Conception, implémentation et mesure des performances d'une architecture de communication à QdS garantie dans un domaine IPv6 à services différenciés

[*]F Garcia, [*]G Auriol, [*]C Chassot, [*]A Lozes **[°]E Lochin, [°]P Anelli**

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

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

Résumé

Les travaux de recherche décrits dans cet article portent sur la conception, l'implémentation et la mesure des performances d'une architecture de communication à qualités de service (QdS) garanties de bout en bout, dans un environnement IPv6 offrant des services différenciés au sein d'un unique domaine DiffServ. L'article présente successivement les principes de conception et les services de l'architecture, puis leur implémentation sur la plate-forme nationale ATM RENATER 2, et enfin une campagne de mesures expérimentales visant à évaluer la QdS offerte au niveau utilisateur. Les principes d'une amélioration de l'architecture en vue d'en simplifier l'utilisation par le programmeur d'application sont introduits en perspective de ces travaux..

Mots clés

Architecture de communication, qualité de service (QdS), Internet, DiffServ, mesure de performance

Abstract

Research reported in this paper 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, then their implementation over the national ATM platform RENATER 2, and finally experimental measurements evaluating the QoS provided at the user level. In order to simplify the application programmer task, principles of an architecture optimization are introduced in the future work part of the paper.

Keywords

Communication architecture, quality of service (QoS), Internet, DiffServ, performance measurement

I. Introduction

Ces dernières années, les évolutions conjointes de l'Informatique et des Télécommunications ont conduit à l'émergence de nouveaux types d'applications distribuées telles que les applications multimédias et coopératives ou la simulation interactive. Comparativement aux applications classiques, ces applications présentent des caractéristiques et des contraintes nouvelles (délai de transit borné, fiabilité, …) auxquelles ont à faire face les réseaux qui les supportent, en particulier l'Internet. Dans ce contexte, de nombreux travaux de recherche se sont focalisés sur la conception de services, protocoles et architectures de communication visant à améliorer et/ou à garantir la qualité de service (QdS) du transfert des données de ces applications. Dans la foulée des travaux initiés par des groupes de travail de l'IETF[1] tels qu'IntServ [1] ou DiffServ [2], plusieurs projets européens adressent aujourd'hui (ou viennent d'adresser) le problème de la QdS ; citons en particulier l'activité TF-TANT [3] et les projets européens GEANT, TEQUILA, CADENUS, AQUILA et GCAP [4, 5, 6, 7, 8], contribuant tous à la proposition d'architectures orientées DiffServ, dont un cadre général est défini dans [9].

1 IETF : Internet Engineering Task Force

Réalisés dans le cadre du projet @IRS[1], les travaux présentés dans cet article s'inscrivent dans ce contexte et portent sur la conception, l'implémentation et la mesure des performances d'une architecture de communication garantissant une QdS de bout en bout, dans un environnement IPv6 constituant un unique domaine DiffServ.

Les contributions ici présentées sont les suivantes :

– définition des principes d'une architecture aux services clairement spécifiés ;
– mise en œuvre de cette architecture sur la plate-forme nationale ATM RENATER 2 ;
– mesure et analyse de la QdS offerte au niveau aux applications par le système de communication.

L'article est structuré de la façon suivante. La section II présente les principes de l'architecture ainsi que ses services, puis décrit la plate-forme d'expérimentation au travers de laquelle elle a été déployée. La section III détaille les scénarios d'expérimentations permettant d'étudier la QdS de bout en bout ; les résultats des mesures correspondantes sont ensuite présentés et analysés. Les conclusions et perspectives de ces travaux sont enfin décrites en section IV.

II. Architecture : principes, services et implémentation

Les deux sections suivantes présentent successivement l'architecture définie de bout en bout et au niveau IP.

1. Architecture de bout en bout

Le principe de base qui sous-tend la proposition d'architecture de bout en bout du projet @IRS est le même que celui de plusieurs autres architectures dédiées au transport de flux multimédias [10, 11, 12]. L'idée est que le trafic échangé dans le cadre d'une application distribuée peut être éventuellement décomposé en plusieurs flux de données, requérant chacun une QdS spécifique (exprimable en termes de délai, de fiabilité, …).

Sur ces bases, les principes de l'architecture de bout en bout du projet @IRS sont les suivants. Au travers d'une session (voir Figure 1), le logiciel applicatif peut établir un ou plusieurs canaux de communication de bout en bout, chacun étant : (1) unicast ou multicast, (2) dédié au transfert d'un seul flux de données, et (3) offrant une QdS spécifique au flux véhiculé. Pour cela, il dispose d'une interface de programmation spécifique, l'API (*Application Programming Interface*), offrant les paramètres et les primitives de service nécessaires.

