if (ACDecision = Accept) and (ψ_{i,n} → Adm_Flows > 0)
    then
        (ψ_{i,n} → Adm_Flows) ⟵ (ψ_{i,n} → Adm_Flows − 1)
        ForwardSyn()                          /* Considering
    else                                       the case of
        DiscardSyn()                           TCP flows */
    (...)
```

$CheckQoSStatus(\psi_{i,(n,m)})$
$(...)$
$\forall_{(P_{i,p},T_{i,p},\beta_{i,p})} \in P_{SC_i}$
$\quad T_{i,p} \longleftarrow \beta_{i,p}P_{i,p}$
$\quad if\ ((\psi_{i,(n,m)} \rightarrow P_p) \leq T_{i,p})\ then$
$\qquad (\psi_{i,(n,m)} \rightarrow AC\_Status) \longleftarrow AcceptCode$
$\quad else$
$\qquad (\psi_{i,(n,m)} \rightarrow AC\_Status) \longleftarrow RejectCode$
$\qquad Return()$
$(...)$
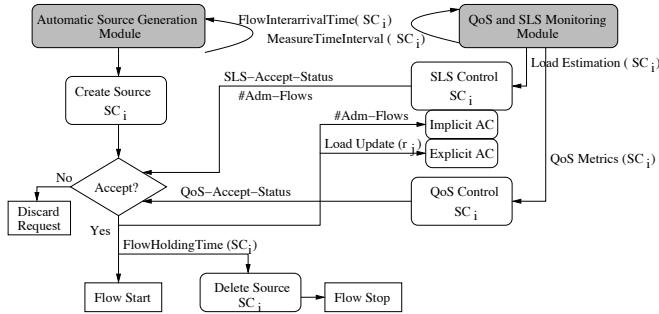
Fig. 1.   Simulation model diagram

configured as in Table III . 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.

The network domain consists of ingress routers I_1, I_2, a multiclass network core and an edge router E_1 connected through 34Mbps links. 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, which is likely to occur in real environments.

The default measurement interval Δt_i is set to $5s$. The results were obtained running a large number of simulations of about ten minutes each, after discarding an initial convergence period. Simulations up to forty minutes were also carried out in order to verify tests' consistency.

A. Proof-of-concept

An initial assessment and tuning of the AC criteria has been matter of study in [9], where has been demonstrated that generically, 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. This section is devoted to

explore and tune the behavior of implicit AC, considering both concurrent traffic and cross-traffic.

The performance evaluation includes verifying the conformance of the negotiated QoS parameters, measuring the utilization level of each class individually and of the network domain globally.

Overview of AC rules effectiveness: When examining in detail which AC rules determine the generic behavior of the model discussed in [9], 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 3); (iii) SC3 flows are mostly controlled by the QoS control rule (Eq. 3); (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 IPLR is kept within the pre-defined thresholds (see Figure 2 (b)).

Test1 - Tuning implicit AC: In the AC evaluation process, it was noticed that for implicit AC, controlling the rate variables included in Section IV brings 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, leads to long AC blocking periods and to a resource taking over effect caused by long-lived flows. Conversely, considering an $AC\_status_{\Delta t_i}$ determined by QoS control has proved to be mandatory in order to keep a "lively" number of active flows (see Figure 2 (a)), while satisfying the classes' QoS requirements (see Figure 2 (b)). In more detail:
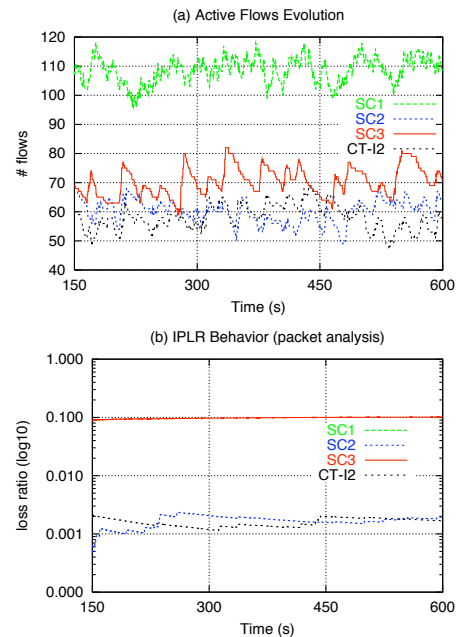

Fig. 2.   (a) active flows evolution; (b) IPLR evolution for all service classes

SC_i	Serv. Type	AC Type	$R^+_{i,Em}$	$\beta^+_{i,Em}$	$P_{i,p}$	$T_{i,p}$	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

(i) when the rate control determines the $AC\_status_{\Delta t_i}$ admittance value, this AC rule is clearly dominant, causing long rejection periods cyclically. In these periods, whose length depends on $Adm\_flows_{\Delta t_i}$, Δt_i, and on the flow interarrival and holding time distributions, long-lived TCP flows progressively take over spare resources freed by departing flows. As a consequence, the rate estimate remains high and $AC\_status_{\Delta t_i}$ is kept in rejection mode until few flows are left. When this stage is reached, the $AC\_status_{\Delta t_i}$ enters in an acceptance mode and a new cycle begins (see Figure 3);

