

---

# Une nouvelle architecture pour la différentiation de services dans l'Internet basée sur le contrôle de congestion

Philippe Owezarski, Célia Martinie

LAAS-CNRS

7, Avenue du Colonel Roche

F-31077 Toulouse cedex 4

e-mail : owe@laas.fr

---

**RÉSUMÉ.** Cet article aborde le problème de la QoS de bout en bout et de la différenciation de services dans l'Internet. Aujourd'hui, plusieurs techniques existent pour optimiser et garantir la QoS au niveau IP : les VPN, le sur-dimensionnement des capacités et DiffServ qui sont les 3 approches les plus significatives. Toutefois, la QoS perçue par l'utilisateur est très différente de celle fournie par IP. En effet, la couche transport (TCP la plupart du temps) induit des phénomènes d'auto-similarité, à cause, en particulier, de ses mécanismes de contrôle de congestion chaotiques. De fait, une nouvelle approche pour assurer la différenciation de services au niveau transport est proposée. Cette approche repose sur l'agressivité des mécanismes de contrôle de congestion, car plus un protocole est agressif, plus la QoS qu'il fournit est bonne, dans le cas d'un réseau bien dimensionné. Cette thèse est démontrée en prenant en exemple les protocoles de transport UDP et plusieurs versions du protocole TCP.

**ABSTRACT.** This paper deals with the issue on end to end QoS and service differentiation in the Internet. Today, several techniques exist for optimizing and guaranteeing IP QoS: VPN, resource over-provisioning, and DiffServ that are the 3 most significant approaches. But QoS seen by users is different from the one provided by IP. In fact, transport protocols (TCP most of the time), induce self-similarity, in particular because of their chaotic congestion control mechanisms. Then, a new approach for guaranteeing service differentiation at transport level is proposed. This approach relies on congestion control mechanisms aggressiveness, as the more aggressive the protocol, the better the QoS in the case of a well provisioned network. This point of view is demonstrated using several transport protocols examples as UDP and several versions of TCP.

**MOTS-CLÉS :** QoS Internet, différenciation de services, contrôle de congestion, TCP

**KEYWORDS.** Internet QoS, Services differentiation, TCP congestion control

---

## 1. Introduction

S'il est un problème qui ne se dément pas, c'est bien celui de la qualité de service (QoS) dans l'Internet, même si ce terme peut avoir différentes connotations selon la personne qui l'emploie. Pour simplifier un peu la problématique, on peut considérer qu'il existe deux points de vues principaux :

- Le point de vue des applications (ou des utilisateurs) : chaque application a des besoins en termes de débit, délai, gigue, fiabilité, etc. Ces besoins sont naturellement différents d'une application à l'autre, et chaque application souhaiterait pouvoir bénéficier d'un service de communication spécifiquement adapté à ses besoins ;
- Le point de vue des opérateurs dont les objectifs sont d'optimiser l'utilisation des ressources de l'infrastructure de communication (et ainsi de maximiser leurs gains), de limiter les pertes et les délais, et de pouvoir facturer de façon juste et cohérente les services rendus aux utilisateurs.

Un des problèmes les plus délicats à régler est d'arriver à faire converger les besoins des utilisateurs et des opérateurs vers une solution unique. Les solutions proposées jusqu'à présent se sont toutes basées sur la définition d'un nombre limité de classes de services garanties (ATM, IntServ [BRA94] qui s'est avéré inadapté à un réseau de grande taille, DiffServ [BLA98], etc.).

La solution la plus étudiée – aujourd'hui – pour fournir de la QoS dans l'Internet est DiffServ, qui a elle-même succédé à IntServ. Cette solution qui canalise beaucoup d'énergie en recherche, développement et déploiement n'est pas encore une réalité, et au contraire, elle a été dépassée sur le terrain par des approches techniquement plus simples et très efficaces. Une de ces approches est celle des CDN<sup>1</sup> (Akamai par exemple) qui consiste à créer une infrastructure de communication ayant un domaine unique et donc sans liens de « peering » qui sont les sources principales de dégradation de la QoS dans l'Internet. Une seconde approche est la solution du sur-dimensionnement des ressources de communication (liens, routeurs, etc. et ceci sur chaque domaine administratif – ou AS<sup>2</sup> – et sur les liens d'interconnexion inter-AS ou liens de « peering ») pour assurer que les ressources requises par les utilisateurs seront toujours disponibles et ainsi assurer une bonne QoS. Certes, ces deux approches ne permettent pas de différencier les services et ont seulement pour objectif d'améliorer la qualité du service sans garantie ou « best effort ». Cela dit, un des objectifs avoués des grands opérateurs est d'offrir à leurs clients 100% du meilleur service possible que l'on appellera *premium* dans la suite<sup>3</sup>. Face à ces deux approches, le modèle DiffServ présente un certain

---

<sup>1</sup> CDN : Content Delivery Networks.

<sup>2</sup> AS : Autonomous system. Cela correspond à un réseau autonome administré par la même entité, comme le réseau d'un opérateur par exemple, ou un réseau d'université.