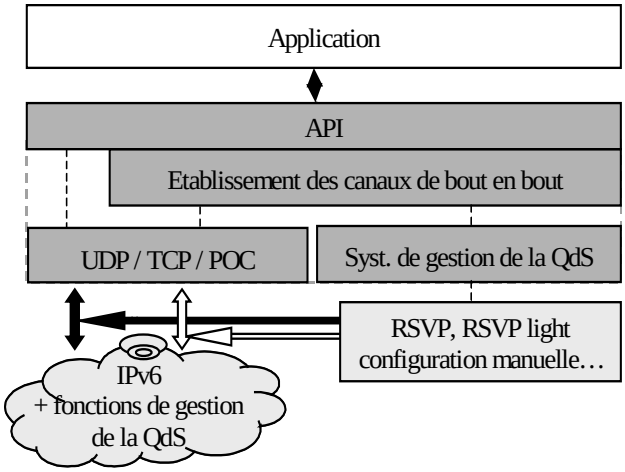

Figure 1. Architecture du système communication de bout en bout

Outre l'API, trois modules conceptuels sont mis en œuvre :

– le premier permet l'accès à plusieurs services de Transport, tels que ceux fournis par TCP, UDP ou le protocole à ordre et fiabilité partiels POC [13, 14] ;

1 Le projet @IRS (Architecture Intégrée de Réseaux et Services - Décembre 1998 - Avril 2001) est un projet français du Réseau National de la Recherche en Télécommunications (RNRT), dont l'objectif a été de développer et d'analyser de nouveaux mécanismes et protocoles pour l'Internet (IPv6, QdS et multicast notamment) dans un environnement de réseaux hétérogènes (réseaux locaux, ATM, sans fil, etc.).

– le second comporte les mécanismes relatifs à l'utilisation des services d'accès à la QdS de niveau IP (RSVP, version allégée de RSVP, …) ;
– le troisième est chargé de l'établissement des canaux de bout en bout.

Nous présentons maintenant les paramètres de services de l'API. Une description plus complète (incluant les primitives) est disponible dans [15].

Pour chaque canal, la QdS est exprimée en les termes suivants :
– un ordre partiel[1] intra-flux, exprimant les contraintes de synchronisation logique (non temporelle) de l'application sur le canal considéré ;
– une fiabilité partielle définie, par exemple, comme étant le pourcentage maximum de pertes et/ou le nombre maximum de pertes consécutives que peut tolérer l'application sur le canal ;
– un délai de transit maximum des données véhiculées au travers du canal.

Enfin, un ordre partiel inter-flux permet à l'application d'exprimer des contraintes de synchronisation entre deux flux véhiculés via des canaux différents (synchronisation d'un flux audio et d'un flux vidéo par exemple).

En plus des paramètres de QdS, l'application doit spécifier les paramètres suivants :
– le premier paramètre caractérise le trafic généré par l'application émettrice ; ici, le modèle retenu est celui du saut à jetons (*Token Bucket*) ;
– le second désigne le protocole de Transport que l'application souhaite voir mis en œuvre ;
– le troisième désigne le système de gestion de la QdS IP que l'application souhaite voir activer (IntServ ou DiffServ par exemple), ainsi que le service retenu (par exemple : les services garantis ou à charge contrôlée de l'IntServ ou les services *Premium* ou assuré de DiffServ) ;
– le dernier paramètre identifie l'adresse (unicast ou multicast) du (des) destinataire(s) applicatif(s).

Bien que l'architecture ait été conçue dans l'optique d'autoriser tous les protocoles de Transport et tous les systèmes de gestion de la QdS IP, l'architecture implémentée dans le projet @IRS ne comporte que les protocoles UDP et TCP au niveau Transport, et une proposition suivant l'approche DiffServ au niveau IP.

2. Architecture au niveau IP

Les fonctions associées à la gestion de la QdS au niveau IP peuvent être réparties en deux groupes : les fonctions relatives au chemin de contrôle (i.e. signalisation ayant pour but de configurer les routeurs afin que la QdS souhaitée puisse être mise en œuvre), et les fonctions relatives au chemin de données (i.e. transfert des données). Dans le projet @IRS, les études ont porté tant sur la partie contrôle que sur la partie donnée ; cependant, seules les fonctions du chemin de données ont été implémentées. Dans cette section, nous décrivons tout d'abord les services définis au niveau IP. Nous détaillons ensuite les principales fonctions nécessaires à leur mise en œuvre ainsi que leur implémentation au travers des routeurs de la plate-forme d'expérimentation.

1) *Services*

Trois services ont été définis au niveau IP :
– le service garanti GS (*Guaranteed Service*)[2], analogue au service *Premium* [16], a été retenu pour les flux applicatifs ayant de fortes contraintes de temps et de fiabilité. Les applications ciblées par ce service sont celles ne tolérant pas de variation de QdS ;
– le service assuré AS (*Assured Service*) a été retenu pour répondre au besoin des flux réactifs n'ayant pas de trop fortes contraintes de délai, mais requérant une bande passante minimale. Un flux servi en AS dispose d'une bande passante minimale garantie pour la partie de son trafic respectant la caractérisation de trafic formulée pour le flux considéré. La partie de son trafic excédant la caractérisation est véhiculée en AS tant qu'aucune congestion n'intervient sur le chemin emprunté par le flux ;
– le service BE (*Best Effort*) n'offre aucune garantie de QdS.

