Measuring QoS in class-based IP networks using multipurpose colored probing patterns

Solange Rito Lima[a], Paulo Carvalho[a] and Vasco Freitas[a]

[a]University of Minho, Department of Informatics, 4710-059 Braga, Portugal

ABSTRACT

Multiclass IP networks open new dimensions and challenges on active monitoring as efficient strategies of in-band probing are required to sense each class performance without causing noticeable side-effects on real traffic. In our study, we provide new insights on how to perform efficiently active monitoring in these networks, suggesting the use of light and multipurpose probing streams able to capture simultaneously the behavior of multiple QoS metrics of each class. Considering one-way-delay, jitter and loss metrics, we explore different spatial-temporal characteristics of probing, focusing on finding patterns adjusted to each class measurement requirements. We demonstrate that commonly used probing streams fail to capture these metrics simultaneously and we propose novel colored probing patterns able to increase multipurpose active monitoring efficiency. As test environment, we consider a diffserv domain where admission control resorts to feedback from edge-to-edge active monitoring to dynamically control hard real-time, soft real-time and elastic traffic classes. Comparing graphically and statistically the probing and passive measurement outcome of each class, the obtained results show that despite being difficult to match the scale and shape of multiple metrics, a single and properly colored probing stream can capture close and simultaneously the behavior of one-way-delay, jitter and loss, for low in-band probing rates.

Keywords: Active Monitoring, Traffic Measurements, Quality-of-Service, Traffic Control, Multiclass Networks, Service Differentiation, Edge-to-edge Service Performance

1. INTRODUCTION

Monitoring performance of today's IP networks assumes a crescent and crucial role both for service providers and customers in the support of service provisioning, management and auditing tasks. This role is further stressed in multiservice network environments following a class-based paradigm, where distinct quality of service (QoS) profiles and service level specifications (SLS) need to be fulfilled. Active monitoring carried out on an edge-to-edge basis, i.e. between the network boundaries in which a service level needs to be enforced, is particularly suitable for QoS control and SLS auditing, providing valuable inputs for ruling the underlying traffic control mechanisms. To assist short and medium term network operation, this monitoring process needs to be performed on-line so that traffic control decisions are not decoupled from the current network status.

Multiclass networks pose additional challenges on on-line active monitoring. As each traffic aggregate receives a distinct treatment either from a node or a domain perspective, probing needs to be carried out in a per-class basis (in-band) so that class measuring requirements and behavior are met and sensed properly. Particularly, in multiclass environments, an efficient on-line monitoring should be an accurate, fast, systematic, low-overhead and low-interfering process. While accurate and fast measurements allow a correct assessment of the current network state providing up-to-date information, systematic measurements allow to auto-correct the measures and capture each traffic class dynamics. The overhead and interference introduced by active monitoring should be minimized to avoid degrading the class QoS, e.g. decreasing the throughput or causing persistent loss or delay. To minimize probing impact on class traffic, a single probing stream should be able to capture multiple QoS metrics of a service class, i.e. it should be *multipurpose*.

In this paper, we explore the use of edge-to-edge active monitoring to effectively control both QoS and SLS parameters in multiclass networks, while reducing the effects of intrusion on real traffic. We focus on finding and tuning multipurpose per class probing patterns so that each class behavior is correctly captured, even if more than a QoS metric is under control. For accomplishing this objective, different spatial-temporal characteristics of probing patterns are studied for the estimation of IPTD, ipdv and IPLR metrics, commonly identified for measuring quality and performance of Internet services.[1−4]

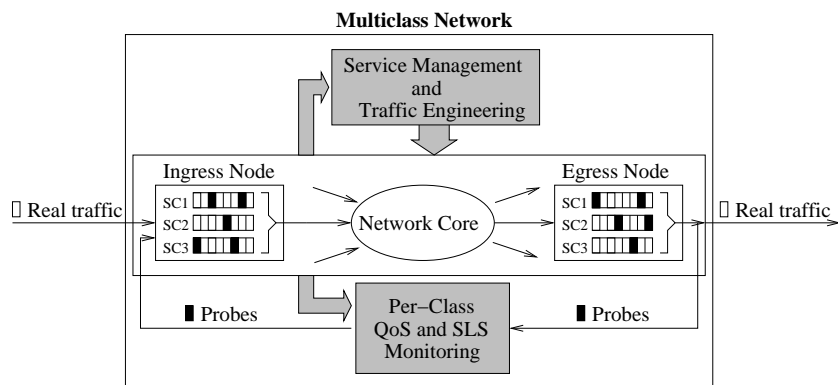E-mail: {solange, pmc, vf}@di.uminho.pt, Telephone: +351253604430, Fax: +351253604471

Figure 1. Multiclass active monitoring

Moreover, we explore a novel approach of coloring probes to improve the active monitoring sensitivity to network congestion and propose a new probing stream able to enhance multipurpose QoS estimation. As a case study, we consider a diffserv domain where a distributed monitoring-based admission control (AC) model[5, 6] is in place to dynamically control traffic entering the domain. The AC model resorts to edge-to-edge on-line monitoring to obtain measures reflecting each class performance so that proper AC decisions can be made. This model is particularly suitable for studying and deploying active monitoring on a per class basis.

The remaining of this document is structured as follows: the motivation of this work facing the related work, the problematic of multiclass QoS monitoring, the identification of relevant QoS metrics and probing pattern characteristics are debated in Section 2; the description of the test environment and the AC model operation and implementation details are summarized in Section 3. Section 4 discusses the tuning of the active monitoring process and corresponding results, while Section 5 includes the main conclusions and contributions of this work.

2. MULTICLASS AND MULTIPURPOSE ACTIVE MONITORING

2.1. Related work

The research community has launched strong efforts in developing and improving measurement methodologies, monitoring systems and tools for the estimation of relevant network performance parameters. As regards the measurement methodologies, they are typically classified as passive and active.