<sup>3</sup> Fournir 100% de service premium présente plusieurs avantages pour les utilisateurs. D'abord, la quantité d'applications demandant des QoS élevées, et la quantité de trafic prioritaire qu'elles génèrent, n'est pas bornée. Ceci s'oppose par exemple à l'approche DiffServ pour laquelle les simulations montrent que le service EF (équivalent à notre service premium) ne peut guère être garanti s'il représente plus de 5% du

nombre de lacunes justifiant sans doute la lenteur de sa mise en œuvre réelle. Ces défaillances semblent de deux ordres :

- D'abord, les classes prédéfinies de DiffServ semblent un peu « grossières » par rapport aux besoins des applications, et surtout mal définies, voir mal adaptées à leurs besoins. DiffServ reste une approche orientée opérateur (alors que IntServ, par opposition, était plus orientée application).
- D'autre part, les accords entre opérateurs pour le dimensionnement des différents AS – afin de pouvoir mieux définir la QoS qui sera offerte par chaque classe de service – ne sont pas aisés à trouver, en particulier au niveau des liens inter-AS (problème de la QoS inter-domaines). Ils se heurtent à des problèmes économiques et politiques. En effet, il est difficile d'établir qui possède le lien d'interconnexion, qui finance son installation, et surtout qui en tire le plus de bénéfices (financiers essentiellement) ?

Comme il n'est pas encore évident de savoir ce que sera la solution retenue pour garantir la QoS au niveau IP dans l'Internet, cet article propose une nouvelle approche pour permettre la différenciation de services au niveau transport, de bout en bout, qui tienne compte de la consommation des ressources par chaque flux (et qui soit indépendante de la stratégie choisie pour garantir la QoS au niveau IP). Dans l'Internet, la répartition des ressources disponibles entre les flux (ou connexions) à un instant  $t$  est faite par l'intermédiaire des mécanismes de contrôle de congestion des protocoles de transport. Aussi, cet article propose une nouvelle architecture pour la différenciation de services dans l'Internet basée sur des classes de mécanismes de contrôle de congestion plus ou moins « agressifs<sup>4</sup> » les uns par rapport aux autres.

La suite de cet article adopte le plan suivant : d'abord, la partie 2 montre que garantir la QoS au niveau IP ne suffit pas pour offrir des services garantis aux applications des utilisateurs à cause de la présence d'une couche de transport au comportement irrégulier. Cette partie présente aussi les justifications de l'approche proposée dans cet article, à savoir la différenciation de services de bout en bout en fonction de classes de contrôle de congestion. Dans la partie 3, l'article décrit les expérimentations qui ont été réalisées et qui ont démontré par l'exemple que la différenciation des services par la différenciation de classes de mécanismes de contrôle de congestion était possible et très cohérente. Enfin, la partie 4 conclut ce papier, tout en soulevant les bienfaits de cette approche et dresse une liste de prospectives à ce travail.

## 2. Une approche basée sur le contrôle de congestion

L'approche DiffServ s'avère ainsi assez peu adaptée aux besoins des applications, avec des services pas assez spécialisés. D'autre part, assurer une QoS

---

trafic total [SAH99]. D'autre part, même le trafic élastique – comme le trafic web par exemple – tire grand profit d'un meilleur service réseau offrant plus de bande passante, ainsi que des délais et des taux de pertes bas. Cela permet de meilleurs services élastiques, et surtout une meilleure utilisation des ressources du réseau.

<sup>4</sup> Le terme « d'agressivité » sera défini plus précisément dans la partie 3.

garantie de bout en bout avec DiffServ présente des difficultés non seulement techniques, mais surtout politico-administratives entre les différents administrateurs / opérateurs des différents AS de l'Internet. Ces raisons font certainement partie de celles qui vont à l'encontre d'un déploiement général de DiffServ sur l'Internet global.

D'autre part, il est évident que la couche transport (la plupart du temps TCP dans l'Internet) est aussi problématique car elle modifie les services réseaux perçus par les applications et leurs utilisateurs. En effet, des études météorologiques ont montré que TCP engendrait des modèles de trafics complexes – très éloignés des classiques modèles de Poisson pour les arrivées de flux ou de paquets [PAX95] et du modèle de Gilbert pour les pertes [VER01]. Par exemple, [UHL01] [CRO97] [LEL94] [ERR96] ont montré que le trafic Internet présentait des propriétés d'auto-similarité. Plus encore, [VER00] a montré que les mécanismes de TCP étaient chaotiques. Finalement, que le trafic soit auto-similaire ou chaotiques, cela se traduit par une grande variabilité, largement dominante par rapport à la moyenne [WOL89], et donc par un service final vu par les applications et les utilisateurs très changeant, et évidemment non garanti. L'effet de la garantie de QoS au niveau 3 par DiffServ est donc totalement détruit par les mécanismes de TCP.

Un protocole de transport reste néanmoins indispensable pour pallier aux problèmes de pertes de la couche IP. Le transport étant également en charge d'assurer la QoS de bout en bout, intervenir au niveau de cette couche apparaît architecturalement cohérent. Par ailleurs, on est complètement convaincu qu'une différenciation de services doit se faire au niveau transport, et plus particulièrement au niveau des mécanismes de contrôle de congestion, car ce sont ces mécanismes de contrôle de congestion qui sont responsables de l'aspect chaotique de la QoS offerte par TCP aux applications des utilisateurs [VER00].