1 L'objectif de l'article n'est ni de décrire, ni d'analyser les gains induits par des canaux autorisés à perdre des paquets pour répondre aux besoins en synchronisation des applications. L'utilité de propositions basées sur RTP n'est pas non plus ici étudiée.

2 Les auteurs ont conscience du fait que la dénomination adoptée pour ce service peut prêter à confusion vis à vis du *Garanteed Service* défini suivant l'approche IntServ. Le service n'est cependant pas le même.

2) *Fonctions du chemin de contrôle*

Outre la gestion du multicast (aspect non étudié de façon couplée avec la QdS dans @IRS), les fonctions impliquées dans le chemin de contrôle sont le contrôle d'admission et le maintien des routes. Nous ne présentons ici que le contrôle d'admission.

Le contrôle d'admission a pour but d'accepter ou non de nouveaux flux dans les classes de service AS et GS. Ses décisions sont prises en respect du contrat de trafic (que nous préciserons en section III) établi entre l'utilisateur et le domaine DiffServ fournisseur des services. Notre proposition est différente pour AS et GS :

– pour AS, un contrôle d'admission par flux est appliqué en bordure du réseau (les routeurs de cœur ne sont pas impliqués) ; ce contrôle est basé sur la quantité de trafic AS déjà autorisée à entrer dans le réseau par le routeur de bordure considéré. Il permet donc d'avoir comme seule garantie qu'à tout moment, la quantité de trafic AS « dans le profil » sera toujours au plus égale à la somme des trafics AS autorisés par tous les routeurs de bordure ;

– pour GS, le contrôle d'admission implique tous les routeurs sur le chemin de données. Il est aussi appliqué par flux, chacun étant identifié par les champs *flow_id* et *adresse source* de l'entête des paquets.

3) *Fonctions du chemin de données*

Les fonctions impliquées dans le chemin de données sont le *policing*, l'ordonnancement et le contrôle de congestion.

Le *policing* consiste en les actions à mettre en œuvre lorsque un trafic (AS ou GS) hors profil se présente en entrée du réseau :

– pour AS, l'action est de marquer les paquets hors profil (marqués « OUT ») de sorte qu'ils soient en priorité rejetés en cas de congestion. Ces paquets sont dits *opportunistes* dans la mesure où ils sont traités à l'identique des paquets respectant le profil (marqués « IN ») tant qu'aucune congestion n'intervient sur le chemin emprunté ; en ce cas, le contrôle de congestion décrit ci-après est appliqué. Les applications ciblées sont celles dont le trafic est élastique, c'est à dire dont le profil est variable mais qui requièrent un débit minimum garanti ;

– pour GS, compte tenu du service, il est nécessaire que les ressources soient toujours disponibles. Tout trafic hors profil injecté dans le réseau étant susceptible de rendre indisponibles les ressources pour les trafics respectant leur profil, le *policing* adopté est donc d'écarter systématiquement les paquets GS hors profil.

L'ordonnancement (*scheduling*) est différent pour les paquets AS et GS :

– les paquets GS sont traités de façon prioritaire par le biais d'un mécanisme appelé *Priority Queuing* (PQ), induisant le plus petit délai dans le routage des paquets ;

– le reste de la bande passante est partagé alternativement entre les paquets AS et BE par le biais d'un mécanisme appelé *Weighted Fair Queuing* (WFQ).

Le contrôle de congestion vise à éviter les périodes de congestion, celles-ci pouvant empêcher le réseau d'offrir la QdS annoncée :

– pour GS, du fait du *policing* et de l'ordonnancement appliqués, il n'est pas nécessaire de mettre en œuvre un contrôle de congestion, aucune congestion n'étant susceptible d'intervenir ;

– pour AS, du trafic *opportuniste* (paquets « OUT ») pouvant être injecté dans le réseau, la quantité de trafic présent dans un routeur ne peut pas être connue *a priori* et des congestions sont susceptibles d'intervenir. Dans ce cas, les paquets « OUT » sont éliminés en premier, la politique appliquée étant celle du *Partial Buffer Sharing* (PBS) de préférence à *Random Early Discard* (RED). Ce choix est lié aux conclusions d'études qui ont montré que PBS évitait les problèmes d'oscillations de file mis en évidence pour RED [17, 18].

Nous décrivons à présent l'implémentation de ces fonctions dans la plate-forme @IRS au travers des interfaces d'entrée des routeurs de bordure et de l'interface de sortie de tous les routeurs (bordure et cœur).

Interface d'entrée des routeurs de bordure (Figure 2a)

L'interface d'entrée des routeurs de bordure est la première rencontrée par un paquet désirant entrer dans le réseau. Elle est en charge :

– de la classification des paquets au moyen des champs adresse source et *flow_id* de l'entête des paquets IPv6 (classification « multi-champs ») ;

– de la mesure des flux AS et GS afin de vérifier s'ils sont en ou hors profil ;

– du lissage (*shaping*) des paquets GS et de leur rejet si nécessaire ;
– du marquage « IN » ou « OUT » des paquets AS selon qu'ils sont en ou hors profil ;
– du marquage des paquets avec le bon DSCP (*DiffServ CodePoint)* : marque EF (resp. AF, DE[1]) pour les paquets appartenant à un flux GS (resp. AS, BE).

