Automatic Management of the QoS within an Architecture Integrating New Transport and IP Services in a DiffServ Internet

C. Chassot, G. Auriol, and M. Diaz

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS (LAAS-CNRS)
7, avenue du Colonel Roche, 31077 Toulouse cedex 04. France
{chassot, gauriol, diaz}@laas.fr

Abstract. Lots of research works have been performed for about ten years around the QoS problem in the Internet, both but *separately* at the Transport and at the IP levels. Taking into account the emerging traffic engineering-based QoS solutions (Diffserv-oriented), this paper targets the integration of new Transport services and protocols together with these solutions. Starting from performance measurements performed over a national DiffServ plat-form, contributions exposed here deal with the proposition and the implementation of a session level protocol allowing the application programmers to be masked with the complexity of choosing the underlying new Transport and IP services, still being provided with a per flow QoS.

1 Introduction

Technical evolutions in telecommunications and computer science have led to the development of new types of distributed applications such as interactive videoconferencing systems or distributed interactive simulation. These applications present challenging characteristics to network designers, such as the need for bounded delays together with small loss rates, guaranteed throughputs, etc. In order to tackle these new needs, several works have been performed both but *separately* at the Transport level then at the IP level.

As far as the Transport level is concerned, an important work has been performed in order to define new Transport protocols and architectures, both well suited with regard to the features of multimedia applications and independent of the underlying network layer technology. Particularly, starting from the fact that existing protocols only provide two kinds of service, "reliable and ordered" one (TCP) or "unreliable and unordered" one (UDP), whereas multimedia applications have partial reliability (PR) / partial order (PO)[1] constraints, the need became obvious to provide applications with PR / PO Transport services [1] [2]. Two major benefits result from the use of these new services within multimedia Transport architecture: [3] proved that PR/PO protocols allow to optimize both network resources (end-to-end storage buffers and bandwidth) and the transit delay of the application data units, still respecting reliability and logical synchronization constraints inherent to the distribution of a multimedia document.

---

[1] I.e. logical synchronization between, for instance, audio and video.

A. Marshall and N. Agoulmine: MMNS 2003, LNCS 2839, pp. 286–299, 2003.

As far as the IP level is concerned, several research projects have been initiated to target the QoS problem through traffic engineering in a DiffServ context [4] integrating (often) the MPLS technology. In the European context, let us cite in particular the TF-TANT activity and the TEQUILA, AQUILA and CADENUS IST projects[2], proposing and/or implementing architectural frameworks for providing QoS in a multi-domain environment. In TEQUILA for instance, the focus has been made on a *policy-based* QoS management system, a policy being defined as a "way to guide the behavior of the network through high-level declarative directives" [5] [6] [7].

From the previous considerations, it results the need of an end-to-end communication architecture: (1) *integrating the QoS mechanisms and protocols developed at the IP and Transport levels,* and (2) able to provide the Application level with *guaranteed end-to-end QoS*, including order/reliability parameters together with temporal ones (transit delay).
Work presented in this paper targets this problem, our final goal being to develop an end-to-end architecture for a multi-domain DiffServ environment providing a guaranteed QoS (including reliability and transit delay) for each Application level data flow, and accessible through a generic API[3], allowing its user: (1) to express the required QoS with generic (i.e. non ad hoc) parameters; (2) to be masked with the complexity of choosing and using the underlying new Transport and IP services, by mean of a specific management protocol. Contributions exposed in this paper are the following:

– in section 2, we propose to formalize the conceptual link between the application needs and the communication system by defining how to map the application requirements together with deterministic or non deterministic underlying QoS services (such as AF-based ones at the IP level). This formalization is included in an actual end-to-end communication architecture (the @IRS[4] architecture) whose principles, implementation and performance measurements are also exposed;
– in section 3, we first study how to quantify the end-to-end QoS resulting from the coupling between new PR/PO Transport services together with EF or AF-based IP QoS in a mono domain DiffServ environment. Then we propose an algorithm allowing to select the adequate Transport and IP services from an Application level QoS request. Experimental tests allowing to observe the behavior of the end-to-end communication system for a videoconference application are exposed and analyzed;
– conclusions and future work are finally exposed in section 4.