Les mécanismes de contrôle de congestion régissent aussi les processus d'acquisition et de consommation des ressources de communication. D'ailleurs, au fil des ans, les différentes versions de TCP (Tahoe [STE94], Reno [FAL96], New Reno [FLO99], SACK [MAT99]) se sont différenciées par rapport aux 4 mécanismes de contrôle de congestion, à savoir [JAC98] [ALL99] :

- « Slow-start »;
- « Congestion avoidance »;
- « Fast retransmit »;
- « Fast Recovery ».

Le processus d'évolution de TCP, permettant de constamment améliorer les performances moyennes de ce protocole, a consisté à augmenter son « agressivité » par rapport aux ressources disponibles (tout en restant équitable) [FAL96]. On définit ici « l'agressivité » d'un mécanisme protocolaire comme son aptitude à consommer des ressources. L'idée est ainsi de proposer des classes de services transport basées sur l'agressivité de leurs mécanismes de contrôle de congestion. En effet, on peut penser que plus un protocole consomme des ressources, plus la QoS qu'il offre à son application est élevée. La suite va montrer que cette affirmation est vraie et que des agressivités différentes des mécanismes de contrôle de congestion amènent à des QoS différenciées.

Cette preuve sera faite par l'exemple en ayant choisi 3 protocoles de transport significatifs<sup>5</sup> :

- UDP qui ne présente pas de mécanisme de contrôle de congestion et a donc une agressivité forte. D'ailleurs, UDP est utilisé pour de nombreuses applications temps réel et est pratiquement la seule alternative à TCP aujourd'hui ;
- TCP Tahoe, une ancienne version de TCP réputée peu agressive. Cette version a presque disparu mais elle est utilisée ici pour maximiser les différences de comportement, permettant ainsi de mieux illustrer les propos de cet article ;
- TCP New Reno qui est à l'heure actuelle la version de TCP la plus utilisée sur Internet.<sup>6</sup>

Pour faire la démonstration que la différenciation de services sur ces mécanismes de contrôle de congestion est possible, il sera montré que :

- Les priorités (l'ordre des agressivités) est respecté ;
- Les flux d'une même classe d'agressivité se partagent les ressources équitablement.

### 3. Expérimentations et démonstration par l'exemple

#### 3.1. Description de la plate-forme expérimentale

Cette étude sur la différenciation par les mécanismes de contrôle de congestion du transport a été faite grâce au simulateur de réseaux NS 2 (Network Simulator), le simulateur officiel de l'IETF.

Dans l'optique de la différenciation par les protocoles de transport, nous allons attribuer arbitrairement (sans avoir quantifié les services) une classe à chaque protocole. TCP Tahoe peut représenter un service *Best Effort* étant donné que son algorithme d'évitement de congestion en fait le protocole le moins agressif des trois. UDP peut représenter un service Premium, tandis que TCP New Reno représente un service intermédiaire, son algorithme d'évitement de congestion étant plus agressif envers le réseau que celui de TCP Tahoe et moins agressif qu'UDP qui n'effectue aucun contrôle de congestion.

Les mécanismes FCFS (*First Come First Served*) appelé aussi FIFO (*First In First Out*) et *Tail Drop* pour la gestion des files d'attente au niveau des routeurs ont été choisis pour leur neutralité sur le service réseau et aussi car ils sont les plus répandus actuellement.

La topologie du réseau utilisée pour effectuer les simulations est représentée sur la Figure 1 qui donne aussi les caractéristiques des liens. Cette topologie de test

---

<sup>5</sup> En fait cette étude a été réalisée avec tous les protocoles transport de l'Internet, mais pour des raisons de lisibilité, seulement les plus significatifs sont présentés.

<sup>6</sup> Même si l'utilisation de TCP SACK est aujourd'hui recommandée, TCP New Reno reste dominant. A noter que TCP SACK est plus agressif que TCP New Reno et qu'il peut, à ce titre, constituer une classe d'agressivité et donc de QoS.

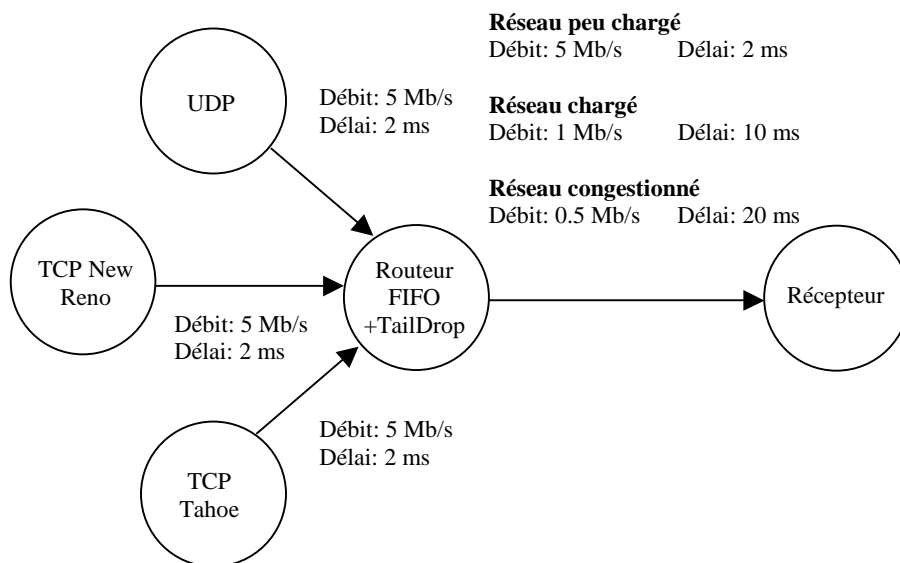
comporte les trois émetteurs UDP, TCP New Reno et TCP Tahoe qui peuvent envoyer un ou plusieurs flux chacun, un nœud intermédiaire de concentration et un nœud récepteur. Le premier nœud sur lequel l'information est dirigée représente un routeur du réseau. Le lien entre le routeur et le récepteur de tous les flux est assimilé au réseau de concentration (réseau d'accès à un Tier 1 par exemple). C'est en modifiant les caractéristiques de ce lien que nous allons pouvoir modifier les niveaux de charge sur ce lien et ainsi démontrer les effets de l'agressivité sur la QoS. Les simulations auront pour but d'étudier le comportement des 3 protocoles de transport dans les cas où le réseau de concentration est :