Interface de sortie des routeurs de bordure (Figure 2b)

Dans le modèle DiffServ, tous les routeurs doivent implémenter un ensemble de comportements appelés *Per Hop Behavior* (PHB), tels que ceux définis dans [19] et [20]. Dans l'architecture @IRS, ces comportements sont réalisés par le biais (en particulier) des fonctions d'ordonnancement et de contrôle de congestion, mises en œuvre à l'interface de sortie de tous les routeurs (bordure et cœur). En outre, cette interface est également en charge :
– de la classification des paquets au moyen du champ DSCP des paquets IPv6 (classification « agrégée ») ;
– d'un contrôle de débit (pour les routeurs de cœur), nécessaire pour éviter les congestion du commutateur ATM local à chaque plate-forme.

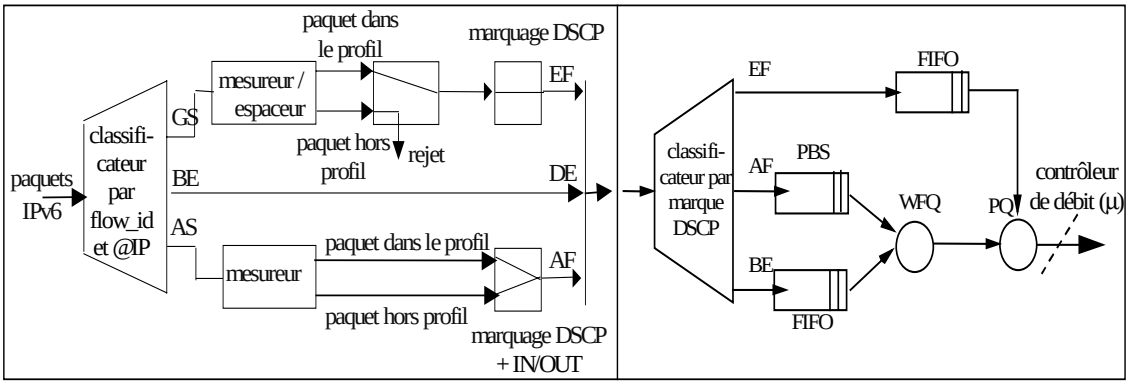

Figure 2(a) Interface d'entrée des routeurs de bordure

Figure 2(b) Interface de sortie de tous les routeurs (bordure et cœur)

III. Mesure et analyse des performances du système

Un premier ensemble de mesures a été réalisé sur la plate-forme @IRS [20]. Le principal objectif de ces mesures était d'évaluer la QdS fournie à un flux UDP de charge variable, servi en AS (resp. GS), en présence d'un flux BE dont la charge allait en s'accroissant jusqu'à saturation complète du réseau. Les résultats de ces mesures ont permis de conclure que les QdS AS et GS étaient conformes à celles attendues, à savoir un impact nul du trafic BE sur la QdS GS (délai minimum, moyen, maximum, fiabilité et débit moyen inchangés), et « acceptable » sur la QdS AS (délai minimum et moyen, fiabilité et débit moyen inchangés, le délai maximum étant variable).

Partant de ces premiers résultats, les expérimentations menées dans la campagne de mesures ici présentée ont pour pour objectifs :
– d'étudier l'impact des différences de charge entre flux AS (resp. GS) concurrents sur la QdS AS (resp. GS) lorsque le réseau est saturé par du trafic BE (le réseau est donc en état de congestion) ;
– d'étudier l'impact du nombre de flux AS et/ou GS sur la QdS AS (resp. GS) lorsque le réseau est saturé par du trafic BE ;
– de discuter ensuite la possibilité de caractériser un service AS tel que celui développé dans @IRS.

Faute de place, nous ne présentons dans cet article que les mesures relatives à l'impact du nombre de flux. Notons cependant que les tests réalisés en faisant varier les charges respectives de 2 flux AS (resp. GS) concurrents dans un réseau saturé par du trafic BE nous ont conduit à observer une QdS similaire pour chacun des flux. En d'autres termes, l'impact des différences de charge entre flux AS (resp. GS) sur la QdS AS (resp. GS) s'est avéré très faible.

Nous détaillons maintenant les mesures visant à évaluer l'impact du nombre de flux AS et/ou GS sur la QdS AS (resp. GS). Nous présentons tout d'abord la configuration de la plate-forme @IRS, puis les scénarios expérimentaux, et enfin les résultats des mesures et leur analyse.

1 EF : Expedited Forwarding, AF : Assured Forwarding, DE : Discard Eligibility.

1. Configuration de la plate forme @IRS

Les mesures présentées ci-après ont été réalisées entre le LAAS-CNRS (Toulouse) et le LIP6 (Paris) dans l'environnement IPv6 illustré Figure 3 représentant une partie de la plate forme @IRS.

