Performance Analysis for an IP Differentiated Services Network

C Chassot[1], F Garcia[1], G Auriol[1], A Lozes[1], E Lochin[2], P Anelli[2]

[1]LAAS/CNRS, 7 avenue du Colonel Roche, 31077 Toulouse cedex 04, France
email: chassot, fgarcia, gauriol, alozes@laas.fr

[2]LIP6, 8 rue du Capitaine Scott, 75015 Paris, France
email: emmanuel.lochin, pascal.anelli@lip6.fr

***Abstract*-Research reported here deals with a communication architecture with guaranteed end-to-end quality of service (QoS) in an IPv6 environment providing differentiated services within a single DiffServ domain. The article successively presents the design principles and services of the proposed architecture, their implementation over a national platform, and experimental measurements evaluating the QoS provided at the user level.**

I. INTRODUCTION

Technical revolutions in telecommunications and computer science have led to the development of new types of distributed applications such as multimedia and co-operative ones or interactive simulation. These applications present challenging characteristics to network designers, such as higher bandwidths, the need for bounded delays, etc. Quality of Service (QoS) describes the assurance of data transfer that fits to the application requirements. This area is one of the main topics of research and development in data networks. In the Internet community, two main efforts (IETF[1] IntServ [1] and DiffServ [2] working groups) have been carried out in order to develop a QoS framework for the TCP/IP protocol suite. Several research projects have also been initiated to target the QoS problem; let's cite the TF-TANT activity [3] and the GEANT, TEQUILA, CADENUS, AQUILA and GCAP projects [4, 5, 6, 7, 8], implementing a differentiated services architecture following more or less the framework given in [9].

Performed within the national French project @IRS[2], work presented in this paper deals with the conception, the implementation and the performance analysis of a communication architecture providing a guaranteed end-to-end QoS in an IPv6 environment constituting a single DiffServ domain. This work provides the following contributions: it proposes an architecture with well-defined services; it validates the service model by an implementation of the architecture over a national ATM network infrastructure named RENATER2; and it evaluates the QoS provided at the user level by mean of experimental measurements.

This article is structured as follows. Section II presents the architecture principles and services, then it describes the experimental platform over which it has been developed. Section III details the experimental scenarios for the study of the end to end QoS; results of the measurements are also provided and analyzed. Conclusions and future work are presented in Section IV.

II. ARCHITECTURE PRINCIPLES, SERVICES AND IMPLEMENTATION

The following two major sections successively present the architecture defined at the end-to-end level and at the network level.

A. *End-to-end level*

The basic underlying principle that supports the proposal of the @IRS end-to-end architecture is one of many dedicated to the transport of multimedia flows [10, 11, 12]. The idea is that the traffic exchanged by a distributed application can be decomposed into several data flows, each one requiring its own specific QoS. That is, an application can request a specific QoS for each of its flow via a consistent API (*Application Programming Interface*). By way of a session (see Fig. 1), the application layer software is then allowed to establish one or many end-to-end communication channels, each one being (1) unicast or multicast, (2) dedicated to the transfer of a single flow, and (3) able to offer a specific QoS.

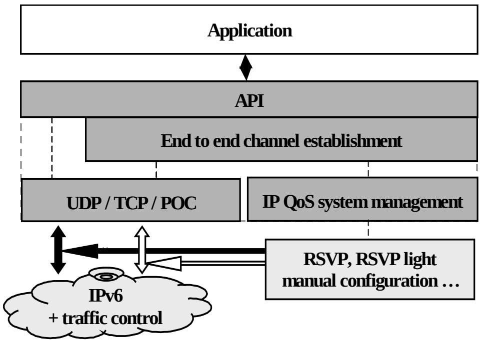

Fig. 1. Architecture of the end-to-end com. system

Besides the API, three other conceptual modules are defined. The first one provides multiple transport layer possibilities, such as TCP, UDP, or the partially ordered / partially reliable POC protocol [13,14]. The second one implements the mechanisms linked to the utilization of the QoS services at the IP layer. The third one is in charge of the end-to-end channel set up. Due to space limits, we only present the service parameters of the API. For each channel, QoS is expressed by means of the following parameters:

- a maximum end-to-end transit delay;
- an intra flow partial order[3], expressing logical synchronization constraints (synchronization of two media transported within the same end-to-end channel);

---

[1] IETF: Internet Engineering Task Force

[2] @IRS project: Integrated Networks and Services Architecture (Dec. 1998-April 2001) is a national project of the France's *Réseau National de la Recherche en Télécom.*, whose objective was to experiment innovative Internet mechanisms within an heterogeneous network infrastructure.

[3] The goal of this paper is not to enforce ordering relationships within channels that may lose packets to reduce data transit delay. Identically, usefulness of RTP based propositions is not studied.

– a partial reliability defined, for example, by a max number of consecutive lost packets and/or a max % of lost packets.

Moreover, an inter flow partial order allows the application to express logical synchronization constraints between channels.

In addition, an application must specify four service parameters:
– the first one characterizes the traffic generated by the application sender (for @IRS, a *token bucket*);
– the second one designates which transport protocol to use;
– the third one designates the IP layer's QoS management desired by the application (IntServ or DiffServ oriented);
– the final parameter identifies the address, either unicast or multicast, of a set of destination application software's.

Although the architecture is designed so as to allow several Transport protocols or IP level QoS systems, the one implemented within the project only includes UDP and TCP at the transport level and a DiffServ oriented proposition at the IP level.

B. Network level

QoS management functions performed at the IP level can be divided in two parts: those related to the *control path* (i.e. all that is related to routers configuration so as a required QoS might be enforced), and those related to the *data path* (i.e. data transfer). If studies performed during the @IRS project tackle the two areas, only the data part has been implemented over the platform. In this section, we first describe the services defined at the IP level, then we detail the main functions required for their implementation.

1) *Services*. Three services have been defined at the IP level*:*
– GS (Guaranteed Service - analogous to the *Premium Service* [15]) is used for data flows having strong constraints both in terms of delay and reliability. Applications targeted by GS are those which do not tolerate QoS variation;
– AS (Assured Service) is appropriate for responsive flows having no strong delay constraints, but requiring a minimum average bandwidth. An AS flow has to be provided with an assured bandwidth for the part of its traffic respecting the characterization profile specified for the flow. Part of the traffic exceeding the characterization is conveyed in AS as far as no congestion occurs on the path used by the flow;
– BE: Best Effort service offers no QoS guarantees.

2) *Control path QoS functions*. Mechanisms involved in the control path are admission control, route change protection and multicast management. We only present admission control principles. The admission control takes care of the acceptance of new AS or GS flows. Its decisions are taken according to a traffic contract established between the user and the DiffServ services provider. Our proposition is different for AS and GS. For AS, the control is applied at the edge of the network only; it is based on the amount of AS traffic already authorized to enter the network. This gives guarantee that the amount of in profile packets in the network will be at most the sum of the AS authorized at each edge router. For GS, as a delay guarantee is needed, the admission control involves all the routers on the data path.

3) *Data path QoS functions.* QoS functions involved in the data path are policing, scheduling and congestion control.

Policing. Policing deals with the actions to be taken when out of profile traffic arrives in a given service class. For AS, action is to mark the out of profile packets with a higher drop precedence than for the in profile traffic. Packets marked "OUT" are called *opportunistic* packets because they are processed like the other "IN" packets, as far as no congestion occurs. Targeted applications are those whose traffic is elastic, i.e. with a variable profile (a minimum still being assured). For GS, as a guarantee exists, one must be sure that enough resources are available and the amount of GS traffic in the network must be strictly controlled. The chosen policing is to shape the traffic at the edge router and to drop out of profile GS packets.

Scheduling. Scheduling is different for AS and GS packets. GS scheduling is implemented by a *Priority Queuing* (PQ) mechanism. This choice is due to the fact that a PQ scheduler adds the smallest delay to the packet forwarding. The remaining bandwidth is shared by a *Weighted Fair Queuing* (WFQ) between AS and BE traffic.