- Peu chargé (Capacité = 5 Mbps, Délai = 2 ms) ;
- Chargé (Capacité = 1 Mbps, Délai = 10 ms) ;
- Congestionné (Capacité = 0,5 Mbps, Délai = 20 ms)

cette notion étant en fait en rapport avec la charge du trafic UDP.

Les autres liens ont pour caractéristiques : Capacité = 5 Mbps, Délai = 2 ms.

Les applications utilisant TCP New Reno et TCP Tahoe sont de type FTP, celles utilisant UDP envoient les paquets à un débit constant de 448 Kbit/s. Toutes les applications envoient 200 paquets de 1000 octets chacun, et démarrent au même moment.



**Figure 1.** Caractéristiques du réseau simulé

### 3.2. Respect des classes différenciées

Les paramètres analysés pour mettre en évidence les différences d'agressivité et de QoS offertes par chaque protocole transport sont les suivants :

- Le temps d'émission de tous les paquets par l'application :

C'est ici le principal paramètre pour quantifier la QoS. C'est en effet le temps d'émission observé par l'utilisateur. Ce paramètre est inversement proportionnel au débit utile pour l'application utilisatrice. Donc, plus ce temps est court, plus le débit utile est haut et plus la QoS perçue par l'utilisateur est élevée ;

- *Le temps moyen d'attente dans les files d'attente :*  
C'est un paramètre important pour quantifier l'agressivité du protocole. En effet, si un protocole est agressif, il va émettre des paquets sans se soucier de l'état de congestion du réseau et des files d'attente des routeurs, et les paquets risquent (en cas de charge élevée) de rester longtemps dans ces files. Donc, plus ce temps est court, moins le protocole est agressif ;
- *Le débit consommé par la connexion :*  
Par rapport au débit utile qui est un paramètre de QoS présenté plus haut, ce débit prend en compte aussi le trafic induit par les reprises de pertes. C'est donc, par rapport au débit utile, plus un paramètre d'agressivité car les pertes sont généralement causées par des dépassement de la capacité des files dans les routeurs, engendrées par une trop grande agressivité du protocole de transport (pour les mêmes raisons que ci-dessus) ;
- *Le taux de perte :*  
C'est également un paramètre significatif pour quantifier l'agressivité, et pour les mêmes raisons que précédemment, un fort taux de perte est significatif d'une forte agressivité.

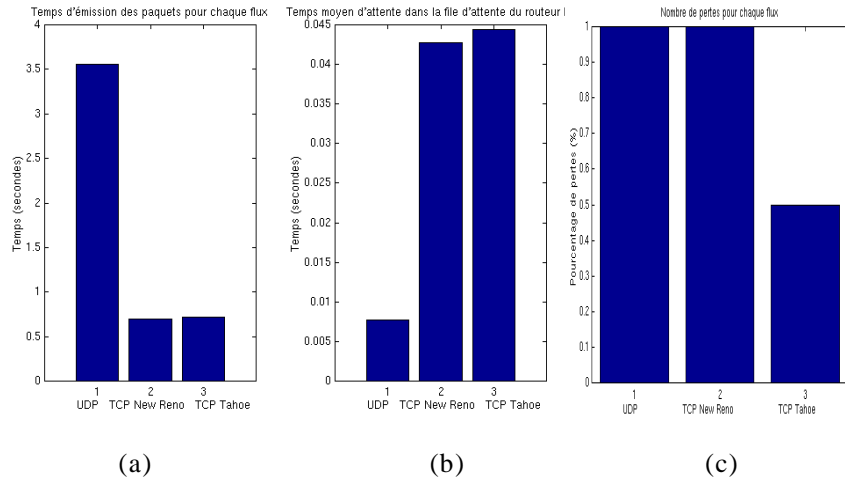
Le tableau 1 dresse le bilan des résultats obtenus :