Active measurements resort to intrusive traffic, or probes, specifically injected in the network for measurement purposes. This type of methodology allows to emulate a wide range of measurement scenarios, providing a straightforward and inherent approach of assessing edge-to-edge QoS and performance objectives (see Fig. 1). For instance, as specific packets are injected in the network containing timestamping and sequencing data, delay and loss estimation are simplified. However, as an intrusive process, probing needs to be tightly controlled not to disturb or interfere with the normal network operation*. Relevant projects and monitoring systems which resort to active measurements include NIMI,[8] RIPE NCC Test Traffic Measurements,[9] Surveyor[10] and AMP.[11] Although active monitoring has been matter of considerable research, extending its study to a multiclass network paradigm is crucial and further study is needed.[12–16]

Passive measurements use existing network traffic for metric computation. Particularly suitable for troubleshooting, passive measurements commonly resort to special-purpose devices and built-in mechanisms available in network devices. Monitoring solutions based on SNMP are representative of this type of measurement, and for instance, a diffserv MIB was recently defined including specific objects for the management of differentiated services. Other examples of solutions oriented to passive measurements are tcpdump, Cisco's NetFlow and Service Assurance Agent (SAA), NeTraMet, and DAG-card based systems. In high speed networks, passive measurements are a particular challenge, specially when all packets have to be accounted for, as the amount of data gathered might be substantial and the packet processing time

---

*Appropriate use of topology-based steering approaches and combined passive and active measurement methodologies can reduce the complexity of measuring the connections between edge nodes from $O(n^2)$ to $O(log(N))$.[7]

very small. To deal with this, sampling techniques, more powerful hardware and new packet buffering techniques may be required.[17] Although passive techniques are usually used to monitor the performance of single nodes, they can also be applied to edge-to-edge measurements, combining hop-by-hop metrics along the network path. This allows to reduce the network interference and amount of synthetic traffic[†], at expense of increasing processing and, eventually, synchronization needs. According to,[13] the accuracy tests when measuring one-way delay and loss through active edge-to-edge and hop-by-hop aggregation produced similar results. While this latter approach is claimed to be more scalable (if assessed in terms of intrusive traffic load), the heterogeneity of network nodes and the need for metrics portability may impair deploying large scale passive monitoring solutions. Moreover, obtaining edge-to-edge estimates combining link-by-link measures is not an efficient and easy solution.[18] An alternative edge-to-edge approach still in the scope of passive measurements relies on the analysis of information of real application flows (e.g. using TCP ACK or RTCP data). This approach is also referred as passive probing.[15] To take advantage of the positive aspects of both methodologies, many authors propose the use of integrated measurement environments, where passive and active measurements are combined to achieve more scalable monitoring systems.[7, 13, 14, 17, 19]

As mentioned earlier, active measurement techniques are particularly suitable for edge-to-edge on-line QoS monitoring purposes, providing that the levels of intrusion are kept tightly controlled. In multiclass networks efficient strategies of in-band probing are fundamental in order to sense each class performance without causing noticeable side-effects on the class real traffic. To reduce intrusion, a probing pattern should be multipurpose, capturing simultaneously multiple QoS metrics of a class. Investigating this possibility is the major focus of this work. In this context, relevant questions are: How to extend the monitoring process to a multiclass network environment, capturing the classes' dynamics with minimal overhead? Can a probing pattern capture accurately more than one metric, according to the service class measuring requirements? Is there a trade-off in the simultaneous estimation of multiple QoS metrics?

2.2. Controlled QoS and SLS metrics

Both IETF IPPM and ITU-T have devoted substantial efforts to the definition of relevant QoS metrics and measurement methodologies for IP networks.[4, 20, 21] IPPM aims at developing a set of standard metrics providing unbiased quantitative measures of quality, performance and reliability of operational Internet data delivery services, proposing also measurement methodologies for the defined metrics,[1-3, 20] and some ongoing projects and operational environments use IPPM outcome (e.g. Surveyor[10] and RIPE[9]). Although using different terminology, ITU-T work is fairly consistent in defining the relevant metrics to be considered, and identifies some worst-case QoS upper bounds for common services and applications.[22] As regards SLS metrics, the available bandwidth, one-way packet loss, one-way loss pattern, IP packet delay variation and one-way delay, are the most relevant SLS metrics.[23] Note that, this order and relevance are rather subjective as they are clearly service dependent.

Taking these works into account, in Table 1 we classify quality, performance and reliability metrics in major groups according to their semantics, i.e. depending on what they intend to measure. It is noticeable that one-way metrics have deserved special attention and preference over two-way metrics. In fact, due to possible asymmetric paths and/or different network resource allocation and queuing behavior in both directions, one-way measurements give more precise information and are, therefore, more useful.

In this paper, we focus on capturing simultaneously the behavior of three one-way metrics - IPTD, ipdv, IPLR - in a dynamically loaded and controlled multiclass environment (see Section 3). These metrics are defined in Table 2 for a measuring time interval Δt_i. The mean value of each metric in Δt_i, measured for an ingress-egress pair (I_n, E_m) and traffic class i, is controlled by the AC module as described in Section 3.

2.3. Probing patterns

The type of probing patterns used for metric estimation varies according to the metrics to be computed and the periodicity required for their evaluation. This variability involves changing both the time and space characteristics of the probing pattern. For instance, for measuring delay and loss related parameters, continuously simple and very low rate probing

---

[†]Actually, gathering and transport of passive measurement data may also interfere normal network operation. To reduce this overhead, event notification and statistics summarization may be used.

Table 1. Quality, performance and reliability metrics

metrics	IPPM	ITU	Other
Delay	One-way delay (OWD) IP packet delay variation (ipdv) Round-trip delay (two-way)	IP packet transfer delay (IPTD) IP packet delay variation (IPDV)	
Rate	Empirical Bulk transfer capacity (framework)	IP packet throughput (IPPT) IP packet octet throughput (IPOT)	Available bandwidth Available capacity Throughput
Loss	One-way packet loss (OWPL) One-way loss pattern (OWLP)	IP packet loss ratio (IPLR)	
Other		IP packet error ratio (IPER) Spurious IP packet ratio (SPR)	ECN marks MTU size
Avail. and Reliab.	Connectivity (one or two-way)	% IP serv. unavailability (PIU) % IP serv. availability (PIA)	Mean downtime Mean time to repair Mean time between failures