Congestion control. The congestion control issue is essential for QoS services, as a congestion can prevent the network from offering the contracted QoS to a flow. GS traffic does not need congestion control as it benefits from a priority queuing ensuring that all its packets are served up to the maximal capacity of a given router. Associated with a drop of out of profile packets at the network boundary, this guarantees that no congestion will occur in routers GS queues. For AS, as opportunistic traffic is authorized to be sent in the network, the amount of AS packets in any router can't be known *a priori*. Therefore, a drop precedence system has been implemented, allowing the drop of opportunistic packets as soon as a congestion is about to occur in an AS queue. A *Partial Buffer Sharing* (PBS) has been chosen on AS queues rather than a *Random Early Discard* (RED) method in order to avoid queue length oscillation problems [16,17].

We now detail the implementation of those functions at routers input and output interfaces.

Input interface of edge router (Fig. 2). This interface is the first encountered when a packet enters the network. It is in charge of:
– classifying packets, by means of *source address* and *flow_id* IPv6 header fields (*Multi-Field Classification*);
– measuring AS/GS flows to determine if they are in profile;
– shaping GS packets and dropping them if necessary;
– marking AS packets either IN or OUT;
– marking packets with the appropriate DSCP (*DiffServ CodePoint*): EF (resp. AF, DE)[1] for GS (resp. AS, BE) flows.

Output interface of all routers (Fig. 3). In the DiffServ model, all routers must implement a set of forwarding behaviors called *Per Hop Behavior*, such as [18,19]. In the @IRS architecture, those behaviors are implemented through scheduling and congestion control at the output interface of each router. The other functions are a *Behavior Aggregate Classifier* which classifies packets

---

[1] EF: Expedited Forwarding, AF: Assured Forwarding, DE: Discard Eligibility

according to their DSCP before scheduling, and a rate control (for core routers only), necessary to avoid congestion at the ATM level.

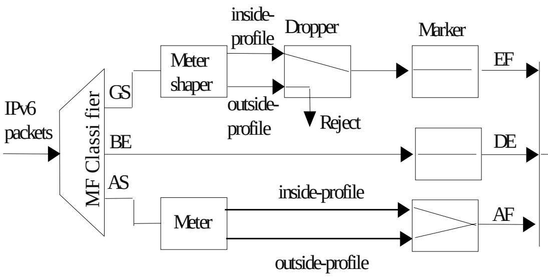

Fig. 2. Input interface structure

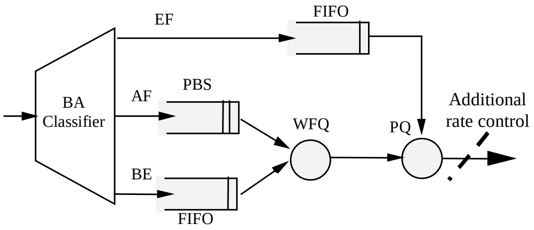

Fig. 3. Output interface structure

III. SCENARIOS AND MEASUREMENTS

A first set of measurements have been realized on the @IRS platform [20]. The major goal was to evaluate the QoS provided to a single UDP flow served in AS (respectively in GS) in presence of a BE traffic whose load was progressively increased until complete overload of the network. Measurement results have allowed to conclude that AS and GS QoSs were conformed to the expected ones, that is: (1) a null impact of BE traffic on the GS QoS: {minimal, maximal and average} values of the transit delay, reliability and average throughput are almost unchanged, and (2) a weak impact on the AS QoS: only the maximal value of the transit delay is increased. Starting from those results, the goal of the experiments presented here is twofold:

- study the impact of the number of AS and/or GS flows on the AS (resp. GS) QoS when the network is overloaded by a BE traffic (the network is in state of congestion);
- discuss the possibility to characterize an AS service for a given configuration of a DiffServ platform like the @IRS one.

We first present the @IRS platform, then the experimental scenarios, and finally the experimental results and their analysis.