2 The @IRS End-to-End Architecture

The basic principle that supports the @IRS architecture is one of many dedicated to the transfer of multimedia documents [3] [8] [9]. The idea is that the exchanged traffic can be decomposed into several data flows (one per media for a multimedia application), each one

---

[2] http://www.dante.net/tf-tant, http://www.ist-tequila.org/, http://www.cadenus.org/, http://www-st.inf.tu-dresden.de/aquila

[3] API: Application Programming Interface

[4] @IRS (*Integrated Networks and Services Architecture)* is the name of a project of the French *National Network of the Telecommunication Research*. Finished since the end of 2001, it is currently pursued in a second phase (@IRS++).

requiring a specific QoS in terms of reliability, transit delay, etc. In the @IRS architecture, the application software is then allowed, through a specific API (see Fig. 1), to establish *sessions* containing one or many *end-to-end communication channels*, each one being: (1) unicast and unidirectional, (2) dedicated to the transfer of a single applicative data flow, and (3) able to offer a specific QoS, including order, reliability and transit delay as explicit parameters, together with new *semantics of guarantee* detailed here after.

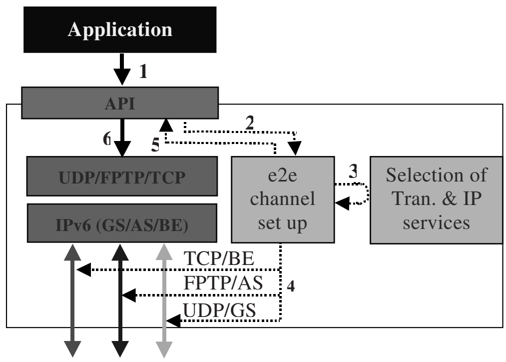

Fig. 1. @IRS end-to-end architecture

To perform the required QoS, two levels of protocols have been considered in the @IRS architecture: the Transport and the IP levels, both being able to provide three kinds of services (described in sections (2.2) and (2.3)). In the first version of the architecture, the Application level had to select for each of its flows both the Transport service and the IP service it wanted to be applied on each end-to-end channel. Such a choice is not obvious for two major reasons: the first one is that application programmers are not supposed to be expert in network protocols ! The second reason comes from a knowledge even network experts do not have yet: what is the QoS resulting from the coupling of new IP services (particularly AF[5]-based services) together with new Transport services ?
The @IRS architecture we are talking about in this paper extends the initial version in that it includes an *automatic* selection of the Transport and IP services able to perform (if possible) the QoS required for a flow (see section 3). Before addressing this part of our work, we first expose the main components of the @IRS architecture and then the networking platform over which it has been implemented, tested and evaluated.

2.1 API : QoS Parameters and Semantics of Guarantee

Several definitions of QoS parameters have been proposed in the past that often depend on the tested applications. The @IRS API is a "generic" API, i.e. parameters have been defined so as to take into account all possible kinds of applications needs.

---

[5] AF: *Assured Forwarding* per hop behavior [10].

QoS parameters. The @IRS QoS parameters are:
– an intra and inter flow partial order, expressing logical synchronization constraints either within or between flows;
– a partial reliability and a maximum end-to-end transit delay defined by:
- the percentage, noted τ_r, of sent packets that the application wishes to receive;
- the percentage, noted (τ_d, [a, b]), of sent packets that the application wishes to receive in the time interval [a, b];
- the maximal number of consecutive lost packets.

Guarantee Semantics
TCP and UDP respectively provide total or null reliability/order guarantee. Analogously, the semantics introduced here allow to establish a link between the applications needs and the communication system as far as the delay and reliability are concerned. More precisely, τ_r et τ_d are associated with the following semantics:
– a *absolute guarantee*, noted *A* meaning that the parameter value will be exactly obtained, for example by using a Transport level retransmission mechanism;
– an *average guarantee*, noted *M* meaning that the parameter value will be obtained by a statistical characterization of delay and reliability. Here, there is an incertitude on the parameter value, even if the communication system accepts the request.
– an *average guarantee with notification*, noted *N*, that extends the M guarantee by a notification to the application when the parameter value is not respected.

2.2 Transport Layer

Three services have been defined at the Transport layer:
– the first one is implemented by TCP and provides total order and total reliability guarantees on the data transfer;
– the second one is implemented by the PR/PO FPTP[6] protocol [11] that provides programmable partial order and partial reliability guarantees on the data transfer. In the following, we consider that this protocol may also be configured so as to limit the number of its retransmissions (when a loss occurs) to a given number *n*;
– the third service is implemented by UDP and provides neither order nor reliability.