Table 2. Controlled QoS metrics

IP Transfer Delay ($IPTD$)	$IPTD_{i,pkt} = t_{E_m,pkt} - t_{I_n,pkt}$
Mean IPTD ($\overline{IPTD}$)	$\overline{IPTD}_{i,\Delta t_i} = (\sum IPTD_{i,pkt} / \sum pkts\_recv_i)_{\Delta t_i}$
IP Packet Delay Var. ($ipdv$)	$ipdv_{i,2pkt} = IPTD_{i,pkt} - IPTD_{i,pkt-1}$
Mean ipdv ($\overline{ipdv}$)	$\overline{ipdv}_{i,\Delta t_i} = (\sum |ipdv_{i,2pkt}| / \sum pkts\_recv_i)_{\Delta t_i}$
IP Loss Ratio ($IPLR$)	$IPLR_{i,tot} = tot\_pkts\_lost_i / tot\_pkts\_sent_i$
Mean IPLR ($\overline{IPLR}$)	$\overline{IPLR}_{i,\Delta t_i} = (\sum pkts\_lost_i / \sum pkts\_sent_i)_{\Delta t_i}$

patterns have been in use in real and experimental network environments (e.g. 2 pps[9] or 4 pps[10]). For bandwidth estimation[‡], several techniques, such as Variable Packet Size (VPS), Packet Train Dispersion (PTD) and Self Loading Periodic Streams (SLOPS) have been proposed.[24, 25] Common measurement tools based on these techniques,[23, 26–28] generically use high volume or rate probing (e.g. 100-5000 pkts) and require a significant amount of time to obtain a singleton measure (e.g. 40-100 RTTs). This slow process impairs its use for continuous estimates due to the underlying overhead, being more appropriate for sparse measurements. For continuous measurements, common methods for collecting sample metrics use either periodic or random sampling. In *periodic sampling* measurements are made evenly spaced in time. Although being attractive for its simplicity, its eventual drawback is a possible synchronization with a periodic behavior either of the metric itself or induced by a network component. Moreover, periodic network perturbations resulting from probing itself can drive the network into a synchronization state, which may end up affecting the measure leading to biased metrics. In *random sampling*, such as Poisson or geometric sampling, the samples are taken at independent, randomly generated time intervals according to a statistical distribution. This avoids possible synchronization effects, yielding to unbiased samples.[20] The major drawback of random sampling is the complexity introduced into the frequency-domain analysis. The Poisson distribution, which leads to unpredictable sampling, is commonly used and recommended.[20] Despite having higher predictability, the uniform distribution is used to bound the interval between samples, speeding up the convergence of the estimation resulting from sampling.

Initial results on per class active monitoring[29] suggest that commonly accepted and used probing patterns may fail to capture some of the metrics in Table 2 either in terms of shape or scale. Thus, in addition to periodic (CBR_P) and Poisson (POI_P) probing streams, we tested on-off exponential streams ($EXPOO_P$) and developed and tested a new back-to-back on-off probing stream ($B2B_P$) with a deterministic on period (that determines the number of back-to-back packets or train length used) and a fixed or uniformly distributed off period. These streams aim at increasing probing sensitivity to queue variations by reducing the interpacket time gap between consecutive probes, while remaining simple and light. Once again, as probing needs to be carried out on a systematic way, overhead has to be tightly controlled to keep any solution deployable in real networks. Moreover, we have also explored the effect of marking probes with different drop precedences, in single

---

[‡]The bandwidth estimation concept involves the estimation of available capacity and available bandwidth. While the former determines the upper bound of the available bandwidth in the path, the latter determines the maximum throughput under current load.

color schemes and interleaved color schemes. This new approach aims at exploring Active Queue Management (AQM) mechanisms actions in case of queue congestion and the different probabilities of packets reaching the network boundary, which may be particularly useful to sense packet loss in the network domain. The study of probing pattern characteristics and its accuracy for multiple metrics estimation in a multiclass test platform is developed in Section 4.

3. TEST ENVIRONMENT: A DIFFSERV DOMAIN WITH MONITORING-BASED AC

To access the effectiveness and accuracy of multipurpose probing patterns, a diffserv domain with monitoring-based admission control (AC) is used as test environment. In brief, the AC model proposed in[5, 6] resorts to edge-to-edge on-line monitoring to obtain feedback of each class behavior and performance so that proper AC decisions can be made. To dynamically control traffic entering a network domain, the model underlying AC rules control both QoS levels in the domain and the usage of active SLSs between domains. Apart from the common tasks usually present in class-based networks, ingress routers perform explicit or implicit AC depending on the application type and corresponding traffic class and egress routers perform on-line QoS monitoring and SLS control. *On-line QoS Monitoring*, carried out on an ingress-egress basis, measures specific metrics for each service type. The obtained measures reflect a quantitative view of the service level available from each ingress node. *SLS Control* monitors the usage of downstream SLSs at each egress node so that traffic to other domains does not exceed the negotiated profiles. QoS monitoring statistics, SLS utilization and associated parameters are then sent to the corresponding ingress routers to update an Ingress-Egress service matrix, which is used for distributed AC and active service management. This notification is carried out periodically, when a metric value or its variation exceeds a limit or the SLS utilization exceeds a safety threshold.

Assuming $I = \{I_1, I_2, ..., I_N\}$ as a set of N ingress nodes and $E = \{E_1, E_2, ..., E_M\}$ as a set of M egress nodes in domain D, the admission criterion resorts to (i) rate-based SLS control rules and (ii) class QoS control rules. More precisely, for each ingress node $I_r \in I$ with $1 \leq r \leq N$ and class i, a new flow is admitted if

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

　　where ρ_s is the current measured load or estimated rate of flows using SLS_s, r_j is the rate specified for flow j, $0 < \beta_s \leq 1$ is the utilization target for the SLS and R_s is the rate defined in SLS_s; and

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

　　i.e. the controlled parameters $P_i = \{p_1, p_2, ..., p_p\}$ of class i checked against the corresponding pre-defined threshold t_p determine an acceptance status for Δt_i, which remains unchanged during this interval. Each t_p is affected by a safety margin to the parameter bound $0 < \beta_p \leq 1$ as in Eq.(4). $\tilde{p}_p$ represents the measured value of p_p for Δt_i.

More details on the AC model including its end-to-end operation are given in.[5, 6]

3.1. Implementing and configuring the test platform

A diffserv domain with AC based on the model described above was developed and implemented in the Network Simulator (NS-2) according to the topology illustrated in Fig. 2. In this test simulation prototype, following current IETF service class guidelines,[30] three fundamental service classes were defined. SC1, oriented to conversational services, provides a high quality service guarantee and is supported by EF PHB. This class may comprise traffic with hard real-time constraints such as VoIP, circuit emulation over IP or conversational UMTS traffic. SC2, oriented to a range of streaming applications with soft real-time constraints, provides a predictive service with low delay, low loss and minimum bandwidth guarantee and is supported by AF PHB. This class may comprise broadcast TV, audio and video streaming, webcasting or UMTS streaming traffic (from 3GPP applications). SC3, oriented to elastic applications, generically, supports TCP adaptive traffic. Depending on the nature of TCP flows (e.g. high throughput vs. undifferentiated traffic), this class can be implemented using a AF or DF PHB. There is also the possibility of injecting concurrent traffic (CT-I2) of any of the above classes, allowing to test the effect of cross-traffic on probing.