C. *@IRS platform configuration*

Measurements have been realized between (LAAS) Toulouse and LIP6 (Paris) over the IPv6 environment illustrated on Fig. 4. Local platforms are connected by edge routers (R_e) to an *Internet Service Provider* (ISP) represented by the national ATM RENATER2 platform. Four core routers (R_c) are introduced within the ISP (physically, they are located in the local platforms).

By means of its edge router, each site is provided with an access point to the ISP, characterized by a traffic contract (the *Service Level Agreement* of [9]). For each service, this SLA consists of several classification and packet (re)marking rules, a traffic profile (the *Traffic Conditioning Agreement* of [9]), and actions to perform when TCA is not respected. It is the edge router's responsibility to implement the SLA as it introduces flows within the ISP. Bandwidth of the link connecting sites to the ISP (via a CBR ATM VP) is such that the maximal throughput provided at the UDP level is 107 Kbytes/s for 1024 bytes length packets. In the following, we use the term *link bandwidth* (LB) to refer to this throughput. Routers are configured with the following hypothesis:

- the maximal amount MA_{GS} (resp. MA_{AS}) of GS (resp. AS) traffic that can be introduced by the edge router (in average) has been fixed to 20 Kbytes/s (resp. 40 Kbytes/s), i.e. about 20% (resp. 40%) of the link bandwidth LB;
- the rate control applied by the core router is 100 Kbytes/s;
- weights associated to the AF and DE packet scheduling within the WFQ mechanism are respectively 0.5 and 0.5.

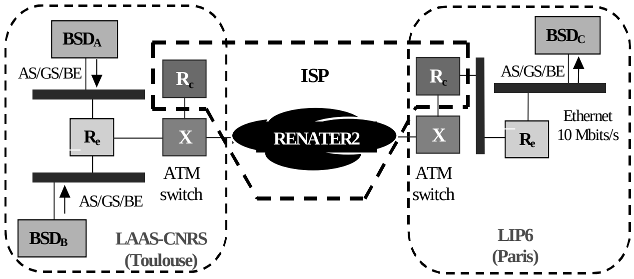

Fig. 4. Platform configuration

D. *Experimental scenarios*

Three scenarios have been designed:

- the first one is aimed at validating the impact of the number of AS flows on the AS QoS, when the network is overloaded; no GS flow is generated;
- the second scenario is aimed at validating the impact of the number of GS flows on the GS QoS, when the network is overloaded; no AS flow is generated;
- the third scenario is aimed at validating the impact of the number of AS (resp. GS) flows on the GS (resp. AS) QoS, when the network is overloaded; here, AS and GS flows are generated together.

For experiment sessions (about 300 seconds), measured parameters are the loss rate and the {min, max, average} values of the transit delay. Distribution of the transit delay is also evaluated.

1) *Scenario 1 (resp. 2).* For those measurements (see Table 1):

- AS (resp. GS) flows are sent from PCs BSD_A and BSD_B to PC BSD_C. Three cases are considered:
  - a single AS (resp. GS) flow is generated from PC BSD_A with a mean rate corresponding to 50% of the maximal amount MA_{AS} (resp. MA_{GS}) allocated for AS (resp. GS) traffic;
  - 2 AS (resp. GS) flows are generated from PCs BSD_A and B; their mean rates are 23 and 27% of MA_{AS} (resp. MA_{GS});
  - 4 AS (resp. GS) flows are generated from PCs BSD_A and B; their mean rates are 11, 12, 13, 14% of MA_{AS} (resp. MA_{GS});
- 1 BE flow (in the first case) or 2 BE flows (in the second and third cases) are sent from PC BSD_A and B. Sum of their mean rate is 100 Kbytes/s, i.e. about the totality of LB.