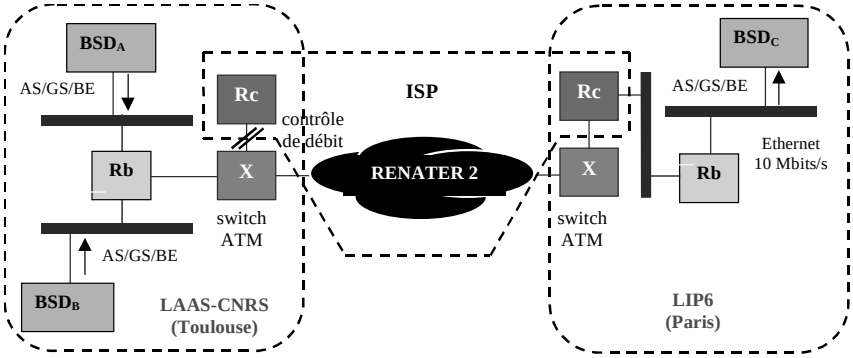

Figure 3. Configuration de la plate forme d'expérimentation

Les plates formes locales sont connectées par le biais d'un routeur dit de bordure (R_b) à l'ISP (*Internet Service Provider*) que représente la plate-forme nationale ATM RENATER 2. Quatre routeurs de cœur (R_c) sont introduits dans l'ISP. Physiquement, ils sont situés sur les plates-formes locales mais logiquement, ils appartiennent l'ISP.

Par le biais de son routeur de bordure, chaque site dispose donc d'un point d'accès à l'ISP caractérisé par un contrat de trafic établi statiquement, équivalent du SLA (*Service Level Agreement*) défini dans [9]. Il est de la responsabilité du routeur de bordure de respecter ce SLA quand il introduit un flux applicatif dans l'ISP. Pour chaque classe de service, ce SLA consiste en :

– les règles de classification et de marquage des paquets ;
– un profil de trafic pour tout le trafic AS et GS correspondant au TCA (*Traffic Conditioning Agreement*) de [9] : pour @IRS, le modèle de trafic retenu est celui du sceau à jetons (*Token Bucket*).
– les actions à entreprendre en cas de non respect du trafic par un flux.

La bande passante du lien connectant les sites à l'ISP (VP ATM de type CBR) est telle que le débit maximal disponible au niveau UDP est de 107 Koctets/s pour des paquets de 1 Koctets (1024 octets). Dans la suite de l'article, nous désignons ce débit par *bande passante du lien* (BPL).

Les routeurs de cœur et de bordure ont été configurés sous les hypothèses suivantes :

– la quantité maximale QM_{GS} de trafic GS que peut introduire le routeur de bordure (en moyenne) a été fixée à 20 Koctets/s, soit environ 20% de BPL ;
– la quantité maximale QM_{AS} de trafic AS que peut introduire le routeur de bordure (en moyenne) a été fixée à 40 Koctets /s, soit environ 40% de BPL ;
– les poids du WFQ appliqué entre paquets AF et DE sont de 0,5 et 0,5 ;
– le contrôle de débit appliqué par les routeurs de cœur est de 100 Koctets/s.

2. Scénarios expérimentaux

Trois scénarios ont été définis :

– le premier a pour but d'évaluer l'impact du nombre de flux AS sur la QdS AS quand le réseau est saturé par du trafic BE. Aucun flux GS n'est généré dans le réseau ;
– le second scénario a pour but d'évaluer l'impact du nombre de flux GS sur la QdS GS quand le réseau est saturé par du trafic BE. Aucun flux AS n'est généré dans le réseau ;
– le troisième scénario a pour but d'évaluer l'impact du nombre de flux AS (resp. GS) sur la QdS GS (resp. AS) quand le réseau est saturé par du trafic BE. Ici, les flux GS et AS sont générés conjointement.

Les paramètres mesurés sont le délai de transit de bout en bout et le taux de perte. Plus précisément :

– les valeurs minimale, moyenne et maximale du délai de transit sont calculées pour des sessions expérimentales d'environ 300 secondes chacune ; la distribution du délai de transit est également évaluée ;

– le taux de perte correspond au ratio {nombre de paquets non reçus / nombre de paquets émis} pour une session complète.

1) *Scénario 1 (resp. scénario 2).*

Pour ces mesures (voir Tableau 1) :

– les flux AS (resp. GS) sont émis depuis les PC BSD_A et BSD_B à destination du PC BSD_C. Trois cas de figure sont considérés :

- 1 flux AS (resp. GS) est généré par le PC BSD_A avec un débit moyen équivalent à 50% de la quantité QM_{AS} (resp. QM_{GS}) autorisée pour le trafic AS (resp. GS) ;
- 2 flux AS (resp. GS) sont générés par les PC BSD_A et BSD_B avec un débit moyen équivalent à 23 et 27% de QM_{AS} (resp. QM_{GS}) ;
- 4 flux AS (resp. GS) sont générés par les PC BSD_A et BSD_B avec un débit moyen équivalent à 11, 12, 13 et 14% de QM_{AS} (resp. QM_{GS}) ;

– parallèlement, 1 flux BE (pour le premier cas) et 2 flux BE (pour les deux autres cas) sont générés par les PC BSD_A et BSD_B. La somme de leur débit moyen est de 100 Koctets/s, soit environ la totalité de BPL.