SC1, SC2 and SC3 traffic is generated resorting to the source models specified and parameterized according to Table 3. While SC1 comprises UDP traffic sources with low rates (8kbps) and small packet sizes (120bytes), reflecting voice-like traffic, SC2 comprises also UDP traffic with higher peak rates (256kbps) and larger packet sizes (512bytes). SC3 comprises long-lived high throughput TCP traffic, from FTP protocol. The flow arrival process is Poisson with exponentially
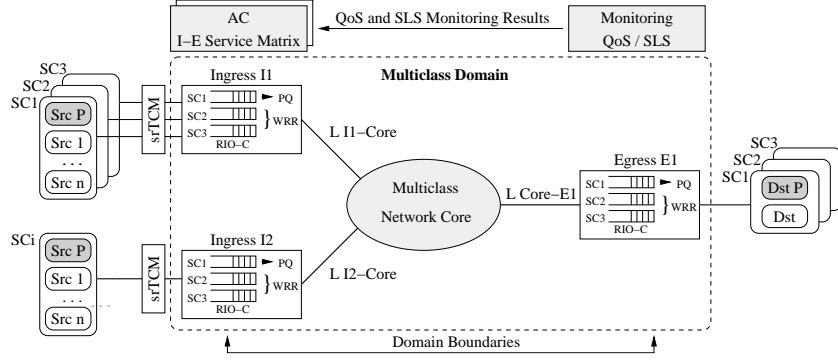
Figure 2. Network simulation topology

Table 3. Traffic source configuration

Class	Protocol	Src Type	Src Parameters	Inter. t	Hold. t
SC1	UDP	$EXPOO_{SC1}$	(r=64kbps, l=120B, on=0.96ms,off=1.69ms)	0.4-2s	90s
SC2	UDP	$EXPOO_{SC2}$	(r=256kbps, l=512B, on=off=500ms)	0.4-2s	120s
SC3	TCP	FTP App.	(r=unspecified,l=512B)	0.4-2s	180s
Probing	UDP	CBR_P	(r=varied, l=100B)	1 src	sim. dur.
	UDP	POI_P	(r=varied, l=100B, poisson)	1 src	sim. dur.
	UDP	$EXPOO_P$	(r=varied, l=100B, on/off=exponential)	1 src	sim. dur.
	UDP	$B2B_P$	(r=varied, l=100B, on=deterministic/off=uniform)	1 src	sim. dur.
	TCP	FTP_P	(r=variable, l=100B)	1 src	sim. dur.

distributed interarrival and holding times. The type of concurrent traffic injected at Ingress I2 is mapped to SC2. For each (I_n, E_m) pair, a single probing source is embedded in each service class. The probing sources considered and their generic characteristics are indicated in Table 3. As mentioned in Section 2.3, five types of probing sources were considered: periodic (CBR_P); Poisson (POI_P); exponential on-off ($EXPOO_P$); back-to-back hybrid on-off ($B2B_P$) and a TCP based (FTP_P). Table 5 specifies the particular probing characteristics used in each of the test scenarios.

The domain routers implement the service classes defined above according to a hybrid Priority Queuing - Weighted Round Robin (PQ-WRR(2,1)) scheduling discipline, with RIO-C for AQM. Each class queue is 150 packets long. The domain internodal link capacity is 34Mbps, with a 15ms propagation delay. At network entrance, SC1 is policed and marked using a Token Bucket (TB) which controls both rate and burst size, whereas SC2 and SC3 are policed and marked using a single-rate Three Color Market (srTCM). The access links to domain boundaries are configured so that intradomain measurements are not affected.

The service-dependent AC rules are parameterized as specified in Table 4[§]. As initial configuration, we have considered three downstream SLS, one per service class. The choice of SLS rate shares (R_s), safety margins (β_s) and QoS parameters thresholds (t_p) are defined in the right hand side of Table 4. As shown, larger β_s and tighter t_p are defined for more demanding classes. The AC thresholds are set taking into account the domain topology dimensioning, queuing and propagation delays and perceived QoS upper bounds for common applications and services.[22, 23, 31]

4. TESTING MULTIPURPOSE PROBING PATTERNS

The main objective of our tests is to determine and tune adequate multipurpose probing streams for the defined service classes. Ideally, probing should be able to reflect both the shape and the scale of the relevant metrics identified for each class so that its behavior is correctly captured. To assess the suitability and effectiveness of multipurpose in-band probing patterns in measuring the QoS parameters defined in Table 4, we have explored alternative probing schemes varying space and time pattern properties such as probing distribution, rate, rejection precedence (color) and packet size, in a per-class

---

[§]This configuration aims at allowing the occurrence of possible QoS degradation or service disruption in order to verify if probing is able to detect it.

Table 4. SLS and QoS control

SLS Control Rule:			$\rho_s + r_j \leq \beta_s R_s$	
Class	Monitoring Inputs ρ_s	Flow Inputs r_j	SLS Rate R_s (% Class share)	Safety Margin β_s
SC1	Traffic load	peak rate	3.4Mbps (10%)	0.85
SC2	Traffic load	mean rate	17.0Mbps (50%)	0.95
SC3	Traffic load	n.a.	13.6Mbps (40%)	1.0
QoS Control Rule:			$\forall p \in P_i : \tilde{p}_p \leq t_p$	
Class	Monitoring Inputs $\tilde{p}_p$	Flow Inputs	QoS Param. Thresh. t_p	
SC1	IPTD, ipdv, IPLR	if available	$35ms; 1ms; 10^{-4}$	
SC2	IPTD, IPLR	if available	$50ms; n.a.; 10^{-3}$	
SC3	IPLR	n.a.	$n.a.; n.a.; 10^{-1}$	

perspective. In the analysis of the measurement results, the probing measurement outcome is cross-checked against the corresponding measures using the real traffic in each class, i.e. active and passive measurement results are compared. This verification process is carried out resorting to a direct comparison of graphical results (pictorial proof) and statistical analysis of collected measurement samples, for the different QoS parameters and service classes.

4.1. Testing scenarios

Table 5 summarizes the planned test scenarios exploring different characteristics of probing patterns. For instance, while Test 1,2 and 6 study the impact of probing rate, color and packet size on multiple metrics estimation, respectively, Test 3 and 5 are devised to enhance probing ability to overcome identified limitations when estimating multiple metrics.[29] Apart from Test4, which uses a TCP probing stream in SC3, the remaining testing scenarios use UDP probing streams in all classes. The following sections will detail each of these tests.