2.3 IP Layer

Three services have been defined and implemented at the IP level:
– GS (*Guaranteed Service* - analogous to the *Premium Service*) provides its user with an almost fixed transit delay and a total reliability concerning routers congestions (however, destination host congestions or bit errors may occur);
– AS (*Assured Service*) provides *in-profile* traffics with an almost total reliability (concerning routers congestions) and an "acceptable" transit delay variation when

---

[6] FPTP (*Fully Programmable Transport Protocol)* is a PR/PO protocol implemented and tested over an international IPv6 platform in the IST GCAP project http://www.laas.fr/GCAP/

congestions occur in the network. Part of the traffic exceeding the characterization profile is conveyed in AS as far as no congestion occurs on the path used by the flow;
– BE (*Best Effort*) provides no QoS guarantee.
Implementation of these services (which follows the specification given in [10] and [12]) is based on several mechanisms described hereafter. Implementation details at routers input and output interfaces may be found in [13].

Control path QoS mechanisms. The main mechanism involved in the control path is the *admission control* which takes care of the acceptance of new AS or GS flows, a flow being identified (in @IRS) by the *flow id* and the source IP address of the IPv6 packets:
– 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 guaranties that the amount of *in-profile* packets (i.e. respecting the traffic contact) 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 is supposed to involve all the routers on the data path. All our experiments have been done under the hypothesis that such an admission control was done.

Note that in the two cases, the path between the considered hosts is supposed to be fixed.

Data path QoS mechanisms. QoS functions involved in the data path are policing, scheduling and congestion control.
Policing and congestion control. Policing deals with the actions to be taken when *out-of-profile* traffic arrives at the edge of the network. For AS, the action is to mark the out-of-profile packets. When congestion occur, packets marked "OUT" are dropped first by means of a congestion control mechanism called *Partial Buffer Sharing* (PBS). For GS, 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 ; the remaining bandwidth is shared by a *Weighted Fair Queuing* (WFQ) between AS and BE traffic.

2.4 @IRS Platform Configurati on and Performance Measurement Results

The @IRS architecture has been implemented over a national IPv6 platform (Fig. 2).

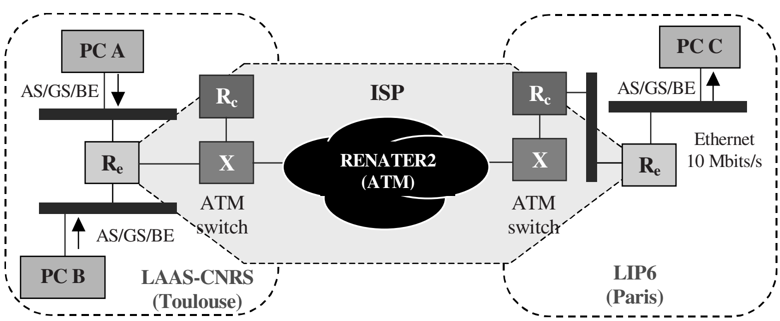

Fig. 2. Platform configuration (2 sites on 7)

Platform configuration. Seven local platforms[7] have been connected by an edge router (R_e) to an *Internet Service Provider* (ISP) represented by the national ATM RENATER2 platform. By means of its edge router, each site was then provided with an access point to the ISP, characterized by a traffic contract (equivalent to the *Service Level Agreement* of [BLA98]). It was the edge router's responsibility to implement the SLA as it introduced flows within the ISP. Bandwidth of the link connecting sites to the ISP (via a CBR ATM Virtual Path) was such that the maximal throughput provided at the UDP level is 107 Kbytes/s for 1024 bytes length packets.

Edge and core routers were configured with the following hypothesis:
– the max amount of GS (resp. AS) traffic that could be introduced by the edge router was 20 Kbytes/s (resp. 40 Kbytes/s), i.e. 20% (resp. 40%) of the link bandwidth;
– the rate control applied by the core router was 100 Kbytes/s;
– the weights associated to the AS and BE scheduling (WFQ) were resp. 0.5 and 0.5.