Scenario 1 (scenario 2)	% of MA$_{AS}$	Throughput of the BE traffic (% of the link bandwidth LB)
AS (BSD$_A$)	50	100 (BSD$_B$)
AS$_1$ (BSD$_A$) AS$_2$ (BSD$_B$)	23 27	100 (50 % BSD$_A$ - 50 % BSD$_B$)
AS$_{11}$ (BSD$_A$) AS$_{12}$ (BSD$_A$) AS$_{21}$ (BSD$_B$) AS$_{22}$ (BSD$_B$)	11 12 13 14	100 (50 % BSD$_A$ - 50 % BSD$_B$)

Table 1. Traffic spec. for scen. 1 (replace AS with GS for scen. 2)

2) *Scenario 3.* For those measurements (see Table 2):

– AS and GS flows are sent from PC BSD$_A$ and B to PC BSD$_C$; two cases are considered:

- a single AS flow is generated from PC BSD$_A$ with a mean rate corresponding to 100% of MA$_{AS}$. In parallel, a single GS flow is generated from PC BSD$_B$ with a mean rate corresponding to 100% of MA$_{GS}$;
- 2 AS flows are generated from PCs BSD$_A$ and B; for each one, the mean rate corresponds to 50% of MA$_{AS}$. In parallel, two GS flows are generated from PCs BSD$_A$ and B; for each one, the mean rates corresponds to 50% of MA$_{GS}$;

– 2 BE flows are sent from PC BSD$_A$ and $_B$. Sum of their mean rates is 100 Kbytes/s, i.e. about the totality of LB.

Scenario 3	% of MA$_{AS\ or\ GS}$	Throughput of the BE traffic (% of the link bandwidth LB)
AS (BSD$_A$) GS (BSD$_B$)	100 100	100 (50 % BSD$_A$ - 50 % BSD$_B$)
AS$_1$ (BSD$_A$) AS$_2$ (BSD$_B$) GS$_1$ (BSD$_A$) GS$_2$ (BSD$_B$)	50 50 50 50	100 (50 % BSD$_A$ - 50 % BSD$_B$)

Table 2. Traffic specification for scenario 3

Let us precise that all flows are generated by bursts of one 1024 bytes length UDP packet by means of a software tool named *Debit6*, able to send UDP traffic respecting a token bucket like profile. Throughput and loss rate for a given session, and transit delay for each packet, are collected by *Debit6* in reception; inter-packet delay is the parameter used to change the throughput of the generated flows. Hosts are synchronized using *Network Time Protocol*, inducing a +/- 5 ms uncertainty on delay measurements.

E. Results and analysis

Results are given by means of: a figure representing on the y-axis the % of packets received with a transit delay less than the value denoted on the x-axis, and a table indicating for each flow the loss rate and {min, average. and max} values of the transit delay.

1) *Scenario 1.* The impact of the number of AS flows on the AS QoS is weak. Indeed, Table 3 indicate a variation smaller than 5 ms for the average value of the delay. Fig. 5 enforces this result: for 90% of the packets, delay is almost unchanged. However, one can notice that 10% of the packets (for AS1, 2, 11, 12, 21, 22) have a delay much greater than for the one observed for the single AS flow. Our first explanation was to associate this result to the asynchronism of the PCs OS (Free BSD). As this phenomenon does not appear for GS experiments (see results of scenario 2), this explanation seems not valid. At the present time, no valid explanation as been given. Note that the loss rate is unchanged (no loss).

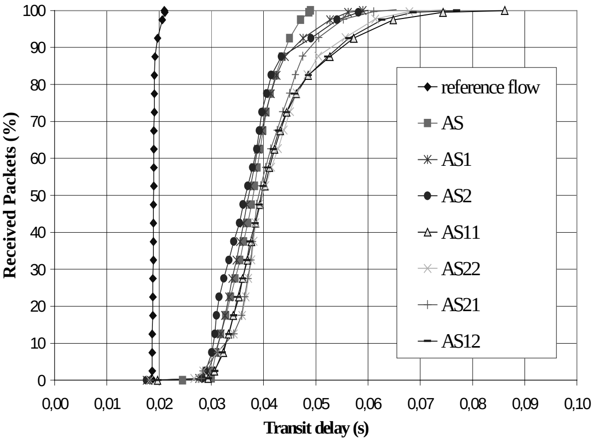

Fig. 5. Results of scenario 1 (1/2)

Note: the curve named *reference flow* has been obtained for a single AS flow without any other traffic in the network.