		Réseau peu chargé	Réseau chargé	Réseau congestionné
<b>Temps d'émission de tous les paquets</b>	UDP	3.55 s	3.88 s	4.24 s
	TCP New Reno	0.69 s	4.28 s	9.83 s
	TCP Tahoe	0.71 s	4.78 s	7.25 s
<b>Temps moyen d'attente dans la file d'attente</b>	UDP	8 ms	277 ms	644 ms
	TCP New Reno	43 ms	258 ms	224 ms
	TCP Tahoe	44 ms	198 ms	250 ms
<b>Bande passante</b>	UDP	448 Kb/s	413 Kb/s	377 Kb/s
	TCP New Reno	2.3 Mb/s	374 Kb/s	163 Kb/s
	TCP Tahoe	2.25 Mb/s	335 Kb/s	221 Kb/s
<b>Nombre de pertes</b>	UDP	1 %	8.5 %	10 %
	TCP New Reno	1 %	2 %	12 %
	TCP Tahoe	0.5 %	6.5 %	10.5 %

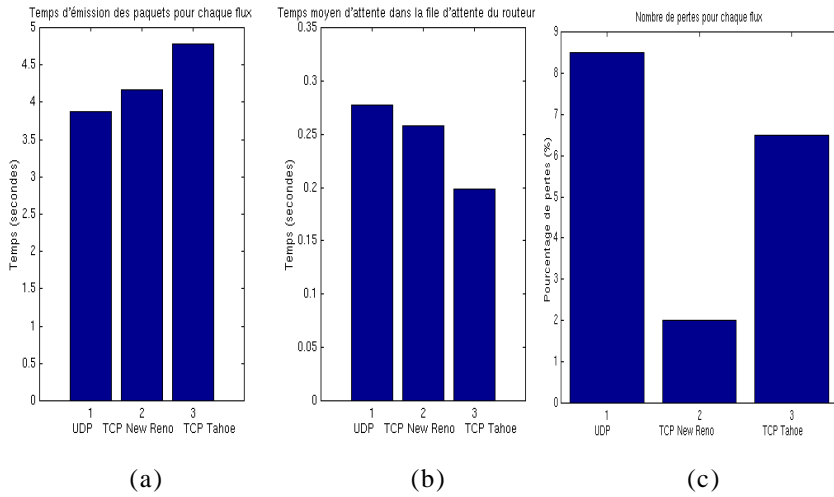
**Tableau 1.** Résultat des simulations

Pour mieux visualiser ces résultats, les figures 2, 3 et 4 présentent ces résultats graphiquement dans les 3 cas considérés pour le réseau : peu chargé, chargé et congestionné. A noter que comme l'on peut aisément obtenir le débit consommé à

partir du débit utile et du taux de perte, les figures ne sont pas surchargées par la mention du débit consommé.

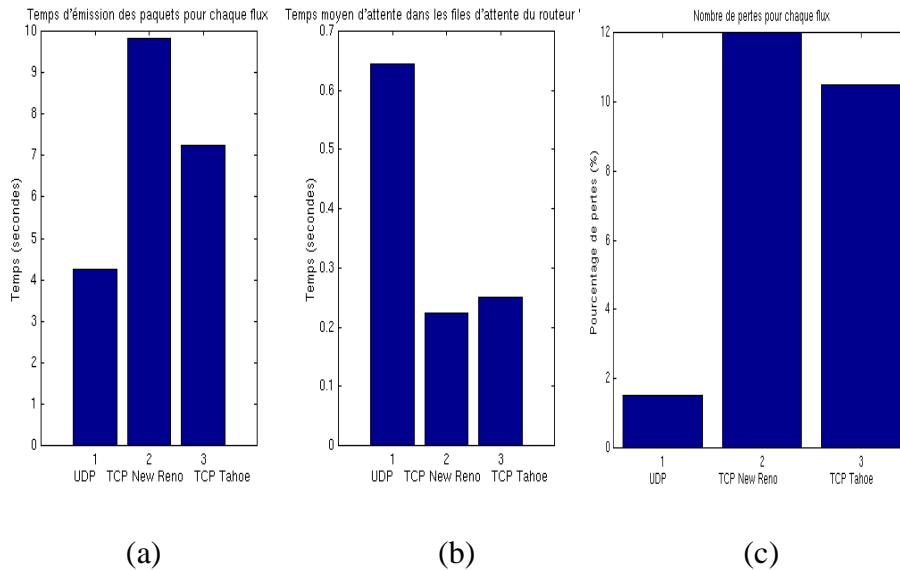


**Figure 2.** Différentiation de services selon l'agressivité des mécanismes de contrôle de congestion dans le cas d'un réseau peu chargé



**Figure 3.** Différentiation de services selon l'agressivité des mécanismes de contrôle de congestion dans le cas d'un réseau chargé





**Figure 4.** Différentiation de services selon l'agressivité des mécanismes de contrôle de congestion dans le cas d'un réseau congestionné

La suite va présenter les conclusions de l'analyse de ces résultats pour les 3 niveaux de charges étudiés.

#### Réseau peu chargé (Figure 2)

Dans ce cas, de façon logique les pertes restent faibles. D'ailleurs, il n'apparaît pas de différence significative entre ces 3 protocoles. Ils sont tous, dans ces conditions, équivalents en terme d'efficacité, même si TCP New Reno reste très légèrement supérieur à TCP Tahoe – comme cela était prévu par notre proposition. UDP, dans ce cas, ne peut pas être comparé aux autres protocoles car son débit est fixé par une source à débit constant. Toutefois, on observe que le temps minimal de transmission du flux UDP est respecté lorsque le réseau est peu chargé ; 200 paquets à 448 Kbit/s sont théoriquement transmis en 3,55 s, ce que l'on vérifie bien.

