GRIP: QoS Support over a Stateless IP Domain by Means of Localized Measurements and Decisions

G. Bianchi[1], N. Blefari-Melazzi[2], M. Femminella[2], F. Pugini[3]

[1]University of Palermo, Italy. [2]University of Perugia, Italy. [3]University of Roma, "La Sapienza", Italy

ABSTRACT

This document consists of two parts. In the first one, we propose an admission control paradigm, called GRIP (Gauge&Gate Reservation with Independent Probing), devised to transparently operate over a stateless IP domain. GRIP relies the decision to admit a new flow upon the successful and timely delivery, through the domain, of probe packets independently generated by the end-points. Failed receptions of probe packets are interpreted as congestion in the network. Our solution is fully distributed and scalable, as admission control decisions are taken at the edge nodes, and no coordination between routers, which are stateless and remain oblivious to individual flows, is required. The performance of GRIP is related to the capability of routers to locally take decisions about the degree of congestion, and suitably block probe packets when congestion conditions are detected. A fundamental feature of the GRIP operation is its backward compatibility (at the expense of experienced performance) with existing routers.

This solution has been devised to operate within a specific domain, developed in a R&D project sponsored by the European Union (project SUITED). Thus, in the second part of the document, we describe such domain, where high performance can be obtained, owing to suitable assumptions on the supported traffic. However, nothing impedes to adopt at least the general principles of GRIP in other IP domains or even in the whole Internet.

Keywords: guaranteed QoS, IP domain, admission control, stateless and distributed procedures.

Corresponding Author:
Nicola BLEFARI-MELAZZI,
DIEI Department, University of Perugia
Via G. Duranti 93 - 06125 Perugia - ITALY
Tel: +39 075 585 3630
E-mail: blefari@diei.unipg.it

CONTENTS

1 INTRODUCTION

The aim of this work is to deliver QoS aware services in a specific network infrastructure, proposed in the framework of the project SUITED (Multi-Segment System For Broadband Ubiquitous Access To Internet Services And Demonstrator), sponsored by the European Union. The SUITED project proposes an integrated broadband communication infrastructure for mobile and portable IP-based services. This infrastructure consists of multiple system components (or segments) with different characteristics that are mutually complementary: i) a Ka-band regenerative satellite system, the so-called EuroSkyWay (ESW) [MST99]; ii) the GPRS (General Packet Radio Service), representing the near term version of the UMTS (Universal Mobile Telecommunication System) and the UMTS itself; iii) a wireless local area network (802.11); iv) a subset of the currently available "best effort" Internet, upgraded with QoS support features. These four segments provide a global coverage in a specific area and constitute an IP domain, managed by a single operator that intends to offer high quality services to the users that subscribe to the domain itself. In SUITED, each user terminal belonging to any of the above four segments must be able to exchange information with any other terminal belonging to the considered domain under the constraint that the users perceive pre-defined performance figures.

As a matter of principle, ad hoc QoS support solutions for our environment might be considered. However, a much more sounding rationale consist in adapting, with the smallest possible modifications, QoS architectures considered in the field of IP-based fixed network (i.e., Integrated Services and/or Differentiated Services).

Nevertheless, as recognized in the recent RFC [R2990], “both the Integrated Services architecture and the Differentiated Services architecture have some critical elements in terms of their current definition, which appear to be acting as deterrents to widespread deployment... There appears to be no single comprehensive service environment that possesses *both* service accuracy and scaling properties”. In fact:

1) the IntServ/RSVP paradigm [R2205, R2210] is devised to establish reservations at each router along a new connection path, and provide "hard" QoS guarantees. In this sense, it is far to be a novel reservation paradigm, as it inherits its basic ideas from ATM and the complexity of the traffic control scheme is comparable. In the heart of large-scale networks, the cost of RSVP soft state maintenance and of processing and signaling overhead in the routers is significant and thus there are scalability problems. In addition to the

complexity issue, it may be that the lack of a total and ultimate appreciation of this paradigm in the Internet market is also due to the fact that RSVP needs to be deployed in all the involved routers, to provide end-to-end QoS guarantees; hence this approach is not easily and smoothly compatible with existing infrastructures. What we are trying to say is that complexity and scalability are really important issues, but that backward compatibility and smooth Internet upgrade in an open, un-standardized, market scenario is probably even more important.

2) Following this line of reasoning, we argue that the success of the DiffServ framework [R2474, R2475] does not uniquely stays in the fact that it is an approach devised to overcome the scalability limits of IntServ. As in the legacy Internet, the DiffServ network is oblivious of individual flows. Each router merely implements a suite of scheduling and buffering mechanisms, in order to provide different loss/delay performance aggregate service assurances to different traffic classes whose packets are accordingly marked with a different value of the DS code-point field in the IP packet header. By leaving untouched the basic Internet principles, DiffServ provides supplementary tools to further move the problem of Internet traffic control up to the definition of suitable pricing/service level agreements (SLAs) between peers. However, DiffServ lacks a standardized admission control scheme, and does not intrinsically solve the problem of controlling congestion in the Internet. Upon overload in a given service class, all flows in that class suffer a potentially harsh degradation of service. The RFC [R2998] recognizes this problem and points out that "further refinement of the QoS architecture is required to integrate DiffServ network services into an end-to-end service delivery model with the associated task of resource reservation". It is thus suggested [R2990] to define an "admission control function which can determine whether to admit a service differentiated flow along the nominated network path".

Our aim is to define such an admission control function, in a DiffServ framework, at least in our domain. Our solution, named GRIP (Gauge&Gate Reservation with Independent Probing), relies the decision to admit a new flow upon the successful and timely delivery, through the domain, of probe packets independently generated by the end-points, upon flow setup. End points interpret failed receptions of probes as congestion in the network and reject the relevant admission requests. This idea is close to what TCP congestion control technique does, but

it is used in the novel context of admission control.

The organization of this paper is the following. Section 2 describes the distributed components of our mechanism. In the same Section, we discuss also how GRIP is compatible and applicable to the Legacy and DiffServ Internet and we provide preliminary performance results. Section 3 presents a GRIP implementation in the SUITED domain, where all routers and traffic sources adhere to a set of hypotheses, and shows that GRIP may ultimately lead to as much as "hard" QoS guarantees. Conclusive remarks are given in section 4.

Finally, we stress that GRIP has been implemented in a test-bed developed in the framework of the SUITED project and is currently running fine and in agreement with our theoretical findings.

2 GRIP: GAUGE&GATE REALISTIC INTERNET PROTOCOL

GRIP is a fully distributed and scalable Admission Control scheme intended to operate over an enhanced DiffServ domain, but, in principle, compatible with the legacy Internet. GRIP is a constructive answer to a criticism often moved to DiffServ, i.e., that, apparently, DiffServ cannot provide end-to-end guarantees to traffic flows, since it does not provide support (e.g., signaling exchange between routers, and per flow states in routers) to admission control procedures. GRIP builds upon the idea that admission control can be managed by pure end-to-end operation, involving only the new flow ingress router (or source host) and egress router (or destination host). In this, GRIP is related to the family of distributed schemes [BOR99, ELE00, BCP00, BIA00, BRE00, GKE99] recently proposed in the literature under the denomination [following BRE00] Endpoint Admission Control (EAC). The driving idea of EAC is that, upon connection set-up, each sender-receiver pair starts a Probing phase whose goal is to determine whether the considered connection can be admitted to the network. In most EAC proposals, during the Probing phase, the source node sends packets that reproduce the characteristics (or a subset of them) of the traffic that the source wants to emit through the network. Upon reception of the first probing packet, the destination host starts monitoring probing packets statistics for a given period of time. At the end of the measurement period and on the basis of suitable criteria, the receiver takes the decision whether to admit or reject the connection and notifies back this decision to the source node.

However, the described mechanism has performance drawbacks mostly related to the measurement time performed at the destination. In fact, either it lasts for a significant amount of time to provide an accurate

estimate of the network status (but the set-up time can increase excessively), or it provides unreliable estimates of the network load. In addition, the mechanism is driven by necessarily imprecise (and variable) network measurements.

To overcome such limits, GRIP inherits the idea of combining endpoint admission control with measurement based admission control, which was first proposed in [ALM98], where the SRP (Scalable Reservation Protocol) was outlined. Since, at that time, EAC ideas had not yet been published, the authors presented their proposal as a possible solution to the scalability problems of RSVP. Unfortunately (see e.g., what stated in [BRE00]), SRP appears much more like a lightweight signaling protocol, with explicit reservation messages, rather than an EAC technique with increased intelligence within the end routers. In fact, MBAC techniques (see e.g., [GRO99, BJS00] and references therein contained) have been proposed as a way to avoid scalability problems. In MBAC, each router measures the aggregate traffic that it is handling. Admission Control decisions are then taken by the routers on such measurements (and, optionally, on a set of parameters carried in the connection request), rather than on rules based on analytical calculation of loss/delay bounds (and on specific traffic models). This procedure does not require maintaining state information but does require to exchange signaling information needed to request and accept the connection and eventually to co-ordinate the CAC mechanism performed by the involved routers. In GRIP, we combine some key ideas of SRP and of MBAC techniques, but in the light of the new paradigm of EAC.

We envision GRIP as a mechanism composed of the following three components: (i) GRIP source node protocol, (ii) GRIP destination node protocol, (iii) GRIP Internal Router Decision Criterion. The source and destination node protocols can be considered running at the ingress and egress nodes (for obvious security reasons) of possibly different Service Providers. However, from a logical point of view, the source and destination node protocols are more naturally envisioned as running on the user's terminals.

2.1 GRIP End nodes operation

GRIP's end nodes operation is extremely simple. Fig. 1 illustrates the setup of an "uplink" (source to destination) monodirectional flow. When a user terminal requests a connection with a destination terminal, the Source Node starts a Probing Phase, by injecting in the network in principle just one Probe Packet. Meanwhile, it

activates a probing phase timeout, lasting for a reasonably low time. If no response is received from the destination node before the timeout expiration, the source node enforces rejection of the connection setup attempt. Otherwise, if a Feedback packet is received in time, the connection is accepted, the probing phase is terminated, and control is given back to the user application, which starts a Data Phase, simply consisting in the transmission of information packets. The role of the Destination Node simply consists in monitoring the incoming IP packets, intercepting the ones labeled as Probes, reading their source address, and, for each incoming probe packet, just relaying with the transmission of a feedback packet, if the destination is willing to accept the set-up request.

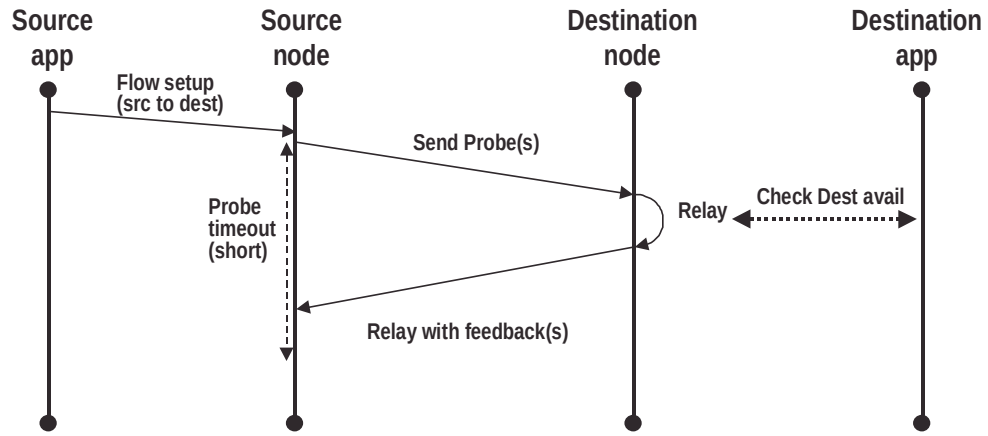


Fig. 1 - End point GRIP operation

When GRIP is envisioned over a DiffServ architecture, the only mandatory requirement is that Probes and Information packets are labeled with different values of the DS codepoint field in the IP packet header. This enables DiffServ routers to provide different forwarding methods to Probes and Information packets, e.g., granting service priority to Information packets. In this case, the Feedback packet shall be labeled as an Information packet (i.e., prioritary).

Note that the described GRIP operation is trivially extended to provide setup for bidirectional connections. In such a case, the destination node will simply relay with a Probe packet instead than with a Feedback packet. A Feedback will be ultimately sent back by the source node upon reception of the destination Probe (to close the three way connection setup handshake – independent probing mechanisms are clearly needed to test both uplink and downlink network paths, which generally differ). GRIP can be adapted to support "downlink" (destination to source) flows. The source node needs to issue a Trigger Packet to drive (by mean of application-level protocol

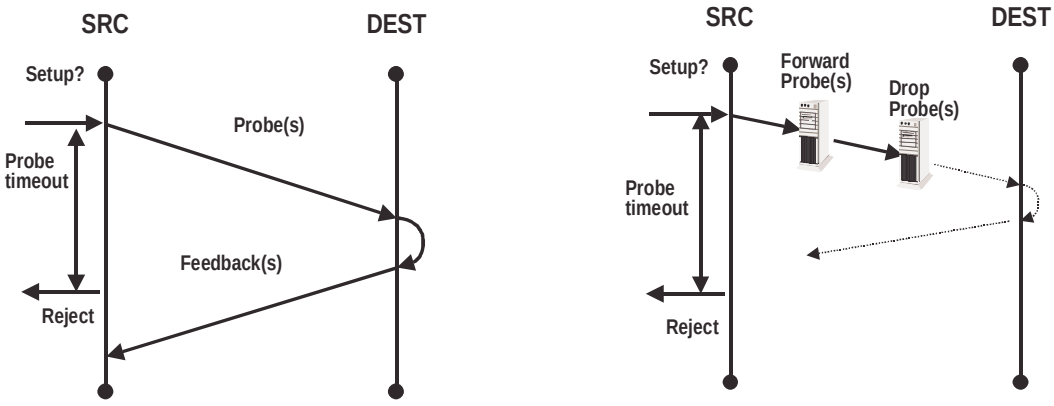
information, contained in the Trigger Packet payload) the destination node to start a Probing Phase on its own. Finally, GRIP leaves the service provider free to provide optional implementation details, including:

- Addition of proprietary signaling information in the probing packet payload or in the feedback packet payload, to be parsed, respectively, at the destination node or at the source node.
- Definition of more complex probing phase operation, e.g., by including reattempt procedures after a setup failure, multiple timers and probes during the probing phase, etc.
- Definition of more complex node protocol operation, such as multiple feedback packets, taking decisions (as in some EAC schemes) at the destination node on the basis of measurements of the probing packet flow (i.e., not just relaying), etc.