**Table 5.** Test scenarios taxonomy

Test Scenario	Tested Variable	Probing pattern characteristics				I-E Total Overhead (%C)	M.Int. Δt_i (s)
		Distribution	Mean rate (pps)	Color	PktSize (B)		
Test1	Rate	POI_P	2 to 192	green	100	0.01 to 1.4	5
Test2	Color	POI_P	2 to 192	red	100	0.01 to 1.4	5
Test3	IPTD vs IPLR	POI_P	2 to 192	interleaved	100	0.01 to 1.4	5
Test4	TCP Probing	FTP_P	n.a.	green,red,inter.	100	varies	5
Test5	ipdv	$B2B_P$	8 and 24	interleaved	100	0.006 and 0.17	5
Test6	Probe size	POI_P	4	red	100 and 512	0.03 and 0.15	5

## 4.2. Estimation of IPTD, ipdv and IPLR

Random sampling has been recommended as a convenient technique for systematic measurements of one-way delay, jitter and loss related metrics.[1–3, 20] Based on these directives, some monitoring systems[9, 10] resort to Poisson traffic with light probing rates, e.g. 2 or 4 pps, aiming at obtaining multiple QoS metrics estimation. In a multiclass diffserv domain,[29, 32] these Poisson probing patterns were used in-band (per class) and marked with high precedence (green) to reduce the chance of losing probes. The initial results on measurements accuracy show that while IPTD metrics can be closely captured (shape and scale) for these probing rates, ipdv and IPLR are not measured properly. In particular, probing clearly overestimates ipdv and misses most of IPLR events, unless heavy loss occurs. These results suggest the use of alternative probing patterns[¶] with smaller probing gaps to better sense ipdv[‖] and different coloring schemes in order to increase probing sensitivity to network congestion and loss events.

[¶]Apart from Poisson both periodic and exponential on-off patterns were also tested. We have noticed that, for the probing rates under test, the captured metrics' behavior and their trends are equivalent. However, for similar probing rates and coloring, a periodic pattern leads to a slightly better IPTD and IPLR match than Poisson patterns and high correlation factors between class and probing measurement outcome.

[‖]As ipdv is a consecutive packet measure, probing gaps may lead to higher measures as consequence of queue occupancy variations.
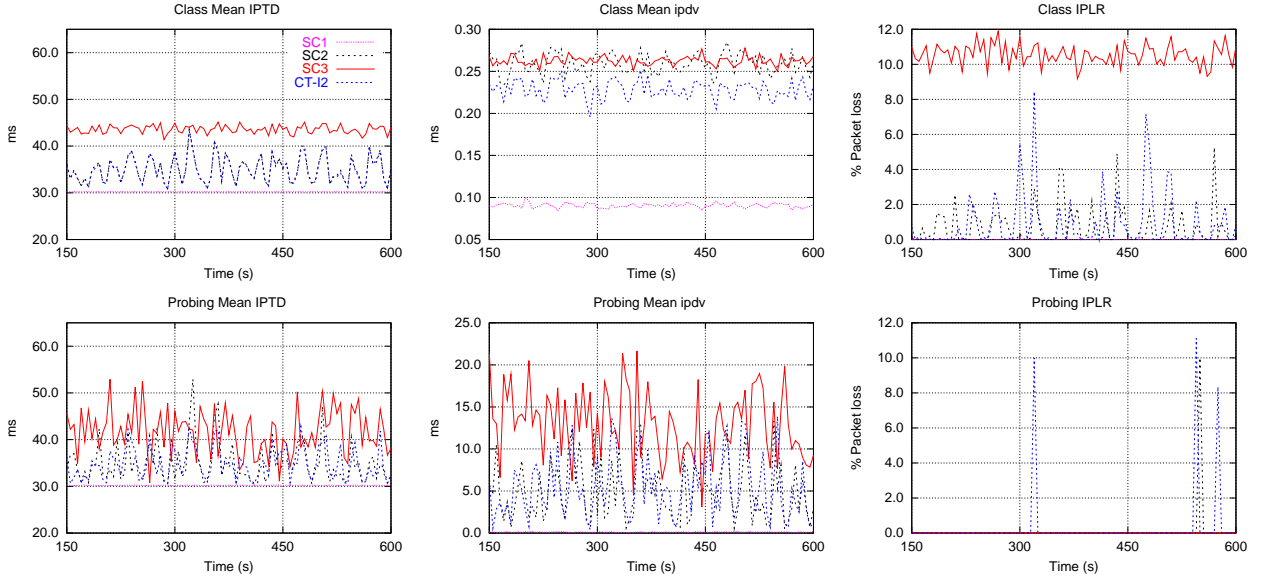
Figure 3. Comparison of class and probing measurement outcome (Test1 @ 2pps)
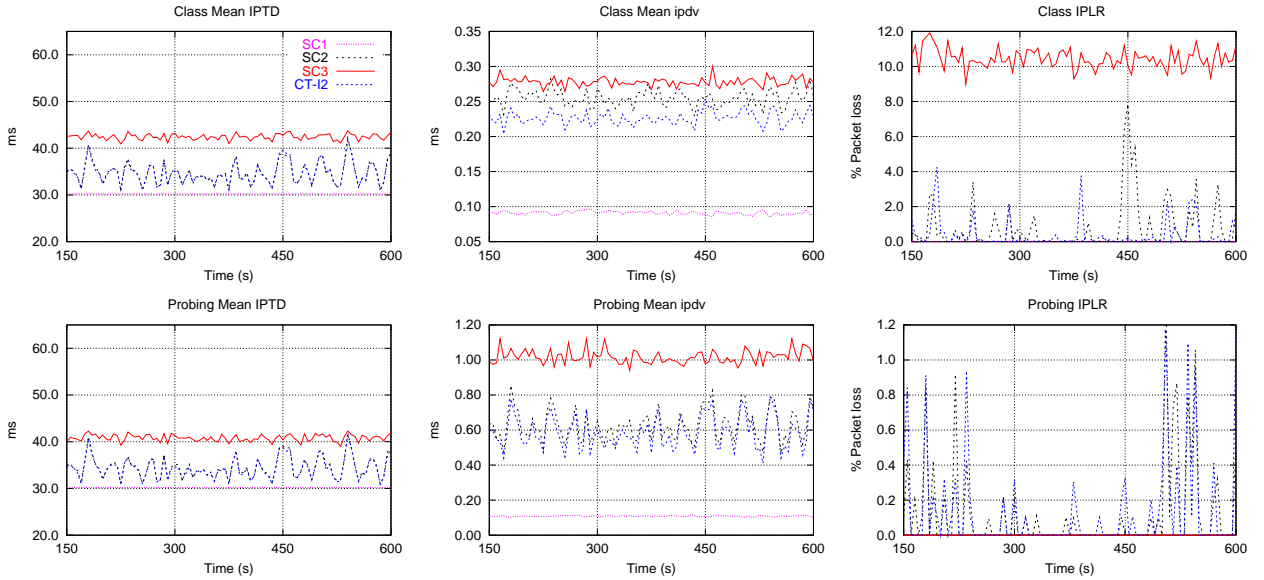
Figure 4. Comparison of class and probing measurement outcome (Test1 @ 192pps)