Enfin, on ne peut pas mentionner, au travers de ces résultats<sup>7</sup>, des différences dans l'agressivité des protocoles car lorsque le protocole demande des ressources il n'a pas à attendre avant qu'elles lui soient attribuées.

<sup>7</sup> Même si les écarts sur la Figure 2(b) semblent graphiquement importants, il faut tout ramener à l'échelle de la mesure. Les valeurs de temps qui sont mesurées sont si basses que les écarts sont en fait

*Réseau chargé (Figure 3)*

Si la charge du réseau augmente, le taux de perte général augmente aussi. Toutefois, la Figure 3 montre un respect total de notre proposition, et QoS et agressivité évoluent bien dans le même sens. Ainsi, les performances d'UDP sont meilleures que celles de TCP New Reno (le temps d'émission total est inférieur et le débit utile supérieur) et elles sont elles-mêmes meilleures que celles de TCP Tahoe – comme prévu par notre proposition. A noter cependant sur la Figure 3(c) que les pertes avec TCP Tahoe sont plus élevées que ce qu'elles devraient être. L'analyse montre qu'en cas de pertes, TCP Tahoe ne détecte la (les) erreur(s) que suite à l'expiration du timer d'émission et passe en *congestion avoidance*, alors que TCP New Reno lance plus tôt une phase de *Fast Recovery*. Ainsi, TCP New Reno ré-émet son (ses) paquet(s) avant TCP Tahoe qui est donc positionné dans la file d'attente après les paquets TCP New Reno [OWE01].

*Réseau congestionné (Figure 4)*

Dans le cas d'un réseau congestionné, la figure 4 montre des entorses à notre théorie QoS / agressivité, mais elles sont effacées par les résultats de la figure 7. On constate sur les deux figures 4 et 7 que le trafic UDP passe sans réelle difficulté (comme prévu par notre théorie). En effet, en cas de congestion, les différentes versions de TCP s'auto-limitant, elles laissent la voie libre à UDP qui ne se soucie pas des problèmes de congestion. Au final, il bénéficie quasiment des ressources dont il a besoin et offre donc un service très convenable à l'utilisateur. Bien sûr, de graves problèmes de congestion peuvent apparaître à cause d'UDP si sa charge augmente trop, et l'intégrité du réseau peut être remise en question pour de longs moments, justifiant l'utilisation d'un contrôle d'admission pour le trafic UDP<sup>8</sup>.

En ce qui concerne TCP Tahoe et New Reno, sur la figure 4, le moins agressif fournit finalement le meilleur service. Ce résultat est dû à la présence du flux UDP qui ne laisse que très peu de ressources libres pour les deux flux TCP<sup>9</sup>. Dans ce cas, l'agressivité de TCP New Reno qui émet des paquets avec moins de certitudes que TCP Tahoe sur l'état des ressources disponibles engendre aussi plus de pertes et une QoS finalement moins bonne. Mais la figure 7 représentant la même expérience mais avec un plus grand nombre de flux, qui réduit donc l'effets des comportements spécifiques (chaotiques) des protocoles, en plus en se plaçant dans un contexte plus réaliste que celui de la figure 4, montre que TCP New Reno fournit bien une QoS supérieure à TCP Tahoe.

Au bilan, notre théorie QoS / agressivité est avérée.

**3.3. Equité des flux d'une même classe**


---

complètement négligeables. Cela montre seulement qu'il n'y a pas de contention importante entre flux grâce au sur-dimensionnement des capacités.

<sup>8</sup> On parle de « TCP friendliness »

<sup>9</sup> On se trouve alors dans le contexte d'un réseau sous-dimensionné.