The given description appears to include all the EAC schemes as particular cases. Besides the packet tagging, the simplest implementation, i.e., a single probe packet and a single feedback packet, is compatible with the H.323 call setup scheme using UDP, which encapsulates a H.225.0v2 call setup PDU into a UDP packet. The GRIP operation is then perfectly compatible with existing applications.

2.2 GRIP over a GRIP-unaware domain

The rationale of GRIP is to reject a new flow setup when a feedback does not return to the source node before the probing timeout expires. The case of flow rejection, when GRIP is operated over a GRIP-unaware domain, i.e., Legacy or DiffServ routers, is illustrated in Fig. 2-a.



(a) GRIP over legacy/diffServ Internet: reject driven by network congestion

(b) GRIP over GRIP-aware Internet: reject driven by internal router decisions

Fig. 2 - GRIP operation over different environments

In this case, flow rejection is purely driven by internal network congestion. Upon congestion, the round trip delay (Probe plus Feedback) may become larger than the probing phase timeout, and thus a flow setup is rejected. Stability is guaranteed by the fact that, when network congestion increases, a corresponding decrease in the probability that setup is successful occurs. Therefore, a lower number of new flows set up, and this allows the network to smoothly decongest.

Routers may be in principle oblivious of Probes, and may treat them as normal IP packets. When packet differentiation is possible, as in the DiffServ scenario, GRIP operation can be enhanced. This particularly occurs when DiffServ routers are configured to distinguish Information packets from Probes on the basis of their DSCP value, and serve information packets with higher service priority (i.e., before) than probing packets. This operation has the advantage that the delay experienced by Probing packets is necessarily worse than that experienced by packets belonging to accepted connections. Thus, probes may detect internal router congestion earlier than data packets, and earlier drive reject decisions at the end points.

Although a full performance evaluation of GRIP over a DiffServ framework is out of the scope of the present paper, it is indeed instructive to discuss the sample simulation run presented in Fig. 3.

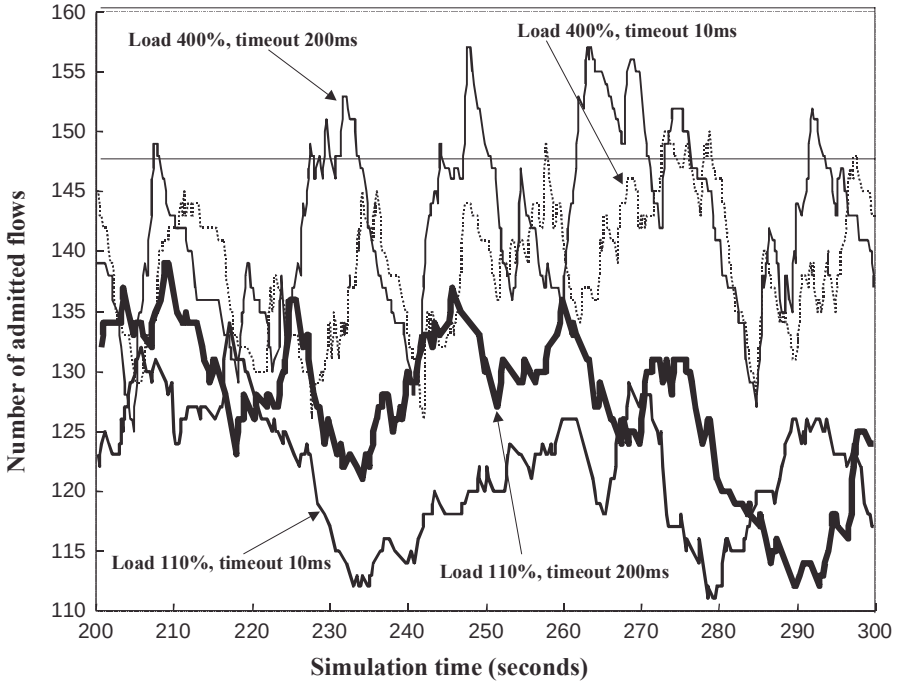


Fig. 3 – Simulation run for GRIP over DiffServ

Voice calls, of average duration equal to 1 minute and exponential distribution, are offered to a single link of capacity 2 Mbit/s. The only delay accounted for, in the simulation, is the queuing delay within the router. Voice sources emit according to an exponential ON/OFF pattern, i.e., they alternate exponentially distributed ON periods of average value 1 second, with exponentially distributed OFF periods of average 1.35 seconds. During the ON period, the source emits fixed-sized packets of 1000 bits at a "peak" emission rate of 32 Kbit/s. Trivial computation yields to an average source rate equal to 32*0.4255≅13.6 Kbit/s. Fig. 3 reports the number of admitted voice flows versus the simulation time, for two different load conditions: 110% offered load (i.e., 2.7 calls/s) and 400% offered load (i.e., 9.8 calls/s). Two source probe timeouts (i.e., the timeouts reported in Fig. 1 and Fig. 2) have been adopted: an extremely short and unrealistic 10 ms timeout, and a more reasonable 200 ms timeout. A first consideration is the capability of GRIP to control link overload. With reference to the 400% load case, Fig. 3 shows that the number of admitted connections can overflow the limit value 146 (i.e., 100% link utilization). However, as the accepted traffic gets greater than the link capacity, data packets queue builds up, and, thanks to the forwarding discipline, the probing buffer server remains blocked until congestion disappears (the "remedy" period, following [GRO99] where a similar behavior is shown to be a characteristic also of centralized MBAC schemes). This proves that, at least, GRIP introduces in plain DiffServ networks an effective and stable form of traffic control, which impedes persistent link congestion. While, in the general case, we are far from satisfactory throughput/delay performance provisioning, Tab. 1 shows that, in the case of light overload such as 110%, better than best effort performance can be achieved.

Load	Timeout	Throughput	95-th delay perc.	99-th delay perc.
110%	10 ms	0.846 ± 0.002	8.7 ms ± 1.9	51.8 ms ± 12.1
110%	200 ms	0.902 ± 0.006	61.3 ms ± 7.9	141.3 ms ± 10.4
400%	10 ms	0.9626 ± 0.001	229 ms ± 22	402 ms ± 46
400%	200 ms	0.9806 ± 0.0004	368 ms ± 27	609 ms ± 71

Tab. 1 - Performance results (with 95% confidence intervals) for the four cases considered in Fig. 3

A second important consideration that can be drawn from Fig. 3 and Tab. 1 is the role of the source timeout. Intuitively, performance calibration seems possible via source timeout tuning, since a short timeout is expected to increase the probability that the setup is aborted before the reception of the feedback packet. Conversely,

results show that very strict timeout settings have limited effect on the system performance. Our 10 ms timeout choice was indeed motivated by the attempt to understand whether GRIP over DiffServ could be ultimately tuned to provide toll-quality delay performance (say few ms 99-th delay percentiles, in turns achieved, with the above source and link parameters, by limiting the link utilization to about 75% [BIA00], i.e., 110 admitted flows). Unfortunately, in high load conditions, the link utilization gets close to 100% (Tab. 1) despite the 10 ms timeout. Although limited, the spare link capacity left by the forwarding discipline to the probing packets appears sufficient to allow occasional periods of very limited queuing delay for probing packets. Hence, probes can thus frequently reach the destination in less than 10 ms and drive flow admissions. In conclusion, we can say that GRIP over DiffServ appears to provide a sort of IntServ Controlled Load QoS support. Finally we note that, paradoxically, the relative insensibility of GRIP with respect to the source time-out can be seen as an advantage: the GRIP operation over a GRIP-aware domain (see Section 2.3) does not strongly depend on accurate estimates of the round trip time, thus making our proposal robust with respect to this parameter.

2.3 GRIP over a GRIP-aware domain

The major conclusion that can be drawn from the investigation in Section 2.2, is that GRIP over DiffServ is only a preliminary step toward guaranteed QoS support. Ultimately, it appears necessary to upgrade routers with effective decision criterions able to explicitly enforce blocking of probes, when the accepted load is in a critical range. In fact, despite the above discussed performance drawbacks, our strongest argument in favor of GRIP is that it opens a smooth migration path toward a future QoS capable global infrastructure. Our thesis is that GRIP widespread deployment may start over the actual best-effort Internet to provide marginal performance improvements, with the promise that QoS will be provided in the future by independent router upgrades in independent IP domains, such as the one of the SUITED project.

To justify our statement, refer to the scenario depicted in Fig. 2-b. Here, network routers are assumed to be able to recognize that packets labeled as Probes carry out an admission control function. Hence, they may intelligently enforce Probe dropping, on the basis of suitable estimations of the QoS provided to the already admitted flows, and on the basis of suitable predictions of emerging congestion conditions. Since internal probe losses drive setup rejections at the distributed end points, independent, localized and proprietary (i.e., not

adhering to a standard) decisions taken at the network routers may substantially improve the QoS provided within a domain.

The GRIP-aware router operation is illustrated in Fig. 4. For convenience of presentation, we assume that the router handles only GRIP controlled traffic. Other traffic classes (e.g., best-effort traffic) can be handled by means of additional queues, eventually with lower priority. At each router output port, GRIP implements two distinct queues, one for data packets, i.e., belonging to flows that have already passed an admission control test, and one for probing traffic. Packets are dispatched to the respective buffers according to the probe/data DSCP tag. The GRIP router measures the *aggregate* accepted traffic. On the basis of the running traffic measurements, the router enforces a Decision Criterion, which continuously drives the router to switch between two states: ACCEPT and REJECT. When in the ACCEPT state, the Probing queue accommodates Probe packets, and serves them according to the described priority mechanism. Conversely, when the router switches to the REJECT state, it discards all the Probing packets contained in the Probing queue, and blocks all new Probing packets arriving.

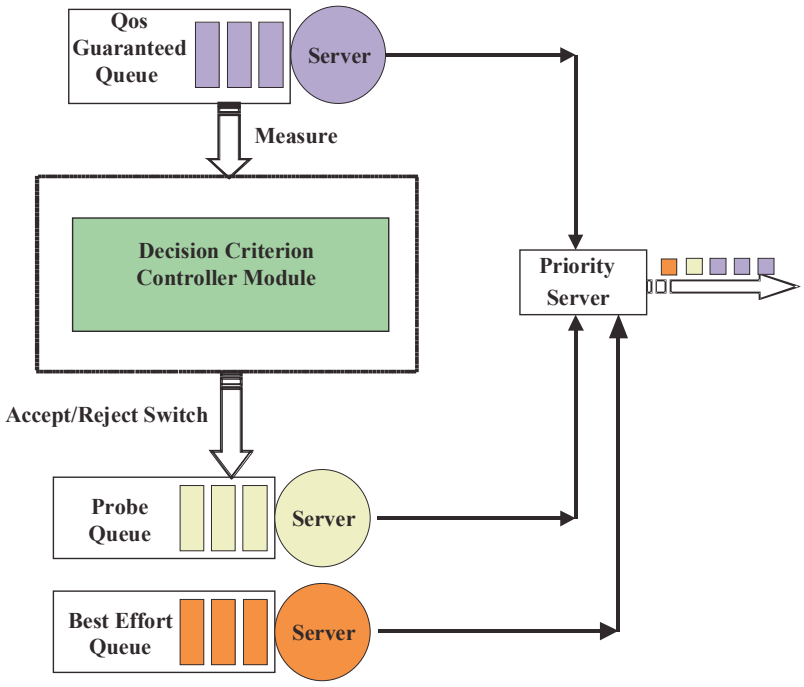


Fig. 4 – GRIP router operation

In other words, the router acts as a gate for the probing flow, where the gate is opened or closed on the basis of the traffic estimates (hence the Gauge&Gate in the acronym GRIP). The Decision Criterion may also be based

on different and simpler means than traffic measurements, e.g., limiting accepted probe packets via probe buffer limitations, or other, tunable proprietary schemes. This mechanism provides an implicit signaling pipe to the end points of which the network remains unaware. Each router is locally in charge of deciding whether it can admit new flows, or it is congested. The notion of internal router congestion is not standardized, and it is up to each specific router DC implementation to determine if, and when, congestion arises. The internal (arbitrarily sophisticated and performing) router decision is summarized in the router state (ACCEPT vs. REJECT), and it is implicitly advertised to the end points (whose flow setup path crosses the considered router) by letting Probes cross through the router (ACCEPT) or blocking probes (REJECT). When the router is in the ACCEPT state, it advertises that it can admit new connections. When the router is in the REJECT state, no probing packets are forwarded. Since the distributed admission control decision is related to the successful reception at the destination of the Probing packets, locally blocking probing packets implies aborting all concurrent setup attempts of connections whose path crosses the considered router. Conversely, a connection is successfully setup when all the routers crossed by a probing packet are found in the ACCEPT state.

As regards performance, it is easy to conclude that the level of QoS support provided depends on the degree of effectiveness of the Decision Criterion implementation. Several Measurement-Based mechanisms [BJS00] have been described in the literature and may be applied to the GRIP router. In Section 3, we apply GRIP in a full-fledged QoS domain, with GRIP capable routers and suitable assumptions on the offered traffic. In this scenario, we define a robust Decision Criterion and we verify that GRIP can provide *hard* end-to-end QoS guarantees. In other words, the performance perceived by the users is *always* the requested one.