Performance measurements: results and analysis. The goal of the performance measurements exposed hereafter was to answer the following question: given that a DiffServ-oriented solution was implemented on the tested platform[8], would it be possible to observe a *reproducible* per flow QoS (for AS and GS flows)?
Measurements have been realized in order to evaluate the QoS provided to several UDP flows served in AS or GS whose number and load were varying. For experiment sessions (about 300 seconds), measured parameters were (1) the loss rate and (2) the minimal, maximal and average values of the transit delay together with the transit delay distribution[9]. A complete description of the three scenarios may be found in [14].
All the measurements have been done between two sending hosts (BSD PCs) located at Toulouse, noted A and B in Fig. 2, and one receiving host located at Paris, noted C. Three scenarios have been defined. Each time, in order to test the "worst" case, a BE traffic used the totality of the link bandwidth:
– the first scenario aimed at validating the impact of the number of AS flows on the AS QoS, when the network was overloaded; no GS flow was generated;
– the second scenario aimed at validating the impact of the number of GS flows on the GS QoS, when the network was overloaded; no AS flow was generated;
– the third scenario aimed at validating the impact of the number of AS (resp. GS) flows on the GS (resp. AS) QoS, when the network was overloaded; AS and GS flows were generated together.
Results (that are partially exposed in Fig. 3) have allowed one to conclude that:
– the impact of the number of GS flows on the AS or GS QoS was weak;
– the impact of the number of AS flows was similar but it might be discussed a little more. Indeed, if AS QoS was almost unchanged for about 90% of the traffic, 10% of the packets had a delay slightly increased. Whereas no solid explanation has been

---

7 Only two local platforms are represented in Fig.2 (LAAS and LIP6 platforms).
8 Let us keep in mind that a DiffServ-oriented QoS approach in the core network is applied on IP packets coming from several flows.
9 The study has been done for a given and unchanged configuration of the platform (size of routers queues, WFQ weights, etc.). Indeed, our goal was neither to evaluate the impact of the platform configuration, nor its topology.

given, 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 (described hereafter in section 3).
Let us precise that these results are not representative of an exhaustive study. Such a study has been performed in simulation (based on *ns*-2), our goal being also to evaluate the impact of the network topology. This work has not yet been published.

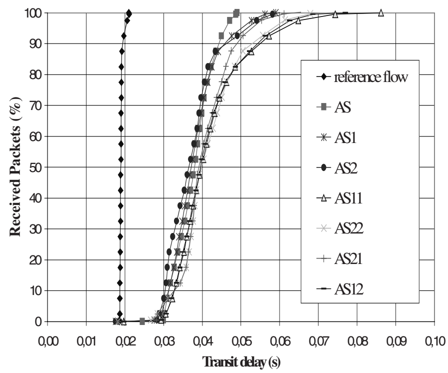

Fig. 3(a). Results of scenario 1

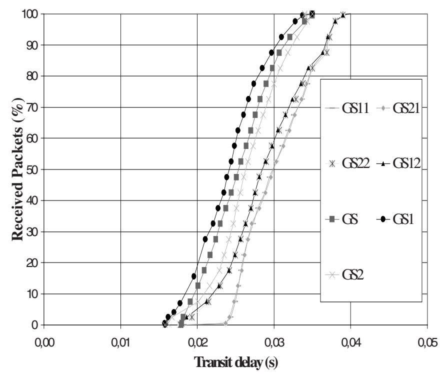

Fig. 3(b). Results of scenario 2

3 Towards an Architecture Integrating Transport and IP QoS

3.1 Characterization of the QoS Resulting from Transport and IP Services

Starting from the previous results, the goal of this section is to propose a characterization of the QoS resulting from the coupling between a DiffServ IP service (such as GS or AS) and a Transport protocol allowing n retransmissions ($FPTP_n$, $n \geq 0$). Two specific QoS parameters are targeted: the loss rate and the transit delay between two points (e.g., two edges routers), and for a given state of the network (e.g., a state of congestion).

Characterization of the QoS for a GS-based service. From the previous performance measurements results, it comes that the GS service may be characterized (per flow) by a total reliability and an end-to-end maximal transit delay. As this service has to provide a total reliability, the interest of its coupling with another Transport service than the ones provided by UDP or $FPTP_0$ is not obvious (except when error bits occur). In the following, we'll do the hypothesis that GS is always associated with UDP or $FPTP_0$.

Characterization of the QoS for an AS-based service. The characterization of the loss rate and the transit delay resulting from the coupling of an AS service together with a Transport service allowing n retransmissions has to be discussed a little more.