Delay (ms)	AS	AS1	AS2	AS11	AS12	AS21	AS22
- minimal	25	18	18	20	18	18	20
- average	38	38	37	42	42	40	41
- maximal	49	59	63	86	75	65	77
Loss rate	0	0	0	0	0	0	0

Table 3. Results of scenario 1 (2/2)

2) *Scenario 2.* The impact of the number of GS flows on the GS QoS is weak. Indeed, Table 4 indicates a variation smaller than 8 ms for the average value of the transit delay. Fig. 6 enforces this result for all packets. Note that the loss rate is unchanged (no loss).

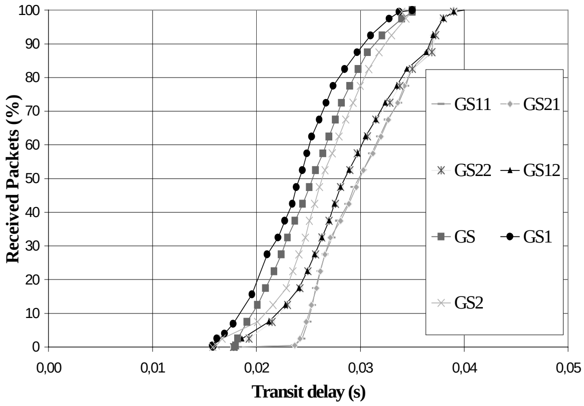

Fig. 6. Results of scenario 2 (1/2)

Delay (ms)	GS	GS1	GS2	GS11	GS12	GS21	GS22
- minimal	19	16	16	18	18	18	18
- average	25	26	26	32	31	33	31
- maximal	33	33	35	33	34	37	38
Loss rate	0	0	0	0	0	0	0

Table 4. Results of scenario 2 (2/2)

Note : the 20 ms maximal difference that appears for the delay is acceptable. Indeed, keeping in mind that: (1) GS transit delay for a given packet is expected to be the minimal one with a possible jitter equivalent to one buffered GS packet, (2) emission of a BE packet can't be interrupted, and (3) all packets have a fixed 1 Kbytes length and are sent with a rate limited to 100 Kbytes/s, it then results that a 20 ms additional delay appears in the worse case.

3) *Scenario 3*. The impact of the number of GS flows on the AS QoS (and reciprocally) is almost null. Indeed, Table 5 indicates a variation smaller than 6 ms for AS and 2 ms for GS for the average value of the delay. This is confirmed by Fig. 7 for all packets. Loss rate is still unchanged (null). Finally note that the transit delay is almost the same as the one observed for the AS (resp. GS) flow of Table 3 and Fig. 5 (resp. Table 4 and Fig. 6).

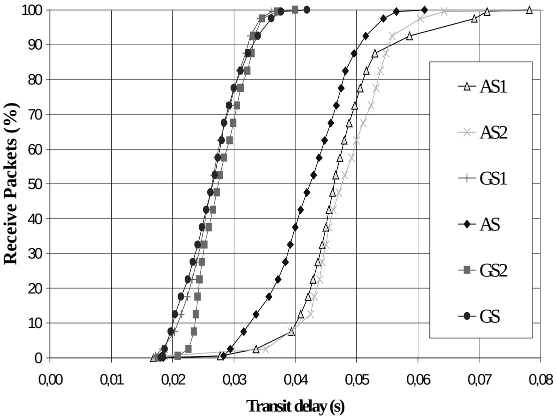

Fig. 7. Results of scenario 3 (1/2)

Delay (ms)	AS	GS	AS1	AS2	GS1	GS2
- minimal	19	18	17	17	17	18
- average	42	26	47	48	27	28
- maximal	61	42	78	88	41	40
Loss rate	0	0	0	0	0	0

Table 5. Results of scenario 3 (2/2)

IV. CONCLUSIONS AND FUTURE WORK

Contributing to the DiffServ area research, work presented in this article deals with the conception, the implementation and the performance analysis of a communication architecture supporting differentiated services at the IP level and a per flow QoS at the end to end level. Architecture principles and services have been exposed in section II; their implementation over the national ATM platform RENATER2 has also been described[1] in this section. Finally, an experimental evaluation of the QoS provided at the user level has been exposed and analyzed in section III.