2.4 GRIP rationale

We remark that no explicit agreement among routers is necessary to run GRIP. This driving principle is the way to provide a smooth migration path consisting in distributed admission control schemes of increasing complexity and effectiveness, which can indeed operate over a multi-provider and multi-vendor Internet. The performance gradually improves as GRIP comes into operation and "hard" guarantees are achieved when all routers apply admission control decisions. In GRIP, multiple decision criterions in the routers (e.g., routers of different vendors, "injected" in the market at different times, and in different domains) along a connection path

do not affect the principles of the GRIP operation, but only its performance. In other words, ours is a market scenario vision of an incremental and backward compatible solution. This must be accounted when comparing GRIP with other schemes (such as MPLS), which require coordination and "bulk upgrade" of the routers. Thus these schemes appear less flexible to respond to the competitive and demanding Internet market scenario. In essence, GRIP guarantees respect for the following principles:

- Backward compatibility: as shown in Section 2.2, GRIP may be operated over Legacy or "standard" DiffServ routers.
- Smooth migration path: GRIP implementation may start over the actual best-effort Internet to provide marginal performance improvements. A future QoS capable global infrastructure can be provided by independent upgrades.
- Scalability: no state information is stored in the routers; the latter handle traffic *aggregates* and not single flows. GRIP does not require any specific protocol implementation in the routers, which are stateless and remain oblivious to individual flows.
- Distributed operation: procedures have a local scope, and each network entity does not have to explicitly co-operate with other entities so that: i) all the network devices operate autonomously, facilitating multi-vendor markets, ii) the exchange of signaling messages is implicit, iii) the inter-working among different sub-networks/operators is greatly simplified.
- Performance calibration: QoS is not granted by the GRIP operation, but it results as a tradeoff between performance and degree of complexity of the GRIP components. Moreover, there will exist a suitable set of "tuning knobs" (i.e., parameters) in each GRIP component, which allows independent network operators to vary the level of achieved performance and utilization within their domains, and thus set target performance levels (as required in [BJS00]).

It is easy to see that, while the above characteristics of GRIP are desirable in any case, they are essential in our heterogeneous framework. In the SUITED domain, explicit signaling exchanges between the “different” segments (satellite, UMTS, 802.11, the Internet) and the interworking functions would be a sheer nightmare.

3 GRIP IN THE SUITED PROJECT SCENARIO

In SUITED, QoS support is envisioned in two phases. In a first phase of the project, we focus on a specific traffic scenario composed of two classes: homogeneous IP Telephony sources with QoS guarantees and best effort traffic. A second phase of the project will focus on the extension to heterogeneous QoS aware sources, i.e., characterized by different emission patterns and performance levels. The aim of this Section is to prove that GRIP can provide as much as hard QoS guarantees within a specific domain. Two key points allow QoS guarantees: traffic control assumptions at the domain edge, and suitable definition of a Decision Criterion to accept or reject probing packets in the routers, based on *aggregate* traffic measurements.

3.1 Edge traffic control

In SUITED, traffic sources are regulated at the edge of the network by standard Dual Leaky Buckets (DLB). This choice stems from the fact that, in the past years, countless CAC rules have been proposed in the literature and different sets of Traffic Descriptors (TDs) have been assumed, but a fully satisfying solution was not found. We believe that one of the reasons why this happened was *the lack of a simple and standardized source traffic model*. In fact, the existence of many different type of sources (e.g., voice, MPEG, FTP traffic, WEB traffic, signaling traffic, etc.) implies the definition of source models and policing algorithms for each of them. This means that in principle each source has its own set of TDs and policing algorithms. Therefore, the task of defining a single, feasible and simple CAC for all of them is very complex. To overcome this problem, a standardized traffic regulator, the so-called Dual Leaky Bucket (DLB), has been adopted.

We point out that the assumptions of DLB regulated traffic sources is adopted in all the approaches discussed in [BJS00] *and* in the IntServ framework. In any case, we make no assumptions on the average behavior of flows, beyond the worst case parameters supplied by the DLB characterization.

The DLB regulates the traffic emitted by a source before it enters the network. The regulated traffic is characterized by four parameters, independently of the source. These parameters are: the Peak Rate and its Tolerance, the Sustainable Rate and its Tolerance. The Sustainable Rate is an upper bound of the average rate of a connection. For simplicity, we assume, as in [ELM97], that the Tolerance of the Peak Rate is equal to zero or included in the Peak Rate parameter. The tolerance of the Sustainable Rate can be expressed by means of the parameter Maximum Burst Size (MBS). The MBS should be used to set an upper bound to the burst length at

peak rate. Instead of the parameter MBS, it is commonly used a parameter called Token Bucket Size. Thus, each DLB regulated source is described by means of the following three Traffic Descriptors:

- P_S: the Peak Rate (in bytes/s);
- r_S: the Sustainable Rate (in bytes/s);
- B_{TS}: the Token Bucket Size (in bytes)

The Token Bucket Size is univocally related to MBS by the expression $B_{TS} = (MBS - 1)(P_S - r_S) / P_S$. The DLB parameters can be defined once and for all for a given traffic class (e.g., IP telephony), or chosen by the user by trading off performance with resource requests and thus cost.

In addition, we specifically require the DLB to enforce that traffic does not underflow the sustainable rate specification. This is accomplished by the emission of "dummy" packets, in order not to waste token. Note that this assumption is not an unrealistic one: if a user requests a QoS service and pays for it on the basis of the selected DLB parameters, it is likely that emission opportunities will not be wasted (greedy sources). The consequence is that the number of bytes, *b(T)*, emitted by a source during an arbitrary time window of size *T* (seconds) is upper and lower bounded by:

$$\max(r_S T - B_{TS}, 0) \le b(T) \le r_S T + B_{TS} \quad (1)$$

Finally, we assume that the sources are divided in traffic classes, each comprising independent and homogeneous sources (i.e., with the same DLB parameters). For simplicity, in this document, we focus on a homogeneous traffic scenario. Note that a possible way to handle also heterogeneous sources is the following. DiffServ envisions different "traffic classes", each with specific requirements. In the view of a small number of traffic classes as premium and QoS aware services (e.g., a class could be IP telephony), homogeneous flows with specific requirements make up a class. Each class can be handled in a differentiated way, with its own pair of DS codepoints for probing and data, by means of suitable scheduling mechanisms, similar to those already defined (e.g., WFQ, separate queues). Other ways to handle heterogeneous sources are discussed in Appendix 2.

3.2 Decision Criterion

The key of GRIP is the definition of a Decision Criterion (DC) able to provide performance guarantees. The SUITED traffic scenario discussed above allows us a number of simplifying assumptions, namely: (i) each

traffic source emission is regulated by a DLB; (ii) the sources are homogeneous, that is they are characterized by the same parameters of the DLB regulators; (iii) the sources are greedy. These restrictions have their cons, but allow defining a DC able to provide hard QoS guarantees, i.e., to really achieve the performance promised to the users.

The localized DC running on each router's output link is based on the runtime estimation of the number of the active sources and on the off-line computation of the maximum number, K, of sources that can be accepted without exceeding target performance (e.g., loss / delay) levels. For DLB regulated sources, a simple and elegant solution for such an off-line computation has been proposed in [ELM97]. For our scopes, we consider K as a "tuning knob", which allow the domain operator to set target performance levels [BJS00]. The operator chooses target performance levels; the latter are mapped in a value of K and GRIP enforces such value. Thus, GRIP is completely independent by the algorithm chosen to evaluate K. For instance, if we consider a link of capacity C bytes/seconds, and a FIFO buffer of size B bytes and if the target performance is no packet loss, than in [ELM97] it is shown that the maximum number K of flows that can be admitted under such constraint is:

$$K = \frac{CB_{TS} + B(P_S - r_S)}{B_{TS} P_S} \qquad (2)$$

where P_S, r_S, B_{TS} are the DLB parameters of the controlled sources. The results provided in [ELM97] allow evaluating values of K such as the packet loss probability or delay quantiles are kept under suitable and pre-defined thresholds. In general, it is easy to see that whatever performance figure can be enforced by a suitable choice of K.

In state-based admission control schemes, it would be sufficient to keep track of the number of admitted connections N, and accept a new connection setup request as long as $N \leq K$. This would imply a suitable signaling protocol. On the contrary, we estimate runtime the number of allocated connections (or flows, in IntServ and DiffServ jargon) thus avoiding signaling and state maintenance.

3.3 Estimation of the number of admitted sources

GRIP estimates the average amount of traffic offered at each router's output link by means of a sliding window. During a window of size T, each router counts the number of bytes passing through the considered link. To define the length of the measurement window T, we use the period of the worst case DLB output [ELM97],

characterized by an activity (On) period with emission at the Peak rate and a silent (Off) period, both with deterministic length. The length of this period is:

$$T_{\min} = T_{ON} + T_{OFF} = \frac{B_{TS}}{P_S - r_S}\frac{P_S}{r_S} \tag{3}$$

The window size is then T, with $T \geq T_{min}$, in order to catch at least the minimum periodicity of the source, associated to the worst case DLB output.

Assume now that a *fixed* number N of flows is allocated on the link, i.e., no flow arrivals and departures occur. Since the traffic emitted by each source is regulated by a DLB and since the sources are greedy, it is easy to see that the number of bytes, $A(T)$, measured in a window of size T is bounded by (see Eq. 1):

$$A_{MIN}^{N} = N(r_S T - B_{TS}) \leq A(T) \leq A_{MAX}^{N} = N(r_S T + B_{TS}) \tag{4}$$

With the above assumptions, it exists a minimum window size, such that the number of flows, N, is evaluated exactly. Such minimum window size, denoted in the following as exact-window, T_{ex}, can be determined by imposing the condition:

$$A_{MAX}^{N-1} < A_{MIN}^{N} \tag{5}$$

which yields:

$$T_{ex} > \frac{(2N-1)B_{TS}}{r_S} \tag{6}$$

This formula implies that the exact-window can last a significant amount of time. We stress that our measurement procedure operates in background and does not influence the flow set-up time, as in some EAC schemes. However, even if we are not constrained to adopt a very short measurement time, we want to avoid using always an exact-window, since its size can reach values in the order of several minutes, depending on the DLB parameters. This is because a too large window has some cons that will be discussed in the sequel. The solution is to trade off the window size with the accuracy in the estimation of N.

In general, given a window T and a number $A(T)$ of bytes measured within the window, according to Eq. 4, the number of flows is a random variable in the range:

$$N_{MIN} = \left\lceil \frac{A(T)}{r_S T + B_{TS}} \right\rceil \le N \le N_{MAX} = \left\lfloor \frac{A(T)}{r_S T - B_{TS}} \right\rfloor \quad (7)$$

If no conjecture is made on the statistical properties of the emission process for each source (which depends on how the traffic source fills the DLB, and we do not rely on any hypotheses on source behaviors), the distribution of N between these two extremes remains unknown. To provide a conservative estimate, the admission control scheme estimates the number of allocated flows as $N_{est}=N_{MAX}$, and a new flow is accepted if $N_{est} \le K$.

In conclusion, the router is in the ACCEPT state and probing packets are allowed to pass the gate, as long as

$$N_{est}=\left\lfloor \frac{A(T)}{r_S T - B_{TS}} \right\rfloor \le K \quad (8)$$

Clearly, the longer T, the narrower the range in Eq. 7, and the higher the link utilization.

Eq. (7) shows that, as the window size increases, the bounds N_{MAX} and N_{MIN} become closer and when the exact-window is reached, N_{est} is evaluated exactly. To evaluate the effectiveness of the estimation procedures let us define the parameter $N_{est,min}$ as the minimum value of N that satisfies the condition:

$$A_{MAX}^{N} \ge A_{MIN}^{K} \quad (9)$$

The parameter $N_{est,min}$ is then the minimum number of flows that can emit a number of bytes greater or equal than that emitted by K flows. Solving inequality (9) versus N yields:

$$N_{est,\min} = \left\lceil K \frac{r_S T - B_{TS}}{r_S T + B_{TS}} \right\rceil \quad (10)$$

Hence, the parameter $N_{est,min}$, represents the worst case estimation, obtained when all the sources emit at their maximum, but the estimation rule, in order to be conservative, is forced to consider them as emitting at their minimum. Thus, such parameter is the minimum number of accepted flows that may drive the admission control rule to switch to a REJECT state.

In Fig. 5 we show $N_{est,min}$ (normalized with respect to K) as a function of the window size T. The horizontal line equal to one is the ideal condition in which $N_{est,min}=K$. Fig. 5 shows that the range of ambiguity (i.e., the range K-$N_{est,min}$), where the conservative admission control rule may reject a connection even if it could be

accepted, reduces and vanishes as T increases.

The figure reports results for two different link capacities: C=2 Mbps and C=5 Mbps; the link buffer size is B=53000 bytes. The values of the DLB parameters are: P_S = 4 Kbytes/s; r_S = 1.7 Kbytes/s; B_{TS} = 5300 bytes. We note that the two curves are almost identical. This example hints that the measurement procedure is robust with respect to the link capacity. In this example, the length of the exact-window is equal to 618 s for C=2 Mbps and 1767 s for C=5 Mbps, thus confirming that the exact-window can be very large. The above analysis has been concerned with the stationary (or steady state) case, when the number of admitted flows is constant and no flow arrivals and departures occur. It is clear that the transient behavior resulting from arrivals and departures must be suitably taken into account.