Scénario 1 (resp. scénario 2)	% de MQ_{AS}	Débit du trafic BE (% de BPL)
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_B) AS_{21} (BSD_A) AS_{22} (BSD_B)	11 12 13 14	100 (50% BSD_A - 50% BSD_B)

Tableau 1. Spécification du trafic pour le scénario 1 (remplacer AS par GS pour le scénario 2)

2) *Scénario 3*.

Pour ces mesures (voir Tableau 2) :

– les flux AS (resp. GS) sont émis depuis les PC BSD_A et BSD_B à destination du PC BSD_C. Deux cas de figure sont considérés :

- 1 flux AS est généré par le PC BSD_A avec un débit moyen correspondant à 100% de la quantité QM_{AS} ; parallèlement, 1 flux GS est généré par le PC BSD_B avec un débit moyen équivalent à 100% de QM_{GS} ;
- 2 flux AS sont générés par les PC BSD_A et BSD_B avec un débit moyen correspondant chacun à 50% de QM_{AS} ; parallèlement, 2 flux GS sont générés par les PC BSD_A et BSD_B ayant chacun un débit moyen équivalent à 50% de QM_{GS}.

– parallèlement, 2 flux BE sont générés par les PC BSD_A et BSD_B. La somme de leur débit moyen est de 100 Koctets/s, soit environ la totalité de BPL.

Scénario 3	% de $MQ_{AS\ ou\ GS}$	Débit du trafic BE (% de BPL)
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)

Tableau 2. Spécification du trafic pour le scénario 3

Notons enfin que :

– tous les flux sont générés au moyen d'un logiciel nommé *Débit6* (développé au LAAS et au LIP6), permettant d'envoyer un trafic UDP dont le profil respecte un modèle de type *Token Bucket* ;

– le débit et le taux de perte pour une session, et le délai de transit pour chaque paquet, sont les informations collectées par *Débit6* en réception ;

– tous les flux sont générés par rafale de 1 paquet UDP de taille égale à 1 Koctets. Le délai inter-paquet est le paramètre utilisé pour ajuster le débit d'un flux ;
– pour toutes les mesures, les flux AS et GS respectent leur profil de trafic ;
– les hôtes sont synchronisés par NTP (*Network Time Protocol*), induisant une incertitude de +/- 5 ms sur les mesures du délai.

3. Résultats et analyses

Les résultats des mesure sont donnés au moyen :
– de la fonction de répartition du délai de transit, soit une courbe représentant sur l'axe des ordonnées le pourcentage de paquets reçus avec un délai de transit inférieur ou égal à la valeur correspondante de l'axe des abscisses ;
– du tableau indiquant le taux de pertes et les valeurs minimale, maximale et moyenne du délai de transit pour chaque flux.

1) *Scénario 1 : {AS * 1, 2, 4 flux} vs. BE (100% de BPL)*

L'impact du nombre de flux AS sur la QdS AS est faible. En effet :
– le Tableau 3 indique une variation inférieure à 5 ms sur la valeur moyenne du délai de transit ;
– ce résultat est conforté par la Figure 4 qui met en évidence que le délai de transit est quasi inchangé pour 90% des paquets. On peut cependant remarquer que 10% des paquets (pour les flux AS1, 2, 11, 12, 21, 22) ont un délai de transit nettement plus important que celui observé pour le flux AS seul. Une première explication a été d'associer ce résultat à l'asynchronisme du système d'exploitation des PC (Free BSD) ; ce phénomène ne se répétant pas pour les mesures en GS (voir résultats du scénario 2), cette explication ne semble pas valide. A l'heure actuelle, aucune explication plausible n'a été formulée ;
– remarquons enfin que le taux de perte est inchangé (nul).

Note : la courbe de la Figure 4 nommée *reference flow* a été obtenue pour un flux AS seul dans un réseau (sans trafic BE ni GS).

Délai (ms)	AS	AS1	AS2	AS11	AS12	AS21	AS22
- min	25	18	18	20	18	18	20
- moyen	38	38	37	42	42	40	41
- max	49	59	63	86	75	65	77
% de perte	0	0	0	0	0	0	0

Tableau 3. Résultats du scénario 1

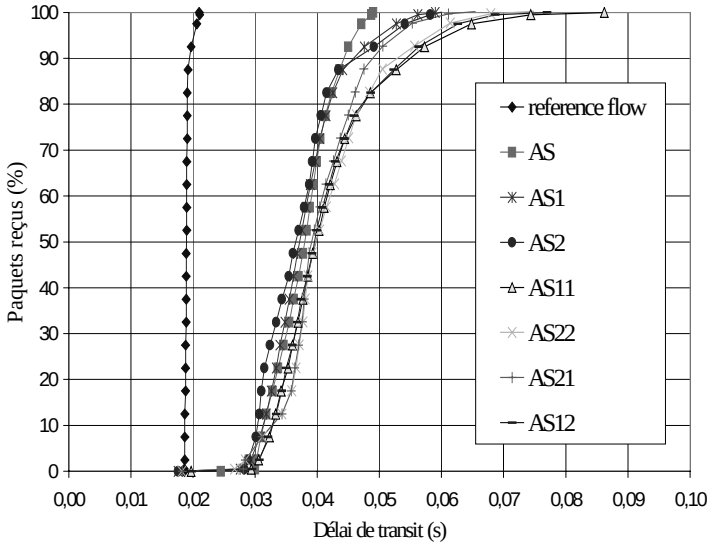

Figure 4. Résultats du scénario 1

2) *Scénario 2 : {GS * 1, 2, 4 flux} vs. BE (100% de BPL)*

L'impact du nombre de flux GS sur la QdS GS est faible. En effet :

– le Tableau 4 indique une variation inférieure à 8 ms sur la valeur moyenne du délai de transit ;
– ce résultat est conforté par la Figure 5 pour 100% des paquets ;
– remarquons là encore, le taux de perte est inchangé (nul).