An immediate and intuitive approach to improve estimations is to increase the sampling frequency without disregarding the overhead introduced. In this way, as specified for Test1 (see Table 5), for each class, we have increased the Poisson probing rate from 2 to 192 pps, setting the measurement time interval to $\Delta t = 5s$. For the number of classes considered, the maximum overhead introduced per ingress-egress pair ranges approximately from 0.01 to 1.4% of the bottleneck link capacity (34Mbps), i.e. the probing rate of each class varies from 1.6 to 153.6kbps. Note that, for classes with low bandwidth share and high priority treatment such as SC1, these values might be relevant whether considering multiple ingress-egress pairs. The active measurement results revealed that the captured behavior is distinct for the classes SC1, SC2 and SC3 under test, and the improvement observed in QoS metric estimation resulting from a probing rate increase also differs among the classes (see Figs. 3 and 4).

SC1 is a very stable traffic class with IPTD and ipdv tightly controlled without suffering any loss, thus, probing is

able to sense this behavior even for a probing rate of 2pps. For this rate, IPTD in SC2 is also well captured, however, a significant decrease on ipdv overestimation (to less than one order of magnitude) is only achieved for the higher probing rates tested (see example in Fig. 6(d)). IPLR also benefits from probing rate increase as loss is gradually better detected, despite a deficit on IPLR scale estimation of approximately one order of magnitude. For the lower probing rates, the behavior of SC3 is clearly the worst captured as regards the three metrics under control. While IPTD and ipdv benefit from a probing rate increase, IPLR events are completely missed even for 192pps.

In summary, Test1 allows to conclude that increasing probing rate, per si, is not enough to overcome QoS metric estimation mismatches. Despite the tendency to improve IPTD and ipdv accuracy, the overhead introduced and the failure of IPLR estimation suggest exploring other probing characteristics either as alternative or complementary approaches.

4.2.1. Exploring probe coloring

Probe coloring is proposed and studied here as a new method of improving IPLR estimation. For the conditions stated in Test2, when changing the precedence of probing packets from high to low, i.e. coloring probes from green to red, a significant improvement in IPLR sensitivity was noticed even for a probing rate of 2pps. This improvement is reflected in a better detection of class loss events, however, IPLR scale is clearly overestimated and IPTD estimation slightly degraded (see Fig. 5). This behavior is justified by the AQM action on probes precedence. When the queue congestion increases, red packets are the first to be dropped; while this increases sensitivity of probes to loss events, previously high delayed green probe packets are now mostly discarded. As a consequence, on average, IPTD estimation is less accurate, while loss detection is improved. The scale magnifying error in IPLR estimation is justified by the huge difference between the total number of probes and class packets in Δt_i. Increasing the rate of red probes revealed fruitful for SC3 IPLR scale accuracy, but for SC2 the previous scale problems remain evident without any noticeable improvement. This is due to the different nature of loss in SC2 (more sparse) and SC3 (more regular), and the volume of red packets in the classes. The side-effects on IPTD estimation were not completely removed either, when higher rates of red probes are used.

In order to improve the compromise between IPLR and IPTD estimation and to adjust IPLR scale, we have tested probing patterns with an interleaved coloring scheme, i.e. green and red packets are alternately sent according to a pre-defined distribution (see Test3 details). Comparing to Test1 outcome, a interleaved probing pattern of 2pps brings significant improvement on IPLR estimation for all traffic classes. Moreover, the degradation of IPTD estimation and IPLR overestimation noticed in Test2 is now much less pronounced. These positive results are further enhanced for higher probing rates where the accuracy of all metrics is increased, as exemplified in Fig. 6(a) and (b) for a probing rate of 24pps. Despite that, IPLR estimation of SC2 still remains overestimated even for 192 pps (see Fig. 6(c)). As in Test1, ipdv estimates are acceptable for probing rates above 96pps (see e.g. in Fig. 6(d) for 192pps). Generically, SC3 is the class which more strongly benefits from interleaved probing, reaching a good metrics estimation compromise for a probing rate of 24pps.

Test2 and Test3 allows to conclude that coloring probes can be particularly useful to provide a valuable qualitative indicator of network congestion (through loss detection) and feedback for traffic control mechanisms, for very low probing rates. However, for SLS auditing and TC mechanisms requiring accurate quantitative inputs, higher colored probing rates are clearly needed.