Loss rate characterization. *Let:*
– ε be the estimated percentage of AS packets which are lost by routers congestions;

– $\mathbf{r_n}$ be the estimated percentage of Application data units transferred between the two considered points), *n* designing the number of Transport level retransmission(s).
For n = 0, i.e. without Transport level retransmission: $r_0 = 1 - \varepsilon$
For n = 1, i.e. after one retransmission: $r_1 = (1 - \varepsilon) + \varepsilon . (1 - \varepsilon) = 1 - \varepsilon^2$
For n = 2, i.e. after two retransmissions: $r_2 = (1 - \varepsilon) + \varepsilon . (1 - \varepsilon) + \varepsilon . [\varepsilon.(1 - \varepsilon)] = 1 - \varepsilon^3$
Finally, r_n is defined by: $\mathbf{r_n = 1 - \varepsilon^{n+1}}$

End-to-end delay transit characterization. Let **f(t)** represent the percentage of AS packets which are received without retransmission with an end-to-end transit delay less than or equal to *t*. This function directly depends on:
– the DiffServ domain configuration, i.e. queue sizes, weights of WFQ, etc.
– the path between the sending and receiving hosts;
– the load of the network on the considered path.
However, it comes from the previous section (2.4) that f(t) is sufficient enough to statistically characterize the AS service between two points on a given path and for a certain state of the network[10] (in our measurements, a state of congestion).
Let $\mathbf{f_{n,T}(t)}$ be the function representing the delay distribution after *n* Transport level retransmissions. For n = 1, the relation between $f_{1,T}(t)$ and f(t) is: $f_{1,T}(t) = f(t) + \varepsilon . f(t - T)$
- T being the value of the TPDU[11] retransmission timer (supposed to be constant).
For n = 2, $f_{2,T}(t)$ is defined by: $f_{2,T}(t) = f(t) + \varepsilon . f(t - T) + \varepsilon [\varepsilon . f(t - 2.T)]$
Finally, $\mathbf{f_{n,T}(t)}$ is defined by: $f_{n,T}(t) = f(t) + \varepsilon . f(t - T) + \ldots + \varepsilon^n . f(t - n.T)$

i.e. $$\mathbf{f_{n,T}(t) = \sum_{i=0}^{n} \varepsilon^i . f(t - i.T)}$$

Let us now have a look at the use of this characterization in order to select a couple (Transport service, IP service) able to match a QoS request expressed for an end-to-end channel by means of the QoS parameters and semantics of the @IRS API.

3.2 Service Selection Algorithm

In order to simplify the following explanations, we'll consider that the QoS request is expressed by means of only two QoS parameters: (τ_d, [a = 0, b]) and τ_r and that the semantics are reduced to the absolute one (A) and the average one (M).
Let an application be requiring a given QoS for one of its flows between two sites noted S_A and S_B. Let us also suppose that:
– UDP, $FPTP_n$ and TCP are the available protocols at the Transport level;
– GS, AS, BE are the available services at the IP level;
– when the network is in state of congestion (supposed to be the "worst" case)
  - GS is characterized between S_A and S_B by a max delay = t_0 and a loss rate = 0;
  - AS is characterized between S_A and S_B by $f_n(t)$ and r_n.

---

[10] Let us recall here that this assertion (hypothesis on which is based the following of this work) is currently studied in simulation (*ns*-2), our goal being to identify its limits, particularly with regard to the network topology.
[11] TPDU: Transport Protocol Data Unit.

Algorithm description. To know if the QoS request may be satisfied, the following algorithm must be applied.

•) Verify the request *coherency* and then:
- in case of incoherency, reject the request;
- in case of coherency:
  - •) Choose a couple (Transport / IP)
    - ▪ if the IP service = AS, evaluate if it allows to satisfy the QoS request
      - if the evaluation fails, choose another couple (that necessarily exists);
  - •) if the IP service = GS or AS then perform an admission control
    - ▪ if the control fails, return to the previous point and choose another couple;
    - ▪ in case of failure for all couples, reject the request.

Request coherency verification. The request is *coherent* when:
– the packet rate (τ_r) the application wishes to receive is higher than the packet rate the application wishes to receive with a specific transit delay (τ_d), i.e: $\mathbf{0 \leq \tau_d \leq \tau_r \leq 1}$
– the maximal transit delay *b* is greater than or equal to the GS end-to-end transit delay, i.e: $\mathbf{b \geq t_0}$.

Choice of a Transport / IP couple. Once done this coherency verification, the choice has to be done of an IP and a Transport services are chosen allowing for the satisfaction of the QoS parameters together with their semantics of guarantee.
This choice is based on the consultation of a knowledge base (see Fig. 4) that provides, for all possible configurations of (τ_r, σ_r) and (τ_d, σ_d), one or many couples (Transport / IP) allowing (a priori) for the satisfaction of the request. The way this base is initialized is described at the end of the section.