(ii) considering $AC\_status_{\Delta t_i}$ only determined by the QoS control rule has proved to be effective in maintaining the QoS parameter IPLR bounded. However, as in (i), SC3 may exceed slightly its defined rate share, taking advantage of SC1/SC2 unused bandwidth resources (e.g., due to their traffic fluctuations or safety margins), increasing the global utilization achieved by the system without an evident QoS degradation of SC1 and SC2;

(iii) controlling $Adm\_flows_{\Delta t_i}$ may also be relevant to assure that the admitted flows receive an acceptable throughput. Controlling SC3 based on a new AC rule that compares the throughput achieved by a probing TCP stream with a reference target value may be an alternative approach for implicit AC that will be explored in future.

Test2 - Testing the impact of cross-traffic: The impact of cross-traffic on the system performance varies according to the service class considered as cross-traffic: (i) for SC3 cross traffic, the model behaves similarly to the concurrent traffic case. 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; (ii) for 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 2 (b); (iii) for SC1 cross traffic, numerous QoS violations in IPTD, ipdv and IPLR become evident and difficult to control despite the rejection indication provided by the QoS control rule. In the presence of an excessive rate at C_1, unmeasured and uncontrolled by E_1, several blocking events may occur at the scheduler (caused by the defined a Max-EF-Rate for PQ treatment). This can be minimized resorting to larger safety margins.

From this set of experiments, the relevance of the defined AC rules becomes evident for assuring service commitments in
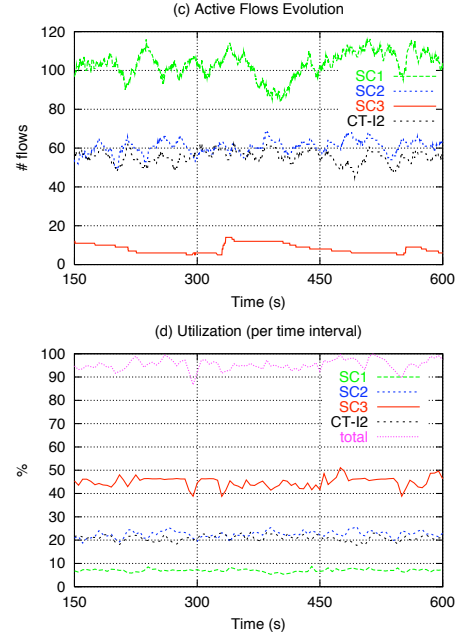

Fig. 3. SC3 behavior with SLS rate control: (a) active flows evolution; (b) achieved utilization

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.

VI. CONCLUSIONS

This paper has focused on debating and testing a flexible and easy to deploy solution to control the implicit admission of flows in multiclass environments. The devised solution, covering also the case of explicit AC, allows to accommodate a wide range of network services and to improve the corresponding QoS guarantees. In particular, the need to control elastic traffic to achieve a more efficient bandwidth utilization has been discussed, and implicit AC rules oriented to elastic and real-time services have been defined. The proof-of-concept of the proposed AC criteria, involving a multiservice environment,

evince the relevance and applicability of the defined AC rules in satisfying multiple service guarantees.

REFERENCES

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

[2] L. Massoulié and J. Roberts. Arguments in Favour of Admission Control for TCP Flows. In *16th International Teletraffic Congress*, pages 33–44, June 1999.

[3] 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.

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

[5] S. Oueslati and J. Roberts. A new direction for quality of service: flow-aware networking. In *EuroNGI'05 - 1st Conference on Next Generation Internet Networks*, pages 226–232, April 2005.

[6] S. Georgoulas, P. Trimintzios, and G. Pavlou. Joint Measurement- and Traffic Descriptor-based Admission Control at Real-Time Traffic Aggregation Points. *IEEE International Conference on Communications (ICC2004), QoS and Performance Symposium*, June 2004.

[7] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853.

[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)*, September 2004.

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