As SC3 is oriented to TCP traffic, we have also extended Test3 so that an FTP probing source is now considered in this class (Test4), approximating the probing to the class traffic characteristics. The results show that the color on this type of
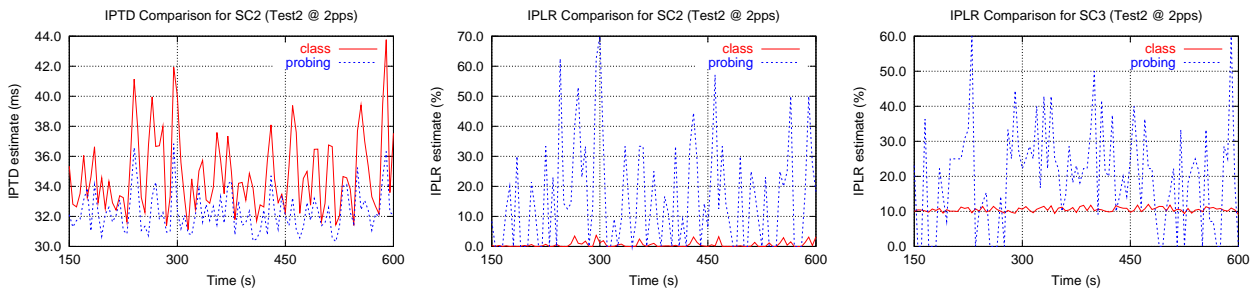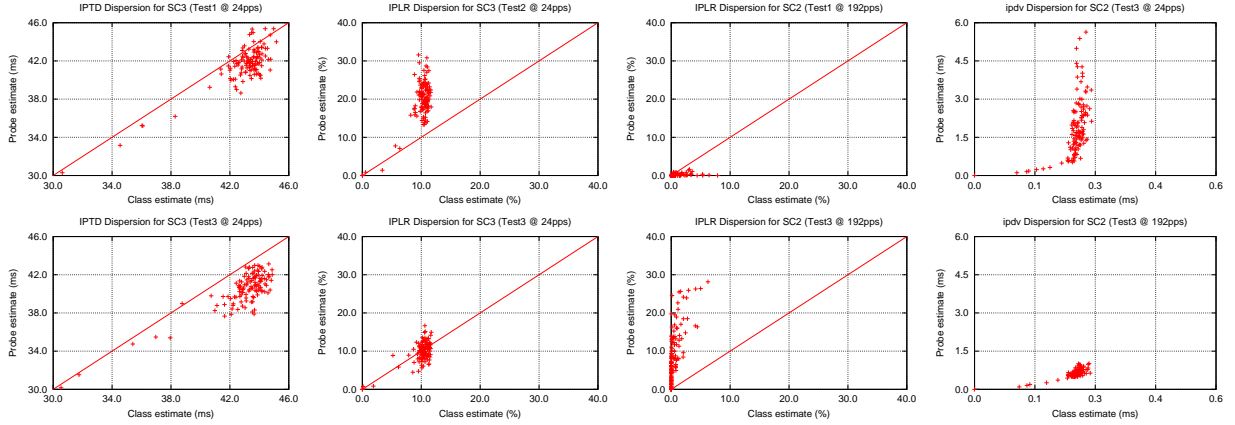

Figure 5. Comparison of class and probing measurement outcome for SC2 and SC3 (Test2 @ 2pps)

Figure 6. Dispersion of (a) IPTD for SC3 - green and interleaved @ 24pps; (b) IPLR for SC3 - red and interleaved @ 24pps; (c) IPLR for SC2 - green and interleaved @ 192pps; (d) ipdv for SC2 - interleaved @ 24pps and 192pps

probing stream is even more relevant than on UDP probing (Test1,2,3). The color is relevant both for the estimate accuracy and for the probing overhead. For instance, green FTP probes still do not detect loss in SC3 and the probing rate may reach a worrying 2Mbps rate, i.e. 15% of the class bandwidth share. This might occur when the probing source does not experience any loss and keeps taking over bandwidth, either because other TCP flows adapt or terminate. Red FTP probes are far less bandwidth demanding (30kbps) with the cost of emphasizing the problems identified in Test2. As for Test3, an interleaved color FTP probing pattern leads to a better compromise between overhead (0.4%) and accuracy.

4.2.2. Improving ipdv estimation

Generically, we have found that ipdv is a rather sensible metric to the network load and its variability. The results illustrated in Test1 and Test3 show that ipdv scale is difficult to obtain regardless the test probing rate. Small class ipdv variations are magnified by probing and ipdv mismatch under the different probing rates suggests that queuing delay oscillations may still persist across multiple time scales. For moderate loads (under 70%), ipdv is more closely measured as the queues remain in a reasonable steady state.[29]

As decreasing the interpacket gaps through a probing rate increase did not successfully resolve ipdv scale estimation, we have considered a new probing pattern which on average tries to keep regular and light the mean probing rate while reducing interpacket gaps. The new probing pattern ($B2B_P$) acts as a packet train regulated by an on-off source, where the number of packets in a train, and the on and off sojourn times are configurable. The rate and duration of the on period determines the amount and how close the probes are; the off period can be either deterministic or regulated by a probability density function such as Exponential, Pareto or Uniform. The use of a non-deterministic off period avoids possible network synchronization effects. In Test5, we have considered interleaved $B2B_P$ probing sources generating 8 and 24pps, respectively, with on and off periods of 125ms, i.e. corresponding to four back-to-back bursts. We have noticed that, when compared POI_P with similar rates, interleaved $B2B_P$ probing streams lead to better estimates of ipdv scale, with also better results on IPTD and IPLR. This improvement is more notorious in the experiment with probing at 24pps as shown in Fig. 7(a), where the correlation coefficient between the class and probing outcome for all QoS metrics is evaluated for all the test scenarios previously considered.

4.2.3. Exploring probing packet size

Intuitively, small size probing packets do not necessarily experience loss when large packets do. In fact, near queue overflow, a small packet is more likely to be enqueued than a larger one. So, apparently, probing packet size should be similar to the mean packet size of the class it intends to measure. In test scenario Test6, we have explored if the probing packet size can influence IPLR estimation. Fig. 7(b) shows the absolute IPLR estimation error ($\epsilon_{IPLR,i} = |IPLR_i^{real} - IPLR_i^{estimated}|$) for SC3 and for a similar number of red probing packets of 100 and 512 bytes (the same as in the class). As the results show, the histograms for both packet sizes are mostly overlapped suggesting minor differences in the observed error. Although we have not found evident impact of packet size on IPLR results, a mixed combination of probing packet sizes is sometimes used.[10]
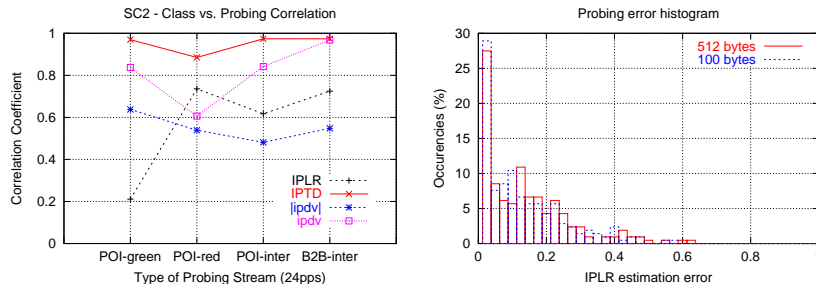
Figure 7. (a) Correlation between class and probing estimates for different probing streams (for SC2) (b) IPLR error estimation for SC3

4.2.4. Additional considerations

Depending on the overall purpose of QoS and performance metrics, and traffic control mechanisms, which may react based on them, a possible solution to overcome excessive sensitivity to fluctuations and mis-scaling of the metrics' behavior is to increase the time interval Δt in which measurements are carried out. A wider Δt will allow a more stable measurement interval as traffic variability is smoothed and more probing packets are taken into account in a single measure. However, to reach accurate measurements the number of probes considered is more relevant than extending the measurement interval,[16] and more probes allow to achieve accuracy earlier. In our case, exploring the impact of Δt on multipurpose measures accuracy was left for further study. Nevertheless, as regards the edge-to-edge monitoring-based admission control mechanism in use, obtaining measurements inputs each 5s revealed to be a good compromise between having an updated vision of network dynamics, the efficiency of AC decisions and the measurement overhead.