<table>
<tr><td></td><td>$\tau_r = 0$</td><td colspan="3">$0 < \tau_r < 1$</td><td>$\tau_r = 1$</td></tr>
<tr><td>$\tau_d = 0$</td><td>UDP/BE$^{(*)}$</td><td colspan="3">FPTP$_n$/BE$^{(*)}$</td><td>TCP/BE$^{(*)}$</td></tr>
<tr><td rowspan="3">$0 < \tau_d$ < 1</td><td rowspan="3">∅</td><td>σ_d \ σ_r</td><td>A</td><td>M</td><td rowspan="3">UDP/GS$^{(*)}$</td></tr>
<tr><td>A</td><td>UDP/GS</td><td>UDP/GS</td></tr>
<tr><td>M</td><td>UDP/GS</td><td>C</td></tr>
<tr><td>$\tau_d = 1$</td><td>∅</td><td colspan="3">∅</td><td>UDP/GS$^{(*)}$</td></tr>
</table>

Fig. 4. Knowledge base ("∅": incoherent request - (*): whatever the semantic)

When several couples are possible (case C in the table of Fig. 4), then the "cheapest" one is selected, the "cost" of a couple being defined as it follows:
– at the IP level: GS cost > AS cost > BE cost
– at the Transport level: TCP cost > FPTP$_n$ cost > UDP cost
– the choice is done so as to minimize first the IP cost (e.g.: FPTP/AS < UDP/GS).
For each choice, there are three possibilities depending on the IP service:
– if the IP service is BE, the choice has no more to be discussed; it is retained;
– if the IP service is GS, the choice is retained if the admission control is positive;

– if the IP service is AS, the choice has to be studied a little more so as to verify if it matches the QoS required for the parameters τ_d and τ_r.

Evaluation of the choice (Transport / IP) when the IP service is AS.

Let $\mathbf{n_r}$ be the minimal retransmissions number allowing for the satisfaction of τ_r

=> n_r must verify: $\tau_r = 1 - \varepsilon^{nr+1}$

It then results that: $\mathbf{n_r = Ent\,[\,(\,ln\,(1 - \tau_r) / ln(\varepsilon)\,) - 1\,)\,]}$

- with: ln is the logarithm function and Ent is the *entire part* function (e.g. Ent(0.5)=0).

Let now $\mathbf{n_d}$ be the maximal retransmissions number allowed by the *b* parameter

=> n_d must verify: $n_d \,.\, T + t_0 = b$

It then results that: $\mathbf{n_d = Ent\,[\,(b - t_0) / T\,]}$

- with: T is the value of the TPDU retransmission timer (supposed here to be constant).

In order to satisfy τ_d, the question is: $\exists ?\ n \in [0, n_d] \,/\, f_n(b) \geq \tau_d$

If *n* does not exist, the couple (* / AS)[12] has to be rejected (because τ_d cannot be satisfied)

else | if $n_d < n_r$ then the couple (* / AS) has to be rejected (τ_r cannot be satisfied)
else the choice is retained and the number of retransmissions is $p = \max\,[n, n_r]$.
In other words, the choice is AS/FPTP$_p$ with $p = \max\,[n, n_r]$[13].

Initialization of the Knowledge Base

The initialization of the knowledge base depends on:

– the available services, both at the Transport and at the IP levels;
– the set of definition of parameters and semantics, e.g.:
  ▪ for τ_r and τ_d: the interval [0,1] divided into three subsets : 0,]0,1[and 1;
  ▪ for σ_r et σ_d: A and M.

For each (τ_r, σ_r) / (τ_d, b, σ_d) possible combination, all the solutions allowing for the satisfaction of each couple separately (i.e. (τ_r, σ_r) on one side, (τ_d, σ_d) on the other side) have been identified, but only the solutions allowing for the satisfaction of the two couples have been kept in the table. Let us illustrate this by way of a simple example. Consider the following combination: ($0 < \tau_r < 1$, $\sigma_r = A$) / ($0 < \tau_d < 1$, [a, b], $\sigma_d = M$) :

– two solutions are possible for the request related to the couple (τ_r, σ_r):
  ▪ * / GS, where * designates all possible Transport services ;
  ▪ FPTP$_\infty$ / *, where * designates all possible IP services ("∞" designating that there is no limit on the retransmission number);
– three solutions are possible for the request related to the couple (τ_d, b, σ_d):
  ▪ UDP / AS if $\tau_d \leq f(b)$
  ▪ FPTP$_n$ / AS if $\exists n \,/\, \tau_d \leq f_{n,T}(b)$
  ▪ UDP / GS if $b \geq t_0$ (condition verified in the coherency test)

The single solution satisfying the two couples is then UDP/GS.

---

[12] * designating either UDP or FPTP$_n$.