Notons que la différence maximale sur le délai de transit (20 ms) est acceptable et explicable. En effet, en conservant à l'esprit que : (1) le délai de transit GS doit être le plus petit possible avec une gigue éventuelle correspondant à la présence d'un paquet en file d'attente, (2) l'émission d'un paquet BE ne peut pas être interrompue, et (3) tous les paquets ont une taille de 1 Koctets et sont émis avec un débit maximal de 100 Koctets/s, il en résulte un écart de 20 ms entre le meilleur et le pire cas.

Délai (ms)	GS	GS1	GS2	GS11	GS12	GS21	GS22
- min	19	16	16	18	18	18	18
- moyen	25	26	26	32	31	33	31
- max	33	33	35	33	34	37	38
% de perte	0	0	0	0	0	0	0

Tableau 4. Résultats du scénario 2

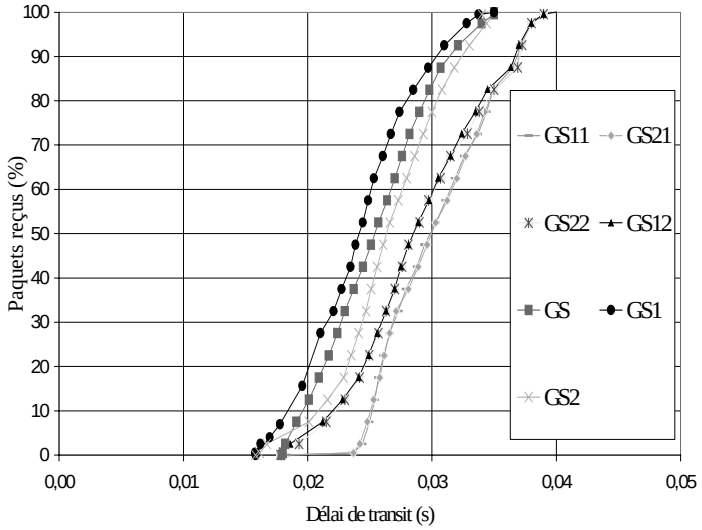

Figure 5. Résultats du scénario 2

3) *Scénario 3 : {AS&GS * 1, 2 flux} vs. BE (100% de BPL)*

L'impact du nombre de flux GS (resp. AS) sur la QdS AS (resp. GS) est presque nul. En effet :

– le Tableau 5 indique une variation inférieure à 6 ms pour AS et 2 ms pour GS sur la valeur moyenne du délai de transit ;
– ce résultat est conforté par la Figure 6 pour 100% des paquets ;
– là encore, le taux de perte est inchangé (nul).

Notons que le délai de transit est quasiment le même que celui observé pour le flux AS seul (resp. GS seul) des Tableau 3 et Figure 4 (resp. Tableau 4 et Figure 5).

Délai (ms)	AS	GS	AS1	AS2	GS1	GS2
- min	19	18	17	17	17	18
- moyen	42	26	47	48	27	28
- max	61	42	78	88	41	40
% de perte	0	0	0	0	0	0

Tableau 5. Résultats du scénario 3

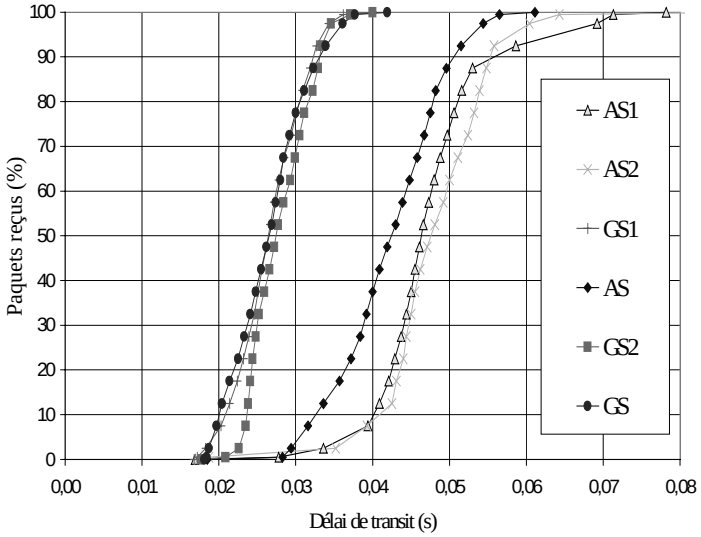

Figure 6. Résultats du scénario 3

IV. Conclusions et travaux futurs

Contribuant à la recherche dans le domaine DiffServ, les travaux présentés dans cet article adressent le problème de la QdS et portent sur la conception, l'implémentation et la mesure des performances d'une architecture de communication à QdS garantie, supportant des services différenciés au niveau IP et une QdS par flux de bout en bout. Les principes de conception de l'architecture et ses services ont été exposés en section II ; l'implémentation de cette architecture sur la plate-forme nationale ATM RENATER 2 a également été décrite dans cette section. Une campagne de mesures de la QdS offerte au niveau utilisateur a été exposée et analysée en section III.