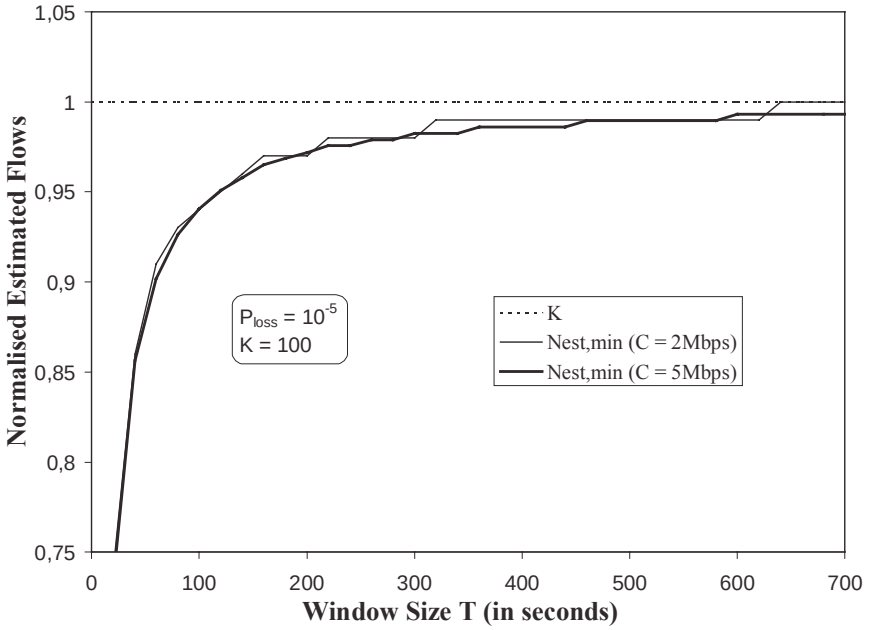


Fig. 5 - Number of estimated flows, as a function of the window size

A key aspect is the dimensioning of the measurement window, which is of fundamental importance in the theory of MBAC (a full treatment has been made in [GRO99]). In general, in MBAC systems, an optimum window dimensioning is the compromise of two different factors:

- the need of a long measurement window, to overcome the incapability of distinguishing different states of the system; a large window permits a good separation between the packets emission intervals associated to different number of active flows;

- the need of a short measurement window in order to react quickly to the variation of the number of active flows due to the flows arrivals and departures.

3.4 Transient Management and Stack Protection

The evaluation of the number of acceptable flows, N_{est}, is executed by counting Data packets. However, when a router "accepts" the Probe packet of a given flow (i.e., the probe finds the gate opened), the relevant Data packets are not yet emitted by the source. In other words, it exists a transient time, during which the router is loaded with a new flow, but it cannot accurately account for the relevant traffic. This implies that, when a large number of new flows activate in a very short time frame, overallocation above the maximum value K may occur, and thus QoS is not guaranteed in all operational conditions.

In “normal” conditions, the conservative admission control rule provides enough margins to avoid that transient effects result in over-allocation of flows. Nevertheless, to provide strict guarantees in any operational condition, we account for transient effects by introducing a stack protection scheme.

In the GRIP operation considered in SUITED, each probing packet accepted (i.e., each probing packet that finds the gate opened), may generate a new stream of active packets. (i.e., it will, unless further routers along the path reject the same probing packet). We have overcome the problem of new flow activation by using a "stack" variable, which keeps memory of the amount of "transient" flows. Whenever a probe packet is accepted, the stack is incremented by one. A timer equal to the duration of the measurement window T is then started, and the stack is linearly decremented at a rate $1/T$ until the timer expires.

The rationale of this technique is simply explained by considering the case of a single flow admitted at time t_1. Neglecting the Round Trip Delay, at time t, with $t\text{-}t_1<T$, the admitted flow has contributed to the measurement during the time interval (t_1,t), while it has been not accounted (i.e., measured idle) during the time interval $(t\text{-}T,t_1)$. Indeed, our linear stack, at time t, is equal to:

$$1-\frac{t-t_1}{T} \qquad (11)$$

and thus it compensates the lack of packets emitted by the source in the time interval $(t\text{-}T,t_1)$. This mechanism changes the router gate condition (Eq. 8) as follows: the router is in the ACCEPT state, as long as:

$$N_{est+STACK} = \left\lfloor \frac{A(T)}{r_S T - B_{TS}} + STACK \right\rfloor \leq K \qquad (12)$$

The described operation is clearly optimal if each accepted probing packet results in a new accepted connection. In fact, the stack variable will take into suitable account the presence of all the flows that became active in the last T seconds. As a side note, we remark that the stack is a simple variable and hence it does not require to process probing packets and extract or maintain state information.

The major drawback of this mechanism is that subsequent routers along the path may discard some of the probing packets. In this case, the stack will provide transient reservation of system resources for a non-existent flow, thus resulting in lower link utilization.

3.5 Numerical results

The effectiveness of GRIP has been evaluated by means of simulation results. As discussed above, we make no assumptions on the behavior of the traffic sources, beyond the worst case parameters supplied by the DLB characterization; however, to generate traffic, we have to load the DLBs with specific sources. We have obtained results for two cases: constant rate sources (labeled as CBR) emitting at rate 1.7 Kbytes/s, and on-off exponential voice sources (labeled as EXPO). As regards the EXPO sources, in the On state (talkspurt) the source emits cells periodically. The time spent in the On and in the Off state is exponentially distributed, with average values of 352 ms and 650 ms respectively: The bit rate during the On period is equal to 4 Kbytes/s. Both sources are regulated by DLBs with parameters: P_S = 4 Kbytes/s; r_S = 1.7 Kbytes/s; B_{TS} = 5300 bytes. We consider a generic router's output link with parameters: link rate: C = 2.048 Mbps; buffer size: B = 53000 bytes. We set, as target performance figure, a packet loss probability, P_{loss}, equal to 10^{-5}. According to the acceptance rule provided in [ELM97], the corresponding maximum number of acceptable flows is K=100. Finally, unless otherwise specified, the call arrival rate, modeled as a Poisson arrival process, has been set to 1 call/s, and each call duration has been drawn from an exponential distribution with mean value 4 minutes. This implies that 240 Erlangs are offered to the link, i.e., more than twice the maximum number of calls that can be, in principle, simultaneously admitted (K=100).

We stress that the choice of such parameters and performance figures (limited to packet loss, but in principle extensible to whatever other figures) has here only a significance of a case study.

A first question is: how effective is the estimation rule for the number of admitted sources?. Fig. 6 reports the number of actually admitted sources, N_{adm}, versus the simulation time, for a window length T=30s, and for the EXPO traffic model (and with the stack mechanism active). Due to our conservative admission control decision rule, the maximum number of admitted sources during the simulation run is guaranteed to remain below the value K=100, representing the maximum number of flows that can be admitted. Also shown is the mean value of N_{adm}, M.

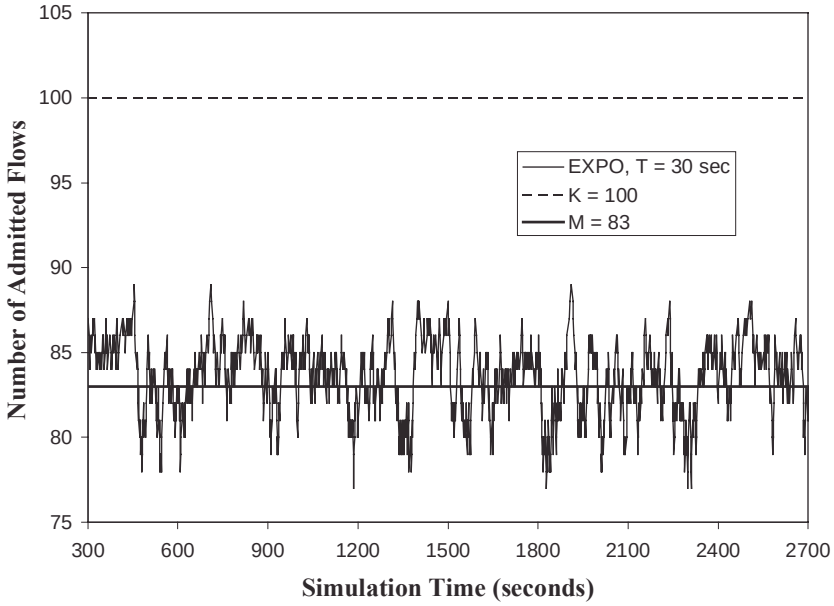


Fig. 6 - Number of admitted flows versus the simulation time, for a window of size T=30 seconds, EXPO case.

We recall that the decision criterion is not aware of the effective number of the admitted sources, but uses a run time estimate. We have proven that, in steady state conditions, the minimum number of flows that can result by such an estimate is given by the value $N_{est,min}$, computed by means of Eq. 10. However, the number of actually admitted flows can be lower than $N_{est,min}$ due to transient effects and to the stack operation. In stationary conditions, the number of allocated sources would be a random variable in the range ($N_{est,min}$,K). In real conditions this is not true. Tab. 2 shows the values of $N_{est,min}$ corresponding to K=100 for three window lengths T=20, 40 and 60 seconds. In the same table, we report the average and the maximum number of admitted sources, N_{adm}, for the EXPO case (with and without the stack mechanism) and for the CBR case. These values have been evaluated by means of simulations.

We note that the average number of admitted sources is lower than $N_{est,min}$ for T=40 and T=60. This fact is

easily explained by considering that the value $N_{est,min}$ reported in the table assumes a fixed number of admitted sources, while in our simulation scenario, the number of active sources varies with time, due to both new flow arrivals (and eventual admission to the system) and departures. The stack mechanism cannot succeed in following exactly this variation, thus limiting the number of admitted sources below the stationary limit. This effect is more evident for large values of the measurement time T. This phenomenon is confirmed by the values of N_{adm} with the stack mechanism turned off: the relevant values are always greater than $N_{est,min}$. Note also that, with the stack turned off, N_{adm} can be greater than K, thus violating the target limit and endangering user perceived performance.

T	Bounds		N_{adm}, EXPO (stack on)		N_{adm}, EXPO (stack off)		N_{adm}, CBR	
	$N_{est,min}$	K	aver.	max	aver.	max	aver.	max
20s	74	100	80	85	84.6	96	80.7	85
40s	86	100	83.3	91	93.2	111	81.1	91
60s	91	100	82.0	91	96.8	119	78.3	91

Tab. 2 - Values of $N_{est,min}$ and of N_{adm}, for three window lengths

The optimal choice of the measurement window results from a trade-off between estimation efficiency and stationary measurement. Fig. 7 shows the utilization coefficient (obtained by means of simulations) as a function of the window size, for different values of the offered traffic, in the EXPO case.

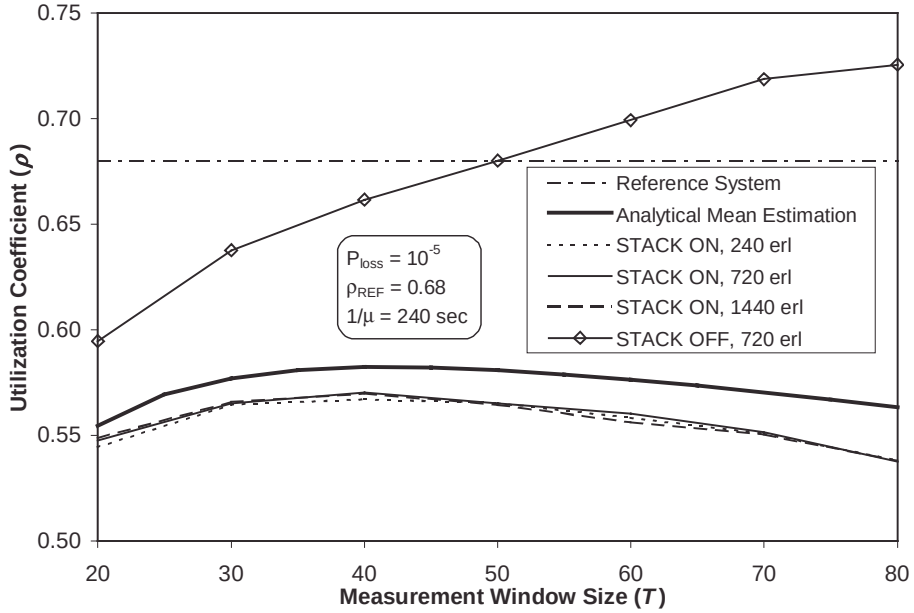


Fig. 7 – Utilization coefficient versus the window size for different values of the offered traffic, EXPO case

Also shown are: i) the horizontal line relevant to the utilization coefficient (or throughput) in the ideal condition of K=100 (labeled "Reference System"); ii) the throughput with the stack mechanism turned off; iii) a mean estimation of the throughput, evaluated analytically in Appendix 1. This figure suggests us the following considerations: i) the throughput with the stack mechanism turned off can overflow the limit value relevant to K=100; this means that the stack mechanism is indeed necessary to provide QoS guarantees; ii) our estimation is indeed close to simulation results; iii) the maximum throughput is achieved with a window size of about 40 seconds. It is intuitive to recognize that the optimal value of the latter parameter, T, is related to the mean call duration (4 minutes in the results shown). We refer to the analysis presented in [GRO99] for a quantification of the optimal measurement length (following [GRO99], an optimal window of 24 seconds is obtained, but our setting is different).

Fig. 8 shows the utilization coefficient as a function of the offered load, evaluated with our upper and lower bounds, evaluated analytically in Appendix 1. Also reported are the ideal throughput deriving from the acceptance rule of [ELM97] (labeled "Reference System") and some simulations results. This figure hints that the performance of our scheme is relatively robust and independent of the external overload.