[13] If $n = n_r = 0$, the choice between UDP/AS and FPTP$_0$/AS depends on the "partial order" QoS parameter.

Let us now look at the implementation and the test the algorithm for an actual multimedia application (a videoconferencing system).

3.3 Implementation and Experimental Tests

In addition to the service selection algorithm, two QoS management protocols have been specified and implemented (see Fig. 5). They allow the channel set up and the acceptation or reject of a QoS requested for the considered channel (including the service selection algorithm).

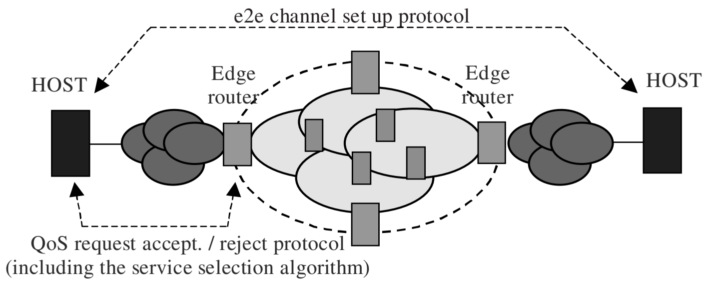

Fig. 5. QoS management protocols (before the data transfer phase)

Due to space limits, these protocols are not described here and the focus is only done on the service selection algorithm.

Implementation of the service selection algorithm. In order to minimize the storage buffer of the points representing the distribution of the transit delay of the AS packets (i.e. f(t)), a model based on a trigonometric function has been adopted:

$g(t) = A_3+A_1.htan(t.A_2-A_4)$ where *htan* designates the hyperbolic tangent function

Calculation of the (A_1, A_2, A_3, A_4) parameters is performed by a Matlab program. From a discrete set of points, this program allows to deduce the best approximation of f(t) for the chosen model.

Parameters given in the example illustrated in Fig. 6 provide the model g(t) of the function f(t) between Toulouse and Paris. It has been deduced from the measures performed on the @IRS platform (whose results are given in section 2.4).

Test specification. The networking platform over which different QoS requests of the application have been tested has been simplified; it only includes two hosts directly connected to each other by an Ethernet link (100Mbits/s):

– the service selection mechanism is implemented on the sending host;
– the admission control is supposed to be positive for each request;
– the service characterization file (on the sending host) contains the following information. Between the sending host and 140.93.200.33, for GS, t_0=18ms; for AS: g(t) parameters are: A_1=48.77; A_2=162.12; A_3=51.54; A_4=6.37 and ε=0.

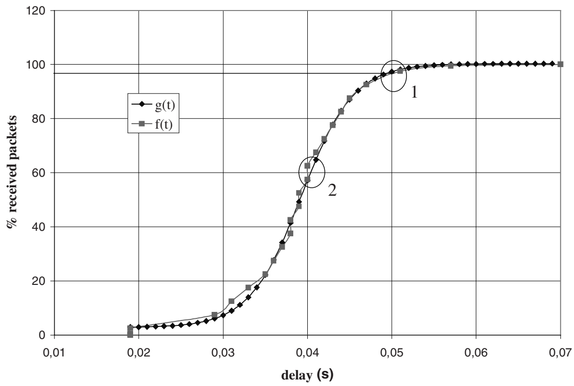

Fig. 6. Distribution of the AS packets transit delay: f(t) vs. g(t)

Three different QoS requests have been tested :

– in first case, the user wants a perfect quality both on the audio and video channels; for both channels (audio and video), this quality may be expressed by: τ_r=1, σ_r=A and τ_d=1, b=40 ms, σ_d=A;

– in the second case, no QoS is required from the user; for both channel, this quality may be expressed by: τ_r=0 and τ_d=0 (no guarantee semantic is specified)

– in the third case, the user may tolerate some possible degradations on the video but wants a sufficient enough quality on the audio so as to understand what is spoken about; this quality may be expressed as it follows:

- for the video channel: τ_r=60%, σ_r=M and τ_d=60%, b=50ms, σ_d=M
- for the audio channel: τ_r=90%, : σ_r=M and τ_d=90%, b=40 ms, σ_d=M

Results and analysis. Here are the Transport and IP services resulting from the automatic services selection for each of the three previous requests (Fig. 7).