Plusieurs conclusions peuvent être établies ; elles étendent celles données dans [20], qui étaient les suivantes :

– une architecture à services différenciés au niveau IP peut être facilement déployée dans un environnement de type VPN (*Virtual Private Network)* tel que celui de @IRS ;

– en présence de trafic BE dont la charge croît jusqu'à saturer le réseau, la QdS d'un flux UDP servi en AS (resp. GS) est conforme à celle attendue.

Les mesures présentées dans cet article permettent d'apporter les conclusions supplémentaires suivantes. Pour des flux (AS ou GS) respectant leur profil de trafic :

– l'impact du nombre de flux GS sur les QdS AS ou GS est faible ;

– l'impact du nombre de flux AS sur la QdS GS est faible ;

– l'impact du nombre de flux AS sur la QdS AS est faible mais doit être discuté davantage. En effet, si ce résultat est vrai pour 90% des paquets, 10% subissent un délai nettement accru. Ce résultat, non expliqué à l'heure actuelle, est cependant acceptable au regard de la spécification du service AS ; en outre, il est particulièrement important pour la caractérisation d'un service de type AS dans une plate-forme DiffServ telle que celle du projet @IRS. En effet, un impact trop important aurait rendu impossible ou très difficile une telle caractérisation. Notons cependant que l'impact des paquets AS « OUT » sur le délai de transit des paquets « IN » n'a pas été évalué.

Rappelons également que des mesures complémentaires (non présentées dans cet article faute de place) ont permis de conclure que pour des flux AS ou GS respectant leur profil de trafic, l'impact de la charge d'un flux AS (resp. GS) sur la QdS AS (resp. GS) était quasiment nul.

Plusieurs perspectives de ces travaux sont actuellement en cours de développement

– la première consiste à évaluer (au moyen du simulateur NS) l'impact des paramètres de niveau IP (taille des files des routeurs, poids des WFQ, etc.) sur la QdS ; en effet, la valeur exacte des paramètres mesurés en dépend en grande partie ;

– la seconde perspective est de formaliser les sémantiques de garantie associées aux paramètres de QdS, de façon à établir un lien entre la relative imprécision du service AS et la marge d'erreur qu'une application peut tolérer pour certains de ses flux ;
– la troisième perspective est de développer un mécanisme (activé par l'API) permettant au programmeur d'une application de faire abstraction du choix des services de niveaux IP et Transport lors de l'accès au système de communication. La motivation sous-jacente à cette proposition est que le système de communication actuel nous semble difficile à utiliser pour un programmeur n'ayant pas de solides compétences en réseau. Ce mécanisme de sélection est basé sur une caractérisation *a priori* des services GS et AS (fonction de répartition du délai de transit et pourcentage de perte), dont nous faisons l'hypothèse qu'elle est envisageable entre deux sites, moyennant un routage faiblement dynamique. Cette hypothèse, que nous souhaitons soutenir de façon plus exhaustive en simulation, est confortée par les résultats des mesures effectuées sur la plate-forme @IRS ;
– plus prospective, la dernière perspective est d'étendre ces travaux à un contexte multi-domaines.

V.Références

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

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

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

[4] M. Campanella, T. Ferrari, S. Leinen, R. Sabatino, V. Reijs "Specification and implementation 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] A. Campbell, G. Coulson, D. Hutchinson, "A QoS architecture", ACM Computer Communication Review, 1994.

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

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

[12] C. Chassot, A. Lozes, M. Diaz, "From the partial order concept to partial order multimedia connection", JHSN, vol 5, n°2, 1996.

[13] P. Amer, C. Chassot, C. Connolly, P. Conrad, M. Diaz, "Partial order Transport service for multimedia and other applications", IEEE/ACM Transaction on Networking, vol.2, n°5, 1994.

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

[15] C. Chassot, F. Garcia, A. Lozes, P. Anelli, T. Bonald, "Architecture de QoS en environnement IPv6 à services différenciés", CFIP'00, Toulouse, France, Octobre 2000.

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

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

[18] T. Ziegler, S. Fdida, C. Brandauer, B. Hechenleitner. "Stability of RED with two-way TCP Traffic", ICCCN, Las Vegas, Oct 2000.

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

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

[21] F. Garcia, C. Chassot, A. Lozes, M. Diaz, P. Anelli, E. Lochin, "Conception, implementation and evaluation of a QoS based architecture for an IP environment supporting differentiated services", IDMS'2001, Lancaster, UK, September 2001.