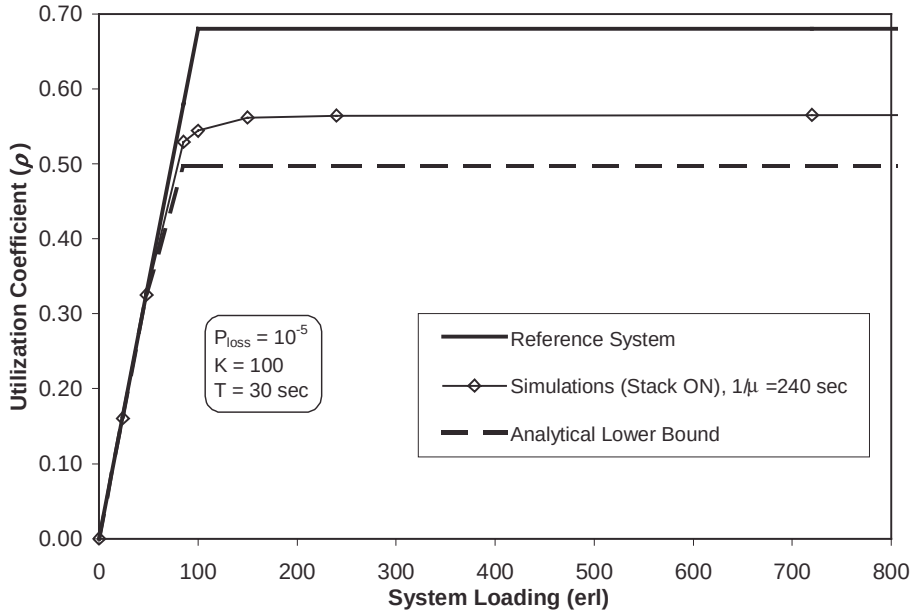


Fig. 8 – Utilization coefficient as a function of the offered load: comparison between Reference System, analytical upper bound and simulation

A final issue to discuss is the effect of the stack implementation when a multi node network scenario is

considered and thus not all probing packets yield to a flow setup. Instead of simulating a multi node topology, we have considered a simplified simulation model where, with a given probability, a probe packet is blocked in later stages of the network. This simulated scenario allows us to evaluate the performance drawbacks induced in GRIP by the lack of explicit signaling messages between routers. In fact, in GRIP, each node independently decides whether to accept or reject a new flow, but there is no explicit mean to determine whether an accepted source has been blocked by later stages of the network.

In multi node networks, the linear stack mechanism described in Section 3.4 apparently should provide overly conservative results, as each accepted flow is accounted, during the measurement time *T*, as an incoming one. However, our numerical results show that the performance degradation induced by the stack implementation is almost negligible. Fig. 9 shows the number of admitted flows in the system versus the simulation time, for different values of the probing packets blocking probability in later stages of the network (offered load equal to 240 Erl). Clearly, in the case of 75% probing packets blocking probability, the number of allocated flows reduces simply because the "valid" offered load gets lower than 100 Erl (25% of 240 Erl). In all the other cases, the throughput performance is very close to that obtained in the optimal case of no probing packets blocking. The most notable effect is a faster transient in the case of no probing packet block.

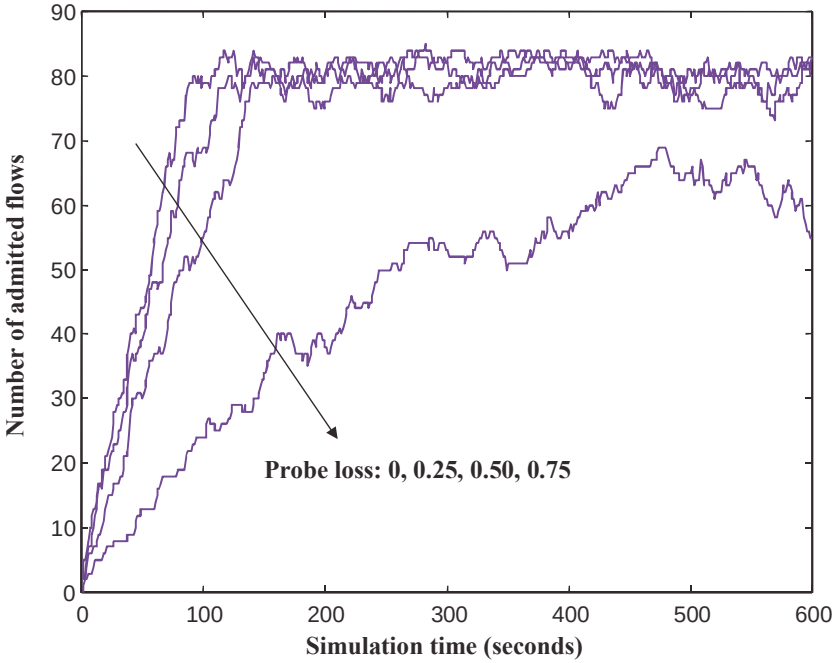


Fig. 9 - Number of admitted flows (CBR) versus the simulation time: T=20 seconds, blocking probability of probing packets equal to 0, 25%, 50% and 75%

To complete our analysis, we have considered a scenario where a router is loaded with new calls arriving at (Poisson) rate 3 calls/s; each call (CBR) has a duration with mean value 4 minutes (exponential distribution). This implies that 720 Erlangs are offered to the link, i.e., more than 7 times the maximum number of calls simultaneously admissible (K=100). In addition, we have assumed that probing packets, once served at the router, may be discarded in the remaining network path. In the simulation run, we have set in the first 1000 seconds a probe discarding probability as high as 90%, between 1000 and 1500 seconds a 0% probe loss, after 1500 seconds a 50% probe discarding probability.

The results of the described simulation scenario are reported in Fig. 10, which shows the number of admitted flows, N_{adm}, versus the simulation time, for both cases of stack protection mechanism active and non active. In addition, the figure shows also the number of estimated flows, N_{est}, with stack off (Eq. 8) and stack on (Eq. 12).

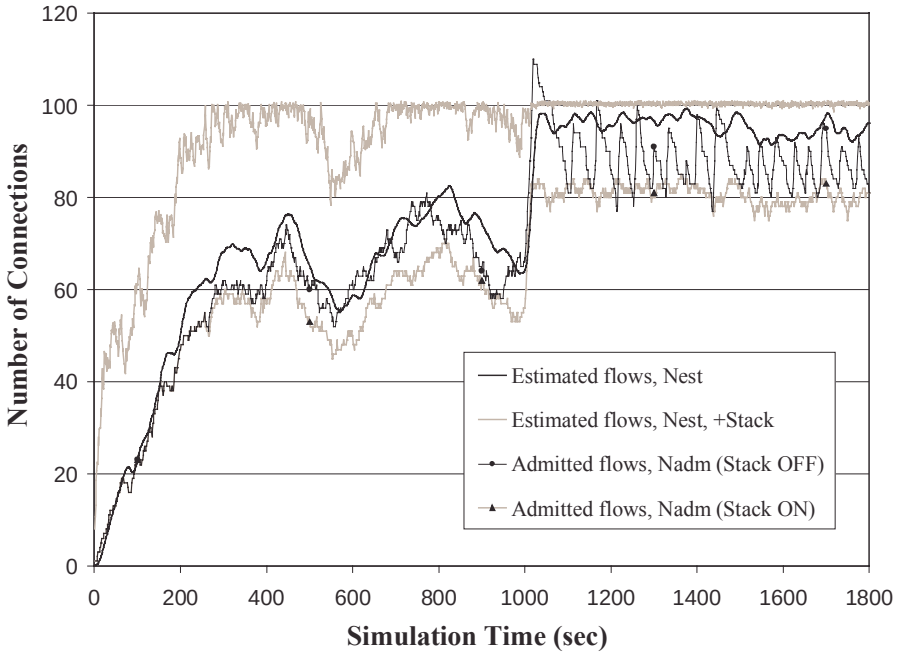


Fig. 10 - Full-fledged GRIP simulation run

The analysis of the figure allows several considerations. During the first 1000 seconds of simulation, the stack protection mechanism slightly penalizes the system throughput, due to a system loading (72 Erl, because of the probe discarding probability equal to 90%) under system capacity (K=100). However, as shown by the simulation at time 1000 seconds, the stack mechanism is indeed necessary to avoid overload. After time 1500, the introduction of a 50% probing discard ratio only marginally affects the throughput performance. Finally, we

note that preliminary results obtained by loading the DLB with MPEG sources (shown in Fig. 11) yield conclusions similar to those drawn with reference to the previous figures. In this case the link capacity is C=100 Mbps and the link buffer size is B=96000 bytes. The values of the DLB parameters are: P_S = 579.6 Kbytes/s; r_S = 54.28025 Kbytes/s; B_{TS} = 12480 bytes. We set, as target performance figure, a packet loss probability, P_{loss}, equal to 10^{-5}. According to the acceptance rule provided in [ELM97], the corresponding maximum number of acceptable flows is K=100. The call arrival rate, modeled as a Poisson arrival process, has been set to 1 call/s, and each call duration has been drawn from an exponential distribution with mean value 4 minutes (a load of 240 erls).

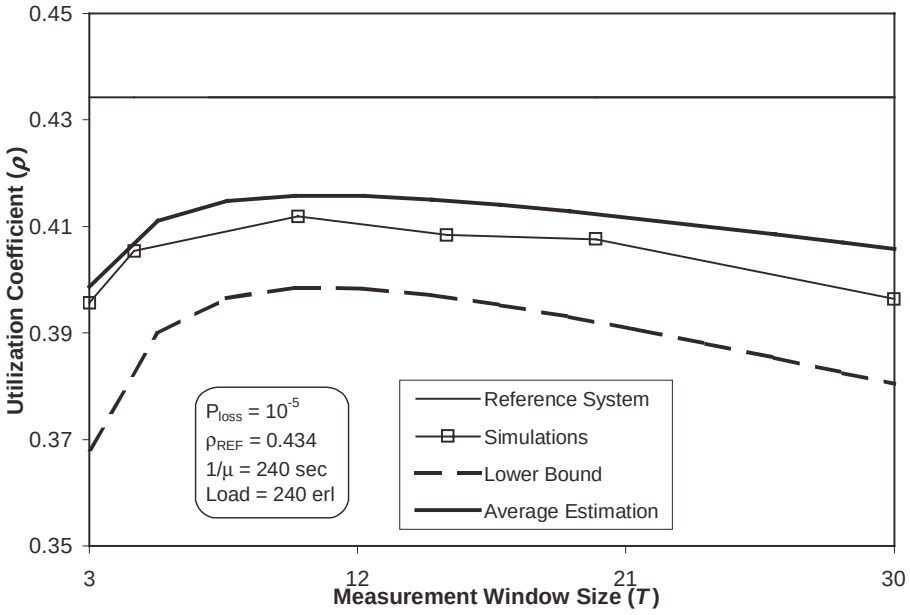


Fig. 11– Utilization coefficient versus the window size, MPEG case

In conclusion, we verified in different operative conditions that the number of admitted sources never overflows the upper limit K=100. Hence, the packet loss requirement used to dimension the value K is always guaranteed in run time operation. This is an important advantage of our proposed (conservative) admission control criterion, and confirms that: (i) hard guarantees can be obtained, thanks to the exploitation of the knowledge of the traffic scenario (i.e., of the DLB parameters); (ii) our scheme appears to provide an explicit performance calibration parameter, i.e., the value K,. The prices to pay are: (i) a system under utilization with respect to the "ideal" value of K (about 18% less), and (ii) the "a priori" knowledge of the traffic regulator parameters adopted at the edge of the network.

The ultimate target of the GRIP operation is to achieve a link utilization as close as possible to the maximum (i.e., K sources), but under the strict QoS requirement that $N_{adm} \leq K$. In order to satisfy the latter requirement, we must accept a throughput penalization, due to the conservative estimation of the number of admitted connections and to the stack protection. We stress that the performance of GRIP must be evaluated in its ability to enforce a specific value of K. The overall throughput efficiency is a direct consequence of the selected value of the "tunable knob" K, and can be thus arbitrarily adjusted by the network operator, who can increase the system utilization at the expense of strict QoS guarantees.

For instance, in our case study, the choice of K=100, results in a system efficiency of about 68%. Then the GRIP operation implies a further decrease of about 18%, so that the overall throughput is about 56%. However, this relatively low value is not entirely imputable to GRIP. In the case study an "ideal" operator has chosen a relatively low value of K. Other values of K with possible other admission rules can be adopted. GRIP just enforces a given choice and it is responsible only in part of the relevant efficiency. For instance, the operator could tune K runtime, to take into account all the overestimation of the mechanism and increase the system utilization at the expense of deterministic guarantees (which would become statistic).

4 CONCLUSIONS

This paper is divided in two parts. In the first part, we have presented a scalable Admission Control scheme. In the authors' opinion, the most important message contained in this part is that GRIP is not a new reservation protocol for the Internet (in this, differing from the SRP protocol [ALM98], from which GRIP inherits some strategic ideas). Instead, GRIP is a novel reservation paradigm that allows independent end point software developers and core router producers to inter-operate without explicit protocol agreements. The evolution from the actual best-effort Internet to a future QoS capable infrastructure can follow two possible migration approaches: (i) an Evolutionary Approach, where new services are offered, but the network architecture is not fundamentally changed and over-dimensioning is the solution to improve the performance (this is in some way in line with DiffServ); and (ii) a Revolutionary Approach, where innovative network architectures are developed, at the expense of inter-operability with older ones; this correspond to introduce real paradigm shifts (this is in some way in line with IntServ).

In the middle between these two extremes, we have identified a smoother Principled Evolutionary Approach. The key is to recognize a new principle, which is revolutionary in its goals and outcomes, but whose ultimate implementation can be delayed in time, by means of subsequent steps of innovation. This approach foresees a continuous evolution in which, in different moments, small and realistic steps of innovation, back-compatible with previous architectures and choices, are asynchronously introduced by different vendors and providers. This requires that a modular and localized concept for innovative features be adopted. We argue that such a concept could be the one of GRIP. GRIP is a very appealing paradigm in terms of mass-market development, since all the required modifications in the migration path toward toll-quality QoS support are incremental, and may be independently offered by different vendors.