	1st case	2d case	3d case
Audio channel	UDP/GS	UDP/BE	UDP/GS
Video channel	UDP/GS	UDP/BE	UDP/AS

Fig. 7. Results of the automatic services selection

Let us verify if this these results are correct:

In the first case, the maximal value required on the transit delay (b = 40 ms) is greater than t_0. As the required guaranty is the absolute one (A), the communication system must select UDP/GS for both channels;

In the second case, as no constraint has been expressed, the communication system must select UDP/BE for both channels (cheapest cost);

In the third case, for both video and audio channels, the question is (see section 3.2.3):

$$\exists?\ n \geq 0 \ /\ f_n(b) \geq \tau_d \text{ with } 0 \leq n \leq n_d = \text{Ent}\ [(b - t_0) / T]$$

T being the retransmission timer value (greater that the minimal RTT, i.e. $2.t_0$)

– for the video channel: b and t_0 values implicate that n=0 (no retransmission); in parallel, the point noted 1 in Fig. 6 indicates that about f(0.050)=98% (more than τ_d=60%) of the packets are estimated to be received with a transit delay less than 50ms. The system must then select UDP/AS ;
– for the audio channel: b and t_0 values implicate that n=0 again; in parallel, the point noted 2 in Fig. 6 indicates that only f(0.040)=60% (less than τ_d=90%) of the packets are estimated to be received with a transit delay less than 40ms. The system cannot select UDP/AS and must the select UDP/GS.

4 Conclusion and Future Work

From the recent evolution of the Internet QoS-oriented communication services and protocols, it comes an important need in a clear definition of a generic architecture integrating all the new solutions both at the IP level and at the Transport level. The work exposed in this paper consists in a particular instantiation of such an architectural framework. Taking into account the emerging traffic engineering-based QoS solutions, the targeted problem concerns the integration of a PO/PR Transport architecture together with these solutions. Starting from performance measurements performed over a DiffServ national platform, contributions exposed here deal with the proposition and the implementation of an end-to-end communication architecture providing guaranteed end-to-end QoS, and allowing the application programmers to be masked with the complexity of the underlying protocols and mechanisms. Our efforts have been made on (1) the definition of the services provided to the application layer (including QoS parameters and semantics of guarantee), and (2) the conception, the implementation and the test of a mechanism allowing the application programmers to be masked with the choice of the underlying new Transport and IP services when using the communication system. The major perspective currently under development tackles a larger problem related to the interconnection of several DiffServ domains (i.e. a multi domain context).

References

1. P. Amer, C. Chassot, T. Connolly et al, “Partial Order Transport Service for multimedia and other application”, IEEE ACM Transaction on Networking, vol.2, n°5, 1994.
2. C. Chassot, A Lozes, M. Diaz. “From the partial order concept to partial order connection”, Journal of High Speed Network, vol5, n°2, 1996.
3. P. Owezarski, M. Diaz, C. Chassot, “A Time Efficient Architecture for Multimedia Applications”. IEEE JSAC, April 1998, vol.16, n°3, pp 383–396.
4. S. Blake, D. Black, M. Carlson, “An Archi. for Differentiated Services”, RFC 2475.
5. P. Trimintzios, et al. "A management and control architecture for providing IP differentiated services in MPLS-based networks". IEEE Communication Magazine, Vol. 39, n°5, May 01.
6. P. Trimintzios, P.Flegkas, G. Pavlou et al. "Policy-based network dimensioning for IP differentiated services networks". IPOM’02, Dallas, USA, Nov 2002.
7. G. Cristallo, C. Jacquenet. "An approach to inter-domain traffic engineering". XVIII World Telecom. Congress (WTC 2002), Paris, France, Sept. 2002.

8. A. Campbell, G. Coulson, D. Hutchinson, “A QoS architecture”, ACM Computer Communication Review, 1994.
9. K. Nahrstedt, J. Smith. “Design, Implementation and experiences of the OMEGA end-point architecture”, IEEE JSAC, vol.14, 1996.
10. J. Heinanhen, F. Baker et al. “An Assured forwarding PHB”, RFC 2597.
11. P. Sénac, E. Exposito, M. Diaz. “Towards a new generation of generic transport protocols”, IWDC 2001., Taormina, Italie, Sept. 2001.
12. V. Jacobson, K. Nichols et al. “An Expedited Forwarding PHB”, RFC 2598.
13. F. Garcia, et al. “Conception, implementation and evaluation of a QoS based architecture for an IP environment supporting differentiated services”, IDMS, Lancaster, UK, Sept. 2001.
14. C. Chassot, F. Garcia, G. Auriol et al. “Performance Analysis for an IP Differentiated Services Network”, ICC02, New York, USA, Mai 2002.