In the context of active measurement, although the use of Poisson probing in real environments is recommended[20] (see also Section 2.3), it may constrain the Δt in use. In fact, for small Δt values and low probing rates (e.g. $\Delta t = 5s$; 2 pps), there might be intervals where no probe packets are generated due to the unbounded nature of exponential interarrivals. In our view, the use of a probing on-off source (such as $B2B_P$) with a deterministic on period and a bounded random off period will allow to take advantage of a pre-defined probing burst while preventing possible synchronization effects.

5. CONCLUSIONS

In this paper, the study of edge-to-edge active monitoring has been extended to multiclass IP networks. The main focus has been on finding light and multipurpose probing patterns able to capture each class behavior, while reducing network overhead and intrusion side-effects. Taking a diffserv domain with monitoring-based admission control as a case study, we have explored multiple aspects of probing patterns such as probing distribution, rate, precedence (color) and packet size, when estimating simultaneously one-way delay, jitter and loss related metrics. Comparing the probing and real traffic measurement outcome, our results show that, generically, a probing rate increase leads to an improvement in the accuracy of QoS metrics estimation. This improvement is, however, rather dependent on each class characteristics and on the metrics being estimated. Furthermore, the trade-off between the estimation accuracy and probing overhead may be prohibitive, and even high probing rates have been unable to match both the scale and shape of metrics such as IPLR. This has motivated exploring alternative or complementary probing features capable of increasing multipurpose active monitoring efficiency. From the tested probing characteristics, we have found that the dropping action of active queue management mechanisms on colored probes can significantly influence the probing measurement outcome. This has proved to be particularly relevant in service class oriented to elastic traffic (SC3), where simultaneous estimation of IPTD and IPLR could only be achieved with an interleaved colored probing stream. As regards the probing distribution, we have proposed the use of a hybrid back-to-back on-off source that allows to generate more flexible probing streams, with controlled number of back-to-back probes and probing gaps. The resulting probing streams, especially conceived to improve ipdv estimation, when complemented by a proper coloring scheme can also increase the estimation of the other metrics. This allows to minimize or overcome the limitations exhibited by common probing streams when estimating multiple QoS metrics simultaneously.

REFERENCES

1. G. Almes, S. Kalindindi, and M. Zekauskas, "A One Way Delay Metric for IPPM." IETF RFC2679, 1999.

2. G. Almes, S. Kalindindi, and M. Zekauskas, "A Packet Loss Metric for IPPM." IETF RFC2680, 1999.

3. C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IPPM." IETF RFC3393, 2002.

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

5. S. Lima, P. Carvalho, A. Santos, and V. Freitas, "A Distributed Admission Control Model for CoS Networks using QoS and SLS Monitoring," in *IEEE International Conference on Communications - ICC'03*, May 2003.

6. S. Lima, P. Carvalho, and V. Freitas, "Distributed Admission Control for QoS and SLS Management," *Journal of Network and Systems Management - Special Issue on Distributed Management* **12**, Sept. 2004.

7. B. Lowekamp, "Combining Active and Passive Network Measurements to Build Scalable Monitoring Systems on the Grid," *ACM Performance Evaluation Review* **30**(4), pp. 19–26, 2003.

8. V. Paxson, A. Adams, and M. Mathis, "Experiences with NIMI," in *PAM'00, Hamilton, New Zeland*, Apr. 2000.

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

10. S. Kalidindi and M. Zekauskas, "Surveyor: An Infrastructure for Internet Performance Measurements," in *INET'99, San Jose, CA, USA*, June 1999.

11. NLANR, "Active Measurement Project (AMP)." http://watt.nlanr.net, Aug. 2001.

12. R. Whitner, G. Pollock, and C. Cook, "On Active Measurements in QoS-Enabled IP Networks," in *PAM'02, Fort Collins CO*, Mar. 2002.

13. H. Asgari, P. Trimintzios, M. Irons, R. Egan, and G. Pavlou, "Building Quality-of-Service Monitoring Systems for Traffic Engineering and Service Management," *Journal of Network and Systems Management* **11**, pp. 399–426, Dec. 2003.

14. F. Strohmeier, H. Dorken, and B. Hechenleitner, "Aquila Distributed QoS Measurement." Aquila Project, Aug. 2001.

15. J. Corral, G. Texier, and L. Toutain, "End-to-end Active Measurement Architecture in IP Networks (SATURNE)," in *PAM'03*, 2003.

16. J. Hill, "Assessing the Accuracy of Active Probes for Determining Network Delay, Jitter and Loss," Master's thesis, MSc in High Performance Computing, University of Edinburgh, 2002.

17. J. Coppens, S. Berghe, H. Bos, E. Markatos, F. Turck, A. Oslebo, and S. Ubik, "SCAMPI: A Scalable and Programmable Architecture for Monitoring Gigabit Networks," in *6th IFIP/IEEE International Conference on Management of Multimedia Networks and Services - MMNS'03*, Sept. 2003.

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

19. T. Lindh, "A New Approach to Performance Monitoring in IP Networks - Combining Active and Passive Methods," in *PAM'02, Fort Collins CO*, Mar. 2002.

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

21. I. IPPM-WG, "IP Performance Measurements Working Group." http:/www.ietf.org/html.charters/ippm-charter.html.

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

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

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

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

26. CAIDA Tools, 2002.

27. C. Dovrolis and M. Jain, "Pathload: A Measurement Tool for End-to-End Available Bandwidth," in *PAM'02, Fort Collins CO*, Mar. 2002.

28. V.Reijs, "Tools for measuring the SLS metric," July 2002.

29. S. Lima, P. Carvalho, and V. Freitas, "Tuning Active Monitoring in Multiservice IP Networks," in *Performance Modelling and Evaluation of Heterogeneous Networks - HET-Nets'04*, Networks UK, July 2004.

30. F. Baker, J. Babiarz, and K. Chan, "Configuration Guidelines for DiffServ Service Classes." draft-baker-diffserv-basic-classes-02.txt, Feb. 2004.

31. V.Reijs, "Perceived quantitative quality of applications," July 2001.

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