In the second part of the paper, we have described the particular GRIP implementation adopted in the SUITED European Union Project. In this scenario, the simplifying (but realistic) assumptions of homogeneous traffic scenario and Dual Leaky Bucket regulators at the end points allow us to provide hard performance guarantees. Our specific proposal combines the benefits of hard guarantees, typical of heavy signaling protocols, with the simplicity of a scheme characterized by the use of distributed and stateless procedures. A fundamental feature of our proposal is that it provides an explicit tunable knob K, which allows the operator to select the optimum operation point (as a trade-off between user perceived performance and system utilization).

Our numerical results show that the system efficiency can be lower than the theoretical maximum. The reason is that all design choices made in this phase of the project were driven by the necessity of an extremely robust scheme, which needs to provide strict performance guarantees. However, we remark that by no means this is an intrinsic limit of our proposal: by using the tunable knob K, independent operators can trade throughput with strict guarantees.

As for future work, we believe that this framework raises a number of teletraffic and system-wide issues. As for the first point, more accurate methods to dimension the window size or other Decision Criteria (based both on different measurement schemes, e.g., [GRO99], or on other principles) could be investigated.

As for the second point, we must better specify the role of GRIP, i.e., whether it is an Internet-wide solution or it is a proposal to be deployed only in restricted ad-hoc domains, with suitable hypotheses. The latter case is

treated in this paper. The former case requires significant additional work. A first effort in this direction can be found in [ID01]. This draft addresses the following issues: i) the GRIP operation in the presence of heterogeneous IP "islands", characterized by different core router decision criterions, or even different paradigms (e.g., IntServ and DiffServ); ii) the Interworking between GRIP and DiffServ and between GRIP and IntServ.

5 APPENDIX 1: PERFORMANCE EVALUATION FOR THE HOMOGENEOUS CASE

A detailed performance evaluation of the effect of the stack variable requires to consider a dynamic scenario, accounting for flow departures and arrivals. In this Section, we derive the mean utilization coefficient and an upper bound and a lower bound of the same quantity.

5.1 Evaluation of the utilization coefficient

Assume that the duration of offered flows is exponentially distributed, with mean value $1/\mu$. To analyze high load conditions (which are the most critical ones), we assume an impulsive load model (see e.g. [GRO99]), in which a new flow is accepted to the system whenever condition (5) is verified, i.e. the router switches from REJECT to ACCEPT state. In other words, users continuously submit new call requests as soon as the system leaves the full occupancy status. Due to the DLB regulation, the average emission rate of each traffic source is assumed equal to the DLB sustainable rate, r_S.

Now, assume that the window size T is small with respect to the mean flow duration $1/\mu$. In these conditions, the probability that a new flow activates and deactivates within the time T is negligible. This approximation has the effect of over-estimating the utilization coefficient, since the STACK mechanism introduces inefficiency. Obviously, this effect increases when the difference ($1/\mu$-T) decreases. In any case, this effect can be accounted for, even if, for simplicity, in this paper we neglect it (given its relatively small contribution to overall performance, see the numerical results).

Let us now define the quantity $T_{react,}$ as the measurement scheme "reaction time" i.e., the time elapsing between the instant of time a flow departs from the system, and the instant of time a state transition REJECT to ACCEPT occurs in the router (and thus, thanks to the impulsive load assumption, a new flow is admitted). Further, we define as $T_{react,ave}$, the average value of $T_{react,}$. In other words, the system can not realize immediately

that a flow has switched off because, since the traffic sources are VBR, a temporary decrease of the measured bit rate could be due to source activity variations. Thus the measurement procedure needs a time $T_{react,}$ to distinguish between real flow de-activations and statistical fluctuations of active source bit rates. Considering that, owing to the impulsive load assumption, each departing flow will be replaced in average after a time $T_{react,ave}$ with a new incoming one, the average number of active flows during the time window T is given by:

$$\overline{N} = N_0 + N_d \frac{T - T_{react,ave}}{T} \tag{13}$$

being N_0 the number of flows that remain active during the whole measurement window time T, and being N_d the number of flows departed during T, and replaced by a newly incoming flow. In turns, the average number of departing flows in the time T is given by

$$N_d = \overline{N}\mu T \tag{14}$$

which combined with (6) yields:

$$N_0 = \overline{N}\left(1 - \mu\left(T - T_{react,ave}\right)\right) \tag{15}$$

We are now able to write condition (12), replacing $A(T)$ with the sum of the average contribute of the N_0 flows that remain active during T, and of the N_d departing/arriving flows:

$$\frac{N_0 r_S T + N_d r_S (T - T_{react,ave})}{r_S T - B_{TS}} + STACK = K \tag{16}$$

Noting that flow arrivals are uniformly distributed in the time window T, the average stack value is $STACK=N_d/2$. Substituting this value in (16), and owing to (14), (15), we can finally write a single equation which yields the average system throughput as:

$$\rho = \frac{\overline{N} r_S}{C} = \frac{K(1 - \frac{T_{OFF}}{T})}{\left(1 + \frac{\mu}{2}(T - T_{OFF})\right)} \frac{r_S}{C} \tag{17}$$

where $T_{OFF} = B_{TS} / r_S$ is the maximum silence period allowed by the DLB. Note that to obtain this result, we do not have to explicitly provide a value for $T_{react,ave}$ (which, in any case, is trivially shown to be given by

$T_{react,ave} = (T - T_{OFF})/2$), since in the evaluation of the utilization coefficient, we have both a positive and a negative contribution of this quantity, which balance themselves.

From Eq. 17, it is straightforward to calculate the optimal value T_{opt} of the measurement window T that maximizes ρ:

$$T_{opt} = T_{OFF} + \sqrt{2\frac{T_{OFF}}{\mu}} \tag{18}$$

This is an important result, since it allows dimensioning the measurement window.

Equations (17) and (18) suggest also the following considerations. If the mean flow duration increases, T_{opt} will increase too (for $\mu \to 0 \Rightarrow T_{opt} \to \infty$). As regards the utilization coefficient, when $\mu \to 0$, the STACK contribution decreases and ρ increases with the measurement window size T:

$$\lim_{\mu \to 0} \overline{\rho} = \frac{K\left(r_S - B_{TS}/T\right)}{C} \xrightarrow{T \to \infty} \frac{Kr_S}{C} \tag{19}$$

In other words, as discussed at the end of Section 3.3, in these conditions ($\mu \to 0, T \to \infty$) the number of flows is evaluated exactly and the utilization coefficient is equal to the ideal one, (i.e. $N=K$).

To give a physical meaning to the above equations, we note that the effect of the STACK protection is taken into account in (10), with respect to the static formula (i.e., when $\mu \to 0$), by means of the term in the denominator $(T - T_{OFF})\mu/2$; such term is equal to the mean reaction time multiplied by the mean rate of departures/arrivals (μ).

5.2 Evaluation of an upper bound of the utilization coefficient

In order to evaluate an upper bound of the utilization coefficient, denoted as ρ_{upp}, we make the following assumptions:

- each flow emits according to its minimum emission profile, i.e. it emits $r_S\ (T_i - T_{OFF})$ bits during the measurement window;
- we consider the minimum value for the reaction time, i.e. $T_{react,min}=0$;
- the per-flow STACK contribution is equal to zero (instead of 1/2).

Consequently, following the same approach used in the preceding section.

$$N_{upp} = K,\ \rho_{upp} = \frac{Kr_S}{C} \tag{20}$$

This is a rather obvious result, reported only to show that the particularization of our derivation gives correct results. In other words, the upper bound of the utilization coefficient is equal to ideal one, when K flows are admitted in the system

5.3 Evaluation of a lower bound of the utilization coefficient

We first derive the measurement scheme "maximum reaction time" $T_{react,max,}$ i.e., the maximum possible value of T_{react}. It is easy to deduce that the value of $T_{react,max}$ is the solution of the following equation:

$$\frac{Nr_S T}{r_S T - B_{TS}} - \frac{Nr_S (T - T_{react,max}) + (N-1) r_S T_{react,max}}{r_S T - B_{TS}} = 1 \Rightarrow T_{react,max} = T - \frac{B_{TS}}{r_S} = T - T_{OFF} \tag{21}$$

This equation is obtained by imposing that the difference between the left value of (5), computed assuming that each flow emits at its average rate and no flows depart, and the left value of (5) computed in the assumption that the departure of one flow at time $(T- T_{react,max})$ is detected after $T_{react,max}$ seconds.

In order to evaluate a lower bound to ρ, , denoted as ρ_{low}, we make the following assumptions:

- each flow emits according to its maximum emission profile, i.e. it emits $r_S (T_i+T_{OFF})$ bits during the measurement window;
- we consider the maximum value for the reaction time, i.e. $T_{react,max}=T-T_{OFF}$;
- the per-flow STACK contribution is equal to one (instead of 1/2).

Following the same approach of Section 5.1, we obtain:

$$N_{low} = \frac{K(T - T_{OFF})}{\mu T^2 + T + T_{OFF}(1 - \mu T_{OFF})},\ \rho_{low} = \frac{N_{low} r_S}{C} \tag{22}$$

Finally, in Fig. 12 the behavior of $\overline{\rho}$ is reported for different value of the mean flow duration $1/\mu$: (100, 200, 400, 800 seconds) and the limit curve, represented by eq. 12. If the mean flow duration increases, the utilization coefficient increases too and its behavior depends mainly on single parameter: the measurement window size T.

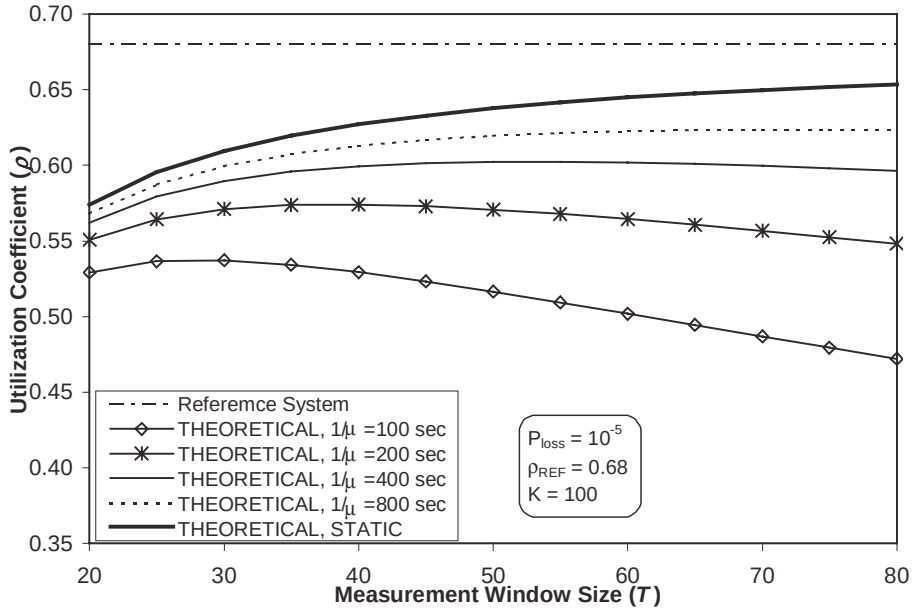


Fig. 12 - Utilization Coefficient versus the Measurement Window Size T for different values of the mean flow duration 1/μ (100, 200, 400, 800 seconds)

6 APPENDIX 2: THE HETEROGENEOUS CASE

To extend GRIP to a heterogeneous traffic scenario (i.e. to a mix of traffic sources regulated with different DLB parameters) we need some introductory considerations. Let us assume that the sources are divided in *I* traffic classes, each comprising independent and homogeneous sources (i.e., with the same DLB parameters). To go further, we note that, *in the homogeneous case*, the evaluation of the number of admissible flows *K* is equivalent to the evaluation of the so-called equivalent bandwidth of each flow, denoted as *e*. The equivalent bandwidth concept is well known in the literature and represents the amount of bandwidth that must be assigned to each flow, in a statistical multiplexing framework, so as to reach some performance levels. The equivalent bandwidth is typically comprised between the mean and the peak bit rate of the traffic source. In our case, $e=C/K$, where *C* is the link capacity (and where *K* is evaluated as in [EMW95, ELM97], once the buffer size *B* and the performance parameters of interest, e.g. $P_{loss}=10^{-5}$ have been fixed). The ideal admission control function "accept new setup requests as long as the number of admitted flows *N* is less than *K*", can then be rewritten as "accept new setup request as long as the sum of the equivalent bandwidth of accepted flows ($N*e$) is less than *C*". In GRIP, due to the lack of signaling, we do not know *N*, so we estimate it; then we make use of the equivalent bandwidth concept by comparing the estimated *N* to *K*. Finally, we add the stack mechanism. In other

words GRIP exploits the DLB characterization two times. The first time to estimate the number of admitted flows and the second time to decide if a new setup request can be accepted. Note that, to this end, each router must be aware of the DLB parameters.

The equivalent bandwidth concept can be in principle easily extended to the heterogeneous case. In [EMW95, ELM97] the Authors propose an efficient algorithm to evaluate this quantity, which in the assumption of DLB regulated sources is additive even in the heterogeneous case. They evaluate the equivalent bandwidth of the *i*-th traffic class e_i, $\{1 \le i \le I\}$ as $e_i = C/K_i$, where K_i is evaluated for each class in isolation, i.e., in a homogeneous system. In other words, the limit value K_i is the number of flows such that a link with capacity C and buffer B, loaded only with traffic belonging to the *i*-th class, offers pre-defined performance levels. With this approach (which is shown to be conservative, even if it leads to a loss of efficiency), the ideal admission rule remains a simple sum: new setup requests are accepted as long as

$$\sum_{i=1}^{I} N_i e_i \le C \tag{23}$$

where N_i is the number of admitted flows belonging to class *i*.