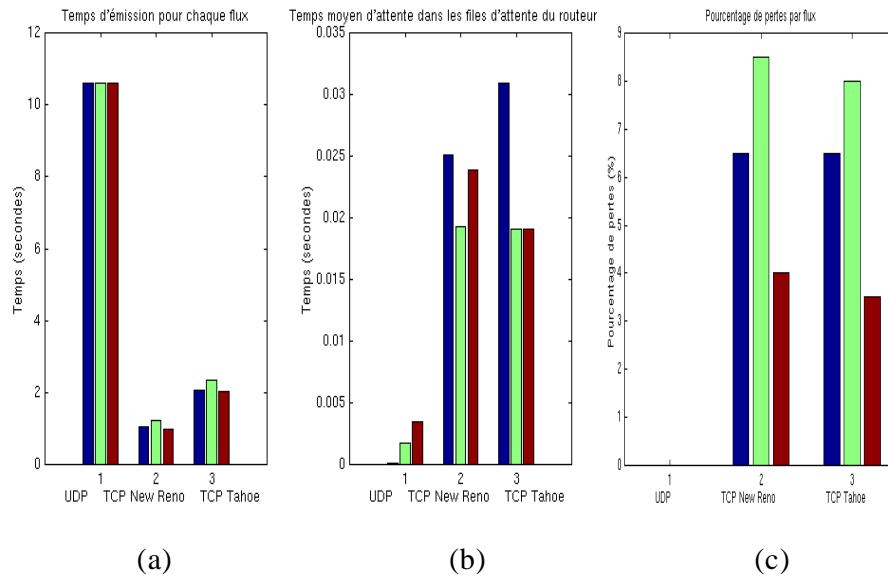


Figure 5. Equité entre flux d'une même classe dans le cas d'un réseau peu chargé

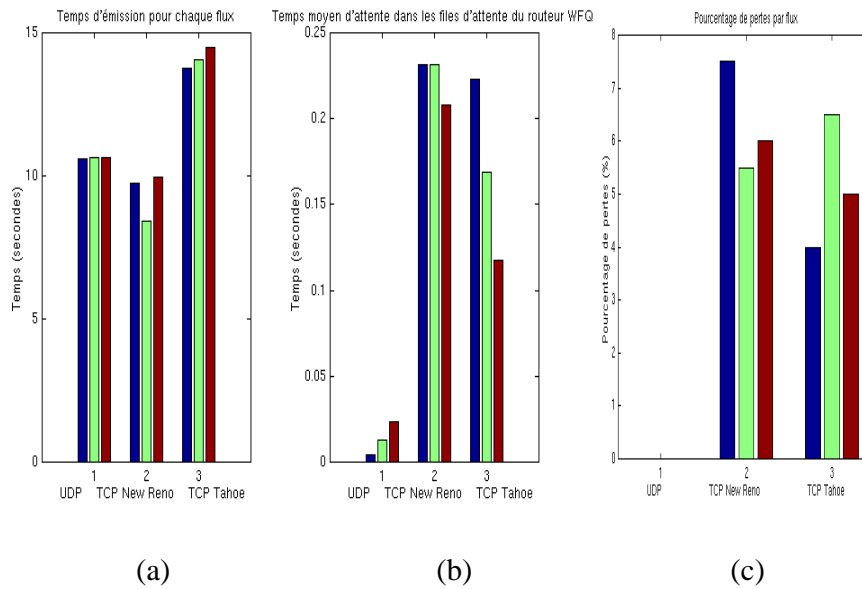
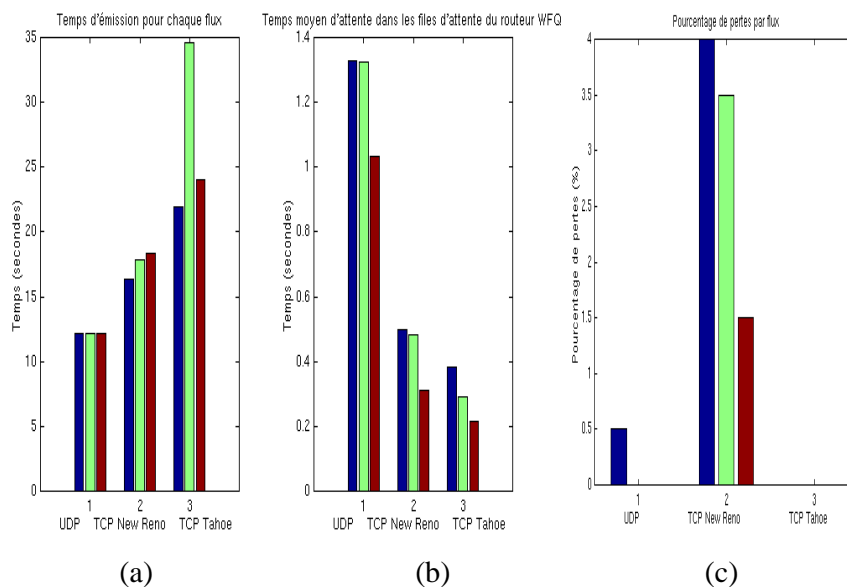


Figure 6. Equité entre flux d'une même classe dans le cas d'un réseau chargé

Dans cette nouvelle simulation, l'objectif est de montrer que si plusieurs flux utilisent le même protocole, soit la même classe de service, alors ils vont bénéficier sensiblement des mêmes ressources et offrir en résultat des QoS équivalentes à leurs applications utilisatrices. Pour cela, trois flux (au lieu d'un) de chaque classe sont générés en parallèle. Les sources UDP émettent à 150 Kb/s. Toutes les applications envoient 200 paquets de 1000 octets chacun et démarrent en même temps. Le temps de transmission minimal de 200 paquets de 1000 octets chacun est donc de 10.61 secondes. Les figures 5, 6 et 7 présentent les résultats obtenus dans les 3 cas de charges du réseau : peu chargé, chargé et congestionné.



**Figure 7.** *Équité entre flux d'une même classe dans le cas d'un réseau congestionné*

Au vu de ces résultats, l'équité au sein de chaque classe est avérée. En effet, dans les différents cas de charges du réseau, les valeurs des paramètres mesurés sont très proches pour un même protocole, c'est-à-dire au sein d'une même classe.

#### 3.4. Effets de l'ordonnement des routeurs sur la différenciation de services

Dans ce travail, nous nous sommes volontairement abstraits des techniques avancées des routeurs : gestion des files d'attente, ordonnancement, seuillage et politiques de pertes, etc. Toutefois, nous avons montré dans [OWE01] que l'on pouvait potentialiser les effets de la différenciation de services par l'agressivité du contrôle de congestion en associant de manière logique les classes de services transport ainsi définies avec les priorités de l'ordonnancement du routeur, par exemple.