Several conclusions may be stated that extend those given in [20] which were: (1) a differentiated services architecture may be easily deployed over a VPN-like environment such as the @IRS one, and (2) QoS evaluated for a single AS (resp. GS) UDP flow is conform to the expected one. Measurements exposed in this paper allows one to conclude that: (1) the impact of the number of GS flows on the AS or GS QoS is weak, and (2) the impact of the number of AS flows is similar but it may be discussed a little more. Indeed, if AS QoS is almost unchanged for 90% of the traffic, 10% of the packets have a delay slightly increased. Although no explanation is given at the present time, this result is acceptable with regard to the AS QoS specification; moreover, it is particularly important for the characterization of an AS-like service on a DiffServ platform like the @IRS one: indeed, a strong impact would have been made difficult such a characterization.

Three major perspectives are currently under development: the first one is to evaluate the impact of the IP parameters (such as routers queue length, WFQ weights, etc.) on the QoS; the second perspective is to formalize the semantics of guarantee associated with the QoS parameters, and then to develop a mechanism allowing the application to be dispensed from the explicit choice of the Transport and IP level services. This mechanism will be based on an *a priori* known characterization of the AS QoS. The third perspective is the extension of this work to a multi-domain environment.

REFERENCES

[1] IntServ: www.ietf.org/html.charters/intserv-charter.html

[2] DiffServ: www.ietf.org/html.charters/diffserv-charter.html

[3] TF-TANT: http://www.dante.net/tf-tant

[4] M. Campanella et al, "Spec. and impl. plan for a Premium IP service", www.dante.org.uk/tf-ngn/GEA-01-032.pdf.

[5] TEQUILLA: http://www.ist-tequila.org/

[6] CADENUS: http://www.cadenus.org/

[7] AQUILA: http://www-st.inf.tu-dresden.de/aquila

[8] GCAP: http://www.laas.fr/GCAP/

[9] S. Blake, D. Black, M. Carlson, "An Architecture for Differentiated Services", RFC 2475.

[10] A. Campbell, G. Coulson, D. Hutchinson, "A QoS architecture", ACM Computer Com. Review, 1994.

[11] K. Nahrstedt, J. Smith, "Design, Impl. and experiences of the OMEGA end-point archi.", IEEE JSAC, vol.14, 1996.

[12] C. Chassot, A. Lozes, M. Diaz, "From the partial order (PO) concept to PO connection", JHSN, vol5, n°2, 1996.

[13] P. Amer et al, "Partial Order Transport Serv. for MM and other Appli.", IEEE/ACM Trans. on Net, vol.2, n°5, 1994.

[14] T. Connolly, P. Amer, P. Conrad, "An Extension to TCP: Partial Order Service", RFC 1693.

[15] K. Nichols, V. Jacobson, L. Zhang, L, "A Two-bit Differentiated Services Archi. for the Internet", 1997.

[16] T. Bonald, M. May, J. Bolot, "Analytic Evaluation of RED Performance", INFOCOM'2000, Tel Aviv, Israel, 2000.

[17] T. Ziegler et al. "Stability of RED with two-way TCP Traffic", ICCCN, Las Vegas, Oct 2000.

[18] J. Heinanhen, F. Baker, W. Weiss, and al, "An Assured forwarding PHB", RFC 2597.

[19] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC 2598.

[20] F. Garcia et al, "Conception, implementation and evaluation of a QoS based archi. for an IP env^t supporting differentiated services", IDMS, Lancaster, UK, Sept. 2001.

---

[1] The resultant platform, named the *@IRSBone*, is now available for GroupWare activities of other RNRT projects.