To extend GRIP to the heterogeneous case, we have to estimate the number of admitted flows of each class and then apply the above admission rule. To this end, we can identify three architectural alternatives. The first (trivial) alternative is shown Fig. 13a. Here we assume that each class is handled in a separated way and recognized by routers by assigning different *pairs* of DS codepoints to different traffic classes (i.e., each class as a DSCP for Probing packets and another one for Information packets). Thus, we need 2**I* different DSCPs and 2**I* different logical queues. Packets belonging to class *i* are classified on the basis of their DSCP tag and dispatched to the relevant queue (Probing or Information). However, we do not require to signal explicit information about the *traffic mix composition*, that is how many active flows per each class are present in each node.

The architecture is complex but the extension of GRIP is straightforward, since the heterogeneous case is reduced to a combination of homogeneous ones. The admission rules becomes:

$$\left\lfloor \frac{A_i(T)}{r_{S,i}T - B_{TS,i}} + STACK_i \right\rfloor \le K_i - 1, \quad with \sum_{i=1}^{I} K_i e_i \le C \tag{24}$$

where $STACK_i = \left(1 - \frac{t - t_1}{T}\right)$ for each new admitted flow. In (24), $A_i(T)$ is the number of bytes emitted during the window T by traffic sources belonging to the i-th class (i.e., at the i-th Information queue) and K_i is evaluated off-line as discussed above.

Note that this alternative implies that each node is aware of the DLB parameters of all possible classes. This architectural alternative is somehow acceptable only in presence of a small number of traffic classes (e.g., a class could be IP telephony), even if it is still compliant with the DiffServ approach. An advantage of this architecture is that it allows implementing procedures to fairly divide the overall capacity among the traffic classes.

In the second alternative (see Fig. 13b), we assume that Probing packets belonging to different classes are handled in a separated way, with multiple probe queues. These packets are recognized by routers by assigning them different DSCPs, while the Information packets of all the I traffic classes are multiplexed together in a common queue. Each class has a DSCP for Probing packets, while the Information packets of all classes share the same DSCP. Thus, we need I+1 different DSCPs and I+1 different logical queues. As above, we do not require to signal explicit information about the traffic mix composition. This duty is assigned to the measuring module, which operates on traffic aggregates only. However, the Decision Criterion has to be suitable modified with respect to the homogeneous case, in order to take into account the presence of different traffic profiles, while still guaranteeing performance.

In this alternative, it is not possible anymore to estimate the admitted flows of each class, N_i, as done above; thus if we want still to guarantee performance, we are left with a worst case approach. Our measurement procedure evaluates now the number $A(T)$ of bytes emitted within a window of size T by *all* traffic classes. To define a GRIP allocation rule we have to evaluate the traffic mix that maximizes the overall equivalent bandwidth

$$e_{TOT} = \sum_{i=1}^{I} N_i e_i \qquad (25a)$$

under the constraints

$$\begin{cases} \sum_{i=1}^{I} N_i a_i^{min} \le A(T) \quad with \quad a_i^{min} = r_{S,i}T\text{-}B_{TS,i} \\ 0 \le N_i \le K_i \end{cases} \tag{25b}$$

In other words, we have to find the values of N_i $\{1 \le i \le I\}$ under the constraint that the admitted flows must be such that they emit $A(T)$ bytes, when using a minimum DLB emission profile (this is the meaning of the apex "*min*" to quantity "*a*").

The main difficulty in resolving the problem in (25) is the need to find an integer solution, since we are dealing with numbers of flows. We start by resolving the associated continuous problem (i.e. by assuming that N_i is a continuous variable) and then we apply a floor operator to the solution (which is a slightly conservative operation), obtaining:

$$\begin{cases} N_x = \left\lfloor \dfrac{A}{a_x^{min}} \right\rfloor \\ N_i = 0 \qquad for \quad i \ne x \end{cases} \tag{26}$$

where *x* is the index of the class with the greatest value of the ratio e_i / a_i^{min} $\{1 \le i \le I\}$.

Because of the lack of any information about the composition of the traffic mix, the measurement procedure reduces the heterogeneous case to an homogenous one, in which, for estimation purposes, only the traffic class endowed with the worst estimation of resource utilization is considered as present in the mix. Such class is labeled with the index "*x*". The worst estimation results from assigning the greatest equivalent bandwidth to a number of bits obtained by assuming the minimum DLB emission profile. The GRIP allocation rule relevant to the *i*-th probing class becomes:

$$\left\lfloor \frac{A}{r_{S,x}T - B_{TS,x}} + \sum_{i=1}^{I} STACK_i + \frac{e_i}{e_x} \right\rfloor \le K_x \tag{27}$$

where $STACK_i$, for each new admitted flow, is equal to:

$$STACK_i = \frac{e_i}{e_x}\left(1 - \frac{t - t_1}{T}\right) \tag{28}$$

and the ratio e_i / e_x accounts for the new incoming flow. Note that this architecture still allows implementing procedures to fairly divide the overall capacity among the traffic classes. As regards performance, this architecture is simpler than the previous one, but the price to pay is a potential smaller system efficiency.

As a third and last alternative, we propose the architecture shown in Fig. 13c. Here we assume that the Probing packets of all the *I* traffic classes are multiplexed together in the same queue and that the Information packets of all the *I* traffic classes are multiplexed together in another common queue. All the traffic classes share the same pair of DSCP tag: one for Probing packets and one for Information packets. Thus we need only two DSCPs. This time we have to adopt a worst case approach not only for the measurement procedure but also for the admission rule. Recall that we use the DLB characterization two times. The first one to estimate the number of admitted flows and the second time to decide if a new setup request can be accepted. Since in this alternative we can not distinguish between probes belonging to different traffic classes, we are forced to interpret each setup request as belonging to the class with the greatest equivalent bandwidth. Let us define e_{max} the maximum value of e_i $\{1 \le i \le I\}$.

The GRIP allocation rule relevant to the *i*-th probing class becomes:

$$\left\lfloor \frac{A(T)}{r_{S,x}T - B_{TS,x}} + STACK + \frac{e_{max}}{e_x} \right\rfloor \le K_x \tag{29}$$

where *STACK,* for each new admitted flow, is equal to:

$$STACK = \frac{e_{max}}{e_x}\left(1 - \frac{t - t_1}{T}\right) \tag{30}$$

The ratio e_{max} / e_x accounts for the new incoming flow and $A(T)$ is the number of bytes emitted within a window of size *T* by *all* traffic classes. The stack variable is incremented by using a scaling factor based on the greatest equivalent bandwidth, to take into account setting up flows in a conservative way. This last alternative is the simplest one. As in the previous ones, we do not require to signal explicit information about the *traffic mix composition*. In addition, this alternative implies that each node has to know the DLB parameters of only one traffic class and the value e_{max} (assuming that the considered traffic classes do not vary). The prices to pay are: i)

a potential smaller system efficiency with respect to the previous cases; ii) the impossibility of implementing procedures to fairly divide the overall capacity among the traffic classes.

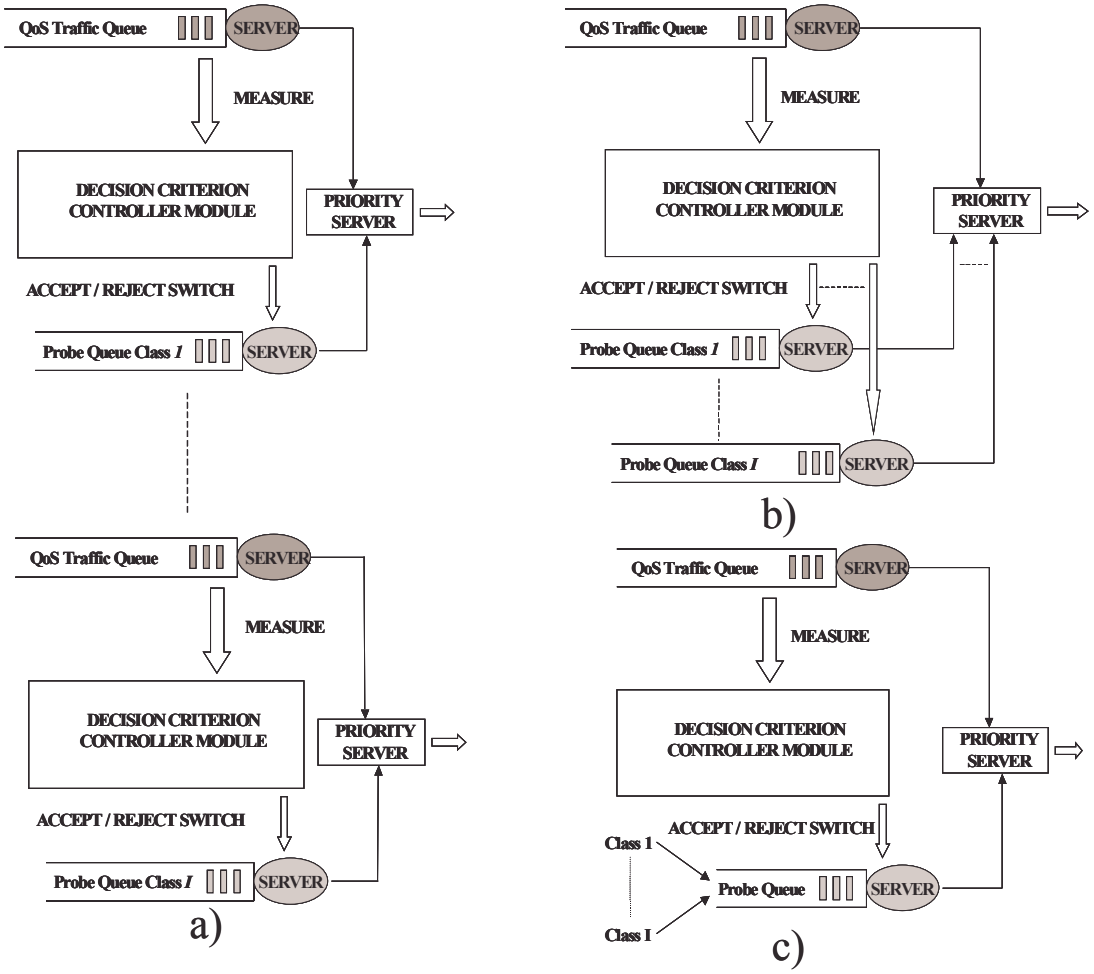


Fig. 13 - Heterogeneous case logical scheme, multiple and single probe queue implementations

6.1 Performance Evaluation of The Heterogeneous Traffic Scenario

In the following we propose a simplified performance analysis, using the following hypothesis:

- steady-state traffic condition (all the transient are excluded);
- use of the fluidic model, by considering also non-integer values for flow number for each class;
- no stack mechanism is considered.

In order to estimate the mean value for the utilization coefficient, a reasonable assumption is that the flows are emitting according to the mean emission profile, obtaining:

$$\rho(\overline{N}) = \sum_{i=1}^{I} \frac{N_i r_{S,i}}{C} \text{ on the } I \text{ dimensional surface } \sum_{i=1}^{I} N_i a_i^{mean} = A_{min} \text{ with } a_i^{mean} = r_{S,i} T \qquad (31)$$

The value A_{min} represents the minimum value of counted bits during the sliding window such that the probe queue will switch to the REJECT state, with $A_{min} = (r_{S,x}T - B_{TS,x})\lfloor C / e_x \rfloor = (r_{S,x}T - B_{TS,x})K_x$.

We introduce a more comfortable notation $N_i = \alpha_i K_i^{mean}$ using the scaling factor α_i with respect to K_i^{mean} (e.g. assuming that only flows belonging to this class are present, but with class *"x"* used for DC):

$$\rho(\overline{N}) = \sum_{i=1}^{I} \frac{N_i r_i}{C} = \sum_{i=1}^{I} \frac{K_i^{mean} \alpha_i r_i}{C}$$

$$\begin{cases} \sum_{i=1}^{I} \left(K_i^{mean} \alpha_i\right) a_i^{mean} = A_{min} \\ K_i^{mean} = \dfrac{A_{min}}{a_i^{mean}} = \dfrac{A_{min}}{r_{S,i}T} \\ 0 \le \alpha_i \le 1 \end{cases} \tag{32}$$

It is clear that, independently by the composition of traffic mix, it results from eq. 32:

$$\left. \rho(\overline{N}) = \right|_{\sum_{i=1}^{I}\left(K_i^{mean} \alpha_i \, a_i^{mean}\right) = A_{min}} \left. \frac{\sum_{i=1}^{I} r_{S,i} K_i^{mean} \alpha_i}{C} = \frac{A_{min}}{TC} = \frac{K_x}{C}\left(r_{S,x} - \frac{B_{TS,x}}{T}\right) = \overline{\rho(T)} \right|_{\sum_{i=1}^{I}\left(K_i^{mean} \alpha_i a_i^{mean}\right) = A_{min}} \tag{33}$$

Hence, under the preceding hypotheses, the mean value for ρ is bounded by the utilization coefficient of the homogeneous case when only the class "*x*" is considered, while the reference system has an utilization coefficient that is an intermediate value (depending of traffic mix sharing the QoS queue) between the maximum and minimum values of ρ of different classes in isolation:

$$\lim_{T \to \infty} \overline{\rho(T)} = \lim_{T \to \infty} \frac{(r_{S,x} - B_{TS,x}/T)K_x}{C} = \frac{K_x r_{S,x}}{C} = \rho_x \tag{34}$$

In general class *"x"*, chosen for estimation purposes, is a function of *T*; however, for enough large values of the measurements window, this class remains the same.

According to the results we have illustrated so far, it is obvious that this analysis is not suitable when dynamic phenomena occur. In particular, we have to take into account several effects. In this section we develop a model for estimate the maximum and minimum value of ρ, in :

- the stack mechanism has to be considered: depending of the algorithm chosen, the contribution of each newly admitted flow will be changed by a suitable factor (depending of the chosen algorithm);
- the estimation process, performed by the class labeled as *"x"*, will realize within different times the departures of flows belonging to different traffic classes (as in the homogeneous case, we do not explicitly calculate the value for $T_{react,i}$);
- different value of μ_i (and in general of the offered traffic) for different classes will lead to different time scales for different classes.