#### 4. Conclusions

Cet article a montré la problématique de la QoS dans l'Internet, notamment en prenant le point de vue DiffServ qui reste dominant dans le domaine de la recherche Internet. Il n'est cependant pas le seul, et deux nouveaux points de vue sont aujourd'hui sous les feux de la rampe : les CDN et le sur-dimensionnement des réseaux qui ont la particularité d'être opérationnels et déployés dans l'Internet réel. Quoi qu'il en soit, cet article a montré que les solutions citées sont insuffisantes pour assurer une QoS garantie aux utilisateurs. En effet, la couche transport, et l'aspect chaotique de ses mécanismes de contrôle de congestion, modifie la QoS réseau offerte aux utilisateurs. L'idée défendue est donc de définir des classes de QoS en fonction de l'agressivité des mécanismes de contrôle de congestion de la couche transport. Cette approche a ainsi été démontrée par l'exemple.

Cette approche propose donc comme partie de la solution au problème de la QoS dans l'Internet une nouvelle architecture de différenciation de services basée sur l'agressivité des mécanismes de contrôle de congestion. Cela suppose donc que chaque application, voir chaque flux, pourra utiliser son propre protocole de transport, éventuellement spécifiquement conçu pour cette application ou ce média. A noter, qu'il sera aussi important de concevoir des mécanismes de contrôle de congestion qui ne soient pas chaotiques. C'est là une des directions prospectives de ce travail.

Il est à noter également que cette approche fait totalement abstraction de la technologie réseau utilisée. Ainsi, que le niveau 3 utilise un service de CDN, des ressources sur-dimensionnées pour fournir 100 % de service premium ou DiffServ, notre approche fonctionne parfaitement. A noter toutefois que si DiffServ est utilisé, on peut adapter au mieux les mécanismes d'ordonnancement et de rejet des paquets dans les routeurs pour optimiser la QoS fournie aux utilisateurs [OWE01].

D'autre part, comme l'agressivité et donc la QoS est dépendante de la quantité de ressources consommées, il est a priori possible d'envisager des mécanismes de facturation simples, suivant cette architecture, qui seront très cohérents par rapport à la logique de l'opérateur et la logique de l'utilisateur. Ce papier a d'ailleurs posé les bases théoriques d'une métrique de la QoS basée sur une métrique de l'agressivité pouvant servir à quantifier à titre prévisionnel les ressources qui seront consommées par un flux. Cet aspect est un second point prospectif qui sera étudié prochainement.

#### 5. Références

- [ALL99] ALLMAN, M., PAXSON, V., STEVENS, W., « TCP Congestion Control », RFC 2581, avril 1999
- [BLA98] BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z., WEISS, W., « An Architecture for Differentiated Services », RFC 2475, December 1998
- [BRA94] BRADEN, R., CLARK, D., SHENKER, S., « Integrated Services in the Internet Architecture : An overview », RFC 1633, June 1994

- [CRO97] CROVELLA, M., BESTAVROS, A., « Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes », *IEEE/ATM Trans. on Networking*, December 1997
- [ERR96] ERRAMILI, A., NARAYAN, O., WILLINGER, W., « Experimental queueing analysis with long-range dependent traffic », *IEEE/ACM Trans. on Networking*, April 1996
- [FAL96] FALL, K., FLOYD, S., « Simulation-based Comparisons of Tahoe, Reno, and SACK TCP », *Computer Communication Review*, July 1996
- [FLO99] FLOYD, S., HENDERSON, T., « The New Reno Modification to TCP's Fast Recovery Algorithm », RFC 2582, April 1999
- [JAC98] JACOBSON, V., « Congestion avoidance and control », *proceedings of SIGCOMM'98*, august 1998
- [LEL94] LELAND, W., TAQU, M., WILLINGER, W., WILSON, D., « On the self-similar nature of Ethernet traffic (extended version) », *IEEE/ACM Trans. on Networking*, 1994
- [MAT99] MATHIS, M., MAHDAVI, J., FLOYD, S., ROMANOV, A., « TCP Selective Acknowledgment Options », RFC 2018, October 1999
- [OWE01] OWEZARSKI, P., MARTINIE, C., « Conception et simulation d'une architecture à différenciation de services alternative à DiffServ », Rapport LAAS No. 01447, Octobre 2001
- [PAX95] PAXSON, V., FLOYD, S., « Wide-Area Traffic: The Failure of Poisson Modeling », *IEEE/ACM Trans. on Networking*, vol. 3, no. 3, June 1995
- [SAH99] SAHU, S., TOWSLEY, D., KUROSE, J., « A Quantitative Study of Differentiated Services for the Internet », *Proceedings of IEEE Globecom'99*, December 1999
- [STE94] STEVENS, W.R., « *TCP/IP Illustrated, Volume 1: The Protocols* », Addison-Wesley, 1994
- [UHL01] UHLIG, S., BONAVENTURE, O., « Understanding the long-term self-similarity of Internet traffic », in *Proc. Quality of Future Internet Services workshop (QoFIS'2001)*, 2001
- [VER00] VERES, A., « The Chaotic Nature of TCP Congestion Control », *In proceedings of the INFOCOM conference*, March 2000
- [VER01] VERLHAC, O., « Métrologie des réseaux IP – Etude des pertes dans l'Internet », Rapport de stage d'ingénieur de l'ENSICA, Septembre 2001
- [WOL89] WOLF, R., « *Stochastic modeling and the theory of queues* », Prentice Hall, 1989