Following the preceding considerations, we have calculated a simple model for estimate the variation area for the mean value of ρ in the heterogeneous case: we calculate the utilization coefficient when there are in the system only flow of class *"i"*, but estimation is performed with class *"x"*. We define the coefficient $\gamma=e_{max}/e_x$ and coefficient $\beta_i=r_{S,i}/r_{S,x}$. Following the same approach used in Section 5.1, we obtain:

$$\overline{\rho_i} = \frac{K_x\left(1-\frac{T_{OFF,x}}{T}\right)}{\left(1+\frac{\mu_i\gamma}{2\beta_i}(T-T_{OFF,x})\right)}\frac{r_x}{C} \qquad (35)$$

Depending of values of coefficient μ_i, β_i and γ, the superior/inferior approximation can be provided by any class. In particular, the lower the value of μ_i and the greater the value of $\rho_i = \frac{\overline{N_i} r_{S,i}}{C}$; the greater the value of β_i and greater the value of ρ_i; the lower the value of γ and greater the value of ρ_i.

6.2 Numerical results

In order to test the effectiveness of the heterogeneous approach, we carried out simulation with two traffic classes. Results presented here are as a case study. As in the homogeneous case, we make no assumption on the traffic source behavior. We used on-off exponential sources (EXPO) with different DLB parameters, to represent two different traffic classes. For the first kind of sources (labeled as Class 1), the On and Off state durations are exponentially distributed, with average values of 352 ms and 650 ms respectively. The bit rate during the On period is equal to 4 Kbytes/s. For the second kind of sources (labeled as Class 2), the On and Off state durations are exponentially distributed, with average values of 400 ms and 1000 ms respectively. The bit rate during the

On period is equal to 6 Kbytes/s. We consider a generic router output link with parameters: link rate, C=2.048 Mbps; buffer size, B=53000 bytes. We set, as target performance figure, a packet loss probability, P_{loss}, equal to 10^{-5}, as in the homogeneous case. Tab. 3 reports the DLB parameters of the sources and the maximum number of acceptable flows K_i, according to the acceptance rules provided in [ELM97]. The call duration for both the sources has been drawn from an exponential distribution with mean value 4 minutes. In order to stress the heterogeneous algorithm operation under different traffic conditions, we considered three different traffic mixes. We defined a parameter, α, in order to identify the composition of the mix. It represents the percentage of the overall utilization coefficient ρ provided by the flows of Class 1, i.e. $\alpha = \rho_1 / \rho_{TOT}$. We set this value to 0.25, 0.50, 0.75, in order to analyze the system response with different traffic mixes.

	P_S	r_S	B_{TS}	K
Class 1	4 Kbyte/s	1.7 Kbyte/s	5300 bytes	100
Class 2	6 Kbyte/s	2.05 Kbyte/s	6360 bytes	76

Tab. 3: Source description and DLB parameters.

For each value of the parameter α, we evaluated the target numbers of flows for both the two classes according to the rules in [ELM97]. Such values are reported in Tab. 4, with the relevant values of utilization coefficient.

	α=0.25	α=0.50	α=0.75
K_1	23	48	74
K_2	58	40	20
$\rho_{TOT,REF}$	0.617	0.639	0.652

Tab. 4: Reference system performance for different values of the traffic mix.

The arrival rate, modeled as a Poisson arrival process, has been set to different values, depending of the composition of the traffic mix. These values have been chosen in order to allow a slight system overload (110%)

with respect to the target values in Tab. 3. It is simple to show that, in order to match the value of α, it is enough to set the ratio between the arrival rates λ_i ($i=1,2$) equal to the ratio of K_i in Tab. 4. These values are reported in Tab. 5.

	α=0.25	α=0.50	α=0.75
λ_1	0.105 s^{-1}	0.216 s^{-1}	0.339 s^{-1}
λ_2	0.266 s^{-1}	0.18 s^{-1}	0.0917s^{-1}

Tab. 5: Source arrival rates for different values of the traffic mix.

In Fig. 14 we show simulation results relevant to the heterogeneous case. On the abscissa we report the measurement window size T, while on the ordinate the value of the utilization coefficient, ρ_{TOT}. For what concerns the comparison between the approach a single traffic and probe queue and the one using separated traffic and probe queues for each class of traffic, we used simulation results for the former, while for the latter we reported the estimation provided by means of Eq. 17.

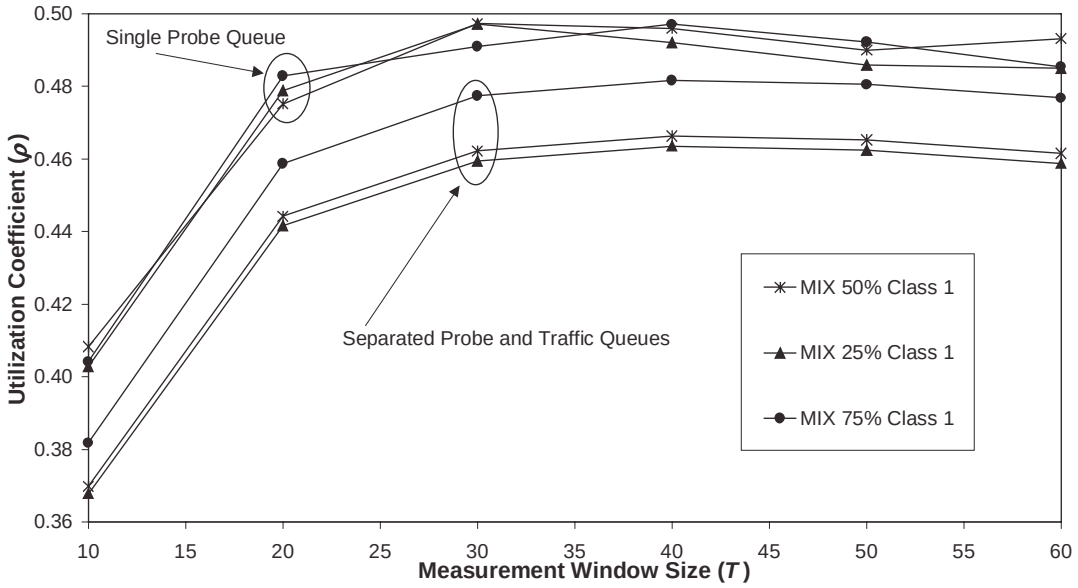


Fig. 14: Utilization Coefficient versus the Measurement Window Size: the Heterogeneous case for different value of the parameter α.

From the analysis of such figure, we can draw some basic considerations:

- the overall utilization coefficient for Reference system result always higher than the one provided by GRIP algorithm (i.e., $\rho_{TOT} < \rho_{TOT,REF}$). However, since our algorithm, also in the heterogeneous version, provides strict QoS guarantees, we expected such behavior, similar to the homogeneous case;
- for our choice of the DLB parameters (a variation of the 50% for the peak rate, a variation of the 20% for the sustainable rate and for the token burst size), the effectiveness of the approach with a unique traffic and probe queue for all the traffic classes is always higher than the approach with separated queues. This phenomenon is explainable in terms of statistical multiplexing gain, and is a good reason to move towards a solution simpler and more effective. However, this choice, even if simpler in terms of architectural design, could not be always the best in terms of utilization coefficient. In particular, when the DLB parameters of the sources loading the queues are very different, the solution with separated traffic and probe queues could be more efficient, even if it exhibits a major complexity.

Finally, a further observation has to be made about the influence of parameter α: the trend of simulation result substantially agrees with the results of Reference system, even if, in the simulation curves, slight variations occur. A more tight control on traffic mix could be obtained by means of using the approach with a single traffic queue and separated probe queues. This approach, due to a more suitable implementation of the stack mechanism, should provide also better results in terms of utilization coefficient, even if it is slightly more complex.

7 APPENDIX 3: AN EFFICIENT SCHEME FOR THE STACK IMPLEMENTATION

From the computational point of view, the transient management presented in previous section can be quite "heavy". In fact, when a probe packet is served, a function call (responsible for temporized updating of the STACK variable by means of a linear decrementing action) is activated, thus creating a function call vector whose maximum size is equal to K_i (the so called "stack").

The MBAC module will consider the sum of each STACK variable (scaled by e_i / e_x) in order to control the admission. The goal is to simplify the treatment, in order to have lower computational cost for this not fundamental issue. The following strategy has been adopted:

- a vector for each probe queue is considered;

- the length L of each the vector is independent of QoS queue size (i.e. capacity and buffer size): the measurement window T is divided in L time interval of length T/L, to which the probe packets served during them are associated.

The approximation process consists in the evaluation of the stack value for a single equivalent flow. We assume the following approximation: if the current decision time is represented by t, all the probe packets served in the time interval $\left(t-(l+1)(T/L),t-l(T/L)\right]$, labeled as $m_i(l)$, are considered served at $t-l(T/L)$, $l=0,..,L-1$. So we can define the under estimation Δ of the mean time distance between the serving instant of the equivalent probing packet (for all the probe packets served during the last T seconds) and the current decision time (i.e. of $t-t_1$, using the notation of Eq. 11), in order to provide an upper bound to the STACK content. If the total number of served probe packets of class *"i"* (i.e. new flows, assuming the more general hypothesis of multiple probe queues) in the window $(t-T, t]$ is equal to $M_i = \sum_{l=0}^{L-1} m_i(l)$, the value of Δ is equal to:

$$\Delta = \frac{\sum_{l=0}^{L} m_i(l)\left(l\frac{T}{L}\right)}{M_i} \tag{36}$$

and so the approximated (upper bound) value of the stack for class "i" is equal to:

$$STACK_i' = M_i\left(1-\frac{\Delta}{T}\right)\left(\frac{e_i}{e_x}\right) \tag{37}$$

Then, the definitive formula used for CAC rules is:

$$\left\lfloor \frac{A}{r_{S,x}T - B_{TS,x}} + \sum_{i=1}^{I} STACK_i' + \frac{e_i}{e_x} \right\rfloor \le K_x \tag{38}$$

However, it is possible to reduce this approach to a single vector of L elements by using the following consideration: we maintain a single stack, and each new flow admitted directly scales the value of the parameters M and $m(l)$ of the value e_i/e_x. This implementation is equivalent to the preceding for functionality, but with minor complexity: in fact it leads to canceling in (38) the sum operation of the $STACK_i'$.

REFERENCES

[ALM98] W.Almesberger, T.Ferrari, J. Y. Le Boudec: "SRP: a Scalable Resource Reservation Protocol for the Internet", IWQoS'98, Napa (California), May 1998.

[BCP00] G. Bianchi, A. Capone, C. Petrioli," Throughput Analysis of End-to-End Measurement Based Admission Control in IP", Proc. of IEEE Infocom 2000, Tel Aviv, Israel, March 2000.

[BIA00] G. Bianchi, A. Capone, C. Petrioli: "Packet management techniques for measurement based end-to-end admission control in IP networks", IEEE/KICS Journal of Commun. Networks, June 2000.

[BJS00] L. Breslau, S. Jamin, S. Schenker: "Comments on the performance of measurement-based admission control algorithms", IEEE Infocom 2000, Tel-Aviv, March 2000.

[BOR99] F. Borgonovo, A. Capone, L. Fratta, M. Marchese, C. Petrioli, "PCP: A Bandwidth Guaranteed Transport Service for IP networks", IEEE ICC'99, June 1999.

[BRE00] L. Breslau, E. W. Knightly, S. Schenker, I. Stoica, H. Zhang: "Endpoint Admission Control: Architectural Issues and Performance", ACM SIGCOMM 2000, Stockholm, Sweden, August 2000.

[ELE00] V. Elek, G. Karlsson, "Admission Control Based on End-to-End Measurements", Proc. of IEEE Infocom 2000, Tel Aviv, Israel, March 2000.

[ELM97] A. Elwalid, D. Mitra: "Traffic shaping at a network node: theory, optimum design, admission control", IEEE Infocom 97, pp. 445-455.

[EMW95] A. Elwalid, D. Mitra, R. H. Wentworth: “A New Approach for Allocating Buffers and Bandwidth to Heterogeneous, Regulated Traffic in an ATM Node”, IEEE J.S.A.C. Vol. 13, N. 9, August 1995, pp. 1115-1127.

[GKE99] R. J. Gibbens, F. P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network", 16th ITC, Edimburgh, June 1999.

[GRO99] M. Grossglauser, D. N. C. Tse: "A Time-Scale Decomposition Approach to Measurement-Based Admission Control", Proc. of IEEE Infocom 1999, New York, USA, March 1999.

[ID01] G. Bianchi, N. Blefari-Melazzi, M. Femminella: “A Migration Path to provide End-to-End QoS over Stateless Networks by Means of a Probing-driven Admission Control”, Internet Draft, draft-bianchi_blefari-end-to-end-QoS-00.txt, work in progress, http://www.ietf.org/ID.html

[MST99] Mertzanis, G. Sfikas, R. Tafazolli, B. G. Evans: “Protocol Architecture for Satellite ATM broadband Networks, Communications Magazine, March ’99, pp. 46-54.

[R2205] R. Braden, L Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification", RFC2205, September 1997.

[R2210] J. Wroclawsky, "The use of RSVP with IETF Integrated Services", RFC2210, September 1997.

[R2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definitions of the Differentiated Service Field (DS Field) in the Ipv4 and Ipv6 Headers", RFC2474, December 1998.

[R2475] S. Blade, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC2475, December 1998.

[R2990] G. Huston, "Next Steps for the IP QoS Architecture", RFC2990, November 2000.

[R2998] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Speer, M., Braden, R., Davie, B., Wroclawski, J. and E. Felstaine, "A Framework for Integrated Services Operation Over DiffServ Networks", RFC 2998, November 2000.