Un Algorithme Résistant au Facteur d'Échelle pour le Routage avec Trois Métriques de QoS

Luís Henrique M. K. Costa[1,2]
Serge Fdida[1]
Otto Carlos M. B. Duarte[2]

[1] *LIP6 – Université Pierre et Marie Curie*
4, place Jussieu - 75252 - Paris Cedex 05 - France
Tél. +33 1 44 27 87 72 - Fax +33 1 44 27 74 95
`{Luis.Costa,Serge.Fdida}@lip6.fr`
[2] *Grupo de Teleinformática e Automação – GTA*
Universidade Federal do Rio de Janeiro – COPPE/EE
P.O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brasil
`otto@gta.ufrj.br`

RÉSUMÉ. *Le routage à Qualité de Service (QoS) est capable de fournir des garanties de QoS aux applications multimédias ainsi qu'une utilisation efficace des ressources du réseau. Le routage à QoS peut pourtant être un mécanisme très coûteux qui ne résiste pas au facteur d'échelle. L'algorithme de routage doit donc être le plus simple possible. Ce papier propose et analyse la performance d'un algorithme de routage à QoS basé sur l'état des liens qui prend en compte trois métriques : le délai de propagation, la bande passante disponible et la probabilité de perte. Une heuristique basée sur la probabilité de perte résiduelle et sur la combinaison de métriques est utilisée pour que l'algorithme passe à l'échelle et soit solvable en temps polynômial. Les résultats obtenus par simulation démontrent que notre algorithme est une solution réalisable pour la construction de chemins avec trois paramètres de QoS.*

ABSTRACT. *Quality of Service (QoS) based routing provides QoS guarantees to multimedia applications and an efficient utilization of the network resources. Nevertheless, QoS routing is likely to be a costly process that does not scale when the number of nodes increases. Thus, the routing algorithm must be simple. This paper proposes and analyzes the performance of a link-state QoS routing algorithm that takes into account three metrics: propagation delay, available bandwidth, and loss probability. A heuristic based on the residual loss probability and metric-combination is used to turn the algorithm scalable and solvable in polynomial time. The simulation results show that our algorithm is a feasible solution to construct paths constrained on three metrics.*

MOTS-CLÉS : *Internet, Qualité de Service, Routage*

KEYWORDS: *Internet, Quality of Service, Routing*

1. Introduction

Pour faciliter l'utilisation des applications multimédias dans l'Internet, des nouvelles caractéristiques et des nouveaux services doivent y être ajoutés. Des nouvelles classes de service autres que le service "au mieux" d'IP doivent être créées pour fournir des garanties de Qualité de Service (*Quality of Service* - QoS) aux applications. Un problème important lié à ce sujet de recherche est la définition d'une architecture de routage qui prenne en compte les besoins de QoS des applications [CRA 98].

Dans le modèle de routage classique, le chemin suivi par les paquets est fonction de l'adresse source et destination, l'algorithme de routage utilisant une seule métrique (le nombre de sauts ou un coût) pour choisir entre plusieurs routes. Le routage à QoS est plus complexe car les décisions de routage se basent sur n métriques qui reflètent les besoins de QoS du trafic. La résistance au facteur d'échelle devient donc un problème très important dans la définition d'une telle architecture de routage [CRA 98, APO 98]. L'algorithme de calcul de routes doit être simple. Certains auteurs considèrent l'utilisation des chemins pré calculés pour minimiser le temps d'établissement de connexion dans le routage à QoS [APO 98, TRA 99]. La taille des tables de routage est un autre point important pour ce qui concerne le passage à l'échelle. Le routage doit garder le moins possible d'information d'état dans les routeurs. L'incertitude des paramètres de QoS peut aussi influencer le routage à QoS. Par exemple, dans un protocole basé sur l'état des liens la base de données représentative de la topologie du réseau peut être périmée sur certaines périodes [LOR 98, FAY 99]. L'algorithme de routage pourrait prendre en compte ces incertitudes, ce qui risquerait de le rendre encore plus complexe. Le choix de la période d'envoi des messages de mise à jour est fait en fonction de la précision des paramètres de QoS et de la surcharge du protocole.

Le routage à QoS risque donc d'être un mécanisme très lourd à gérer. Pour cette raison, on doit éviter le maintien d'information d'état par flux dans les routeurs, cette approche ne résiste pas au facteur d'échelle [BRA 94, BLA 98]. Une solution plus convenable serait le maintien d'information par flux uniquement pour un petit nombre de flux (flux qui ont des besoins de QoS très forts et applications qui payent pour un service plus cher). Les flux moins exigeants peuvent se satisfaire d'un service non orienté connexion, où le routage est réalisé selon des classes de service et non pas par flux. Nous croyons que la combinaison d'une approche basée sur des classes de service avec un routage par flux pour une faible quantité de flux peut améliorer l'efficacité du routage. Le travail dans [RAJ 98] traite le problème du routage multipoint avec des réservations de ressources et adopte à peu près la même idée. Le compromis entre la quantité d'information à maintenir et la construction efficace d'arbres multipoint est étudié. Trois approches différentes sont analysées : information par flux pour tous les flux dans tous les routeurs, aucune information par flux, et une solution partielle qui ne contient que les flux plus gourmands en bande passante. Les conclusions montrent que la solution à information partielle peut être envisageable.

Ce papier discute brièvement les différentes solutions proposées dans la littérature pour le routage à QoS et propose un algorithme de routage *unicast* basé sur l'état des liens. L'algorithme se base sur une version modifiée de l'heuristique SMM (*Single Mixed Metric routing*) présentée auparavant dans [COS 99]. Nous présentons ici une version améliorée de l'algorithme basé sur l'état des liens (Single Mixed Metric routing - SMM-LS) et des résultats obtenus par simulation. SMM-LS a été implémenté comme un module dans NS (*Network Simulator Version 2*) [FAL 00]. L'analyse de performance compare SMM-LS avec cinq autres algorithmes basés, eux aussi, sur l'algorithme pour trouver le chemin le plus court de Dijkstra. Les résultats montrent que SMM-LS est une approche envisageable pour le calcul de routes avec trois besoins de QoS.

Le reste de ce papier est organisé de la manière suivante : la Section 2 discute les principaux problèmes liés au routage à QoS, la Section 3 présente SMM-LS, la Section 4 présente quelques détails de son déploiement dans NS et l'analyse des résultats obtenus et finalement la Section 5 présente quelques conclusions et perspectives de travail futur.

2. Routage à Contraintes Multiples

Le calcul de routes contraint par plusieurs besoins de QoS est un problème complexe [MA 97, WAN 96]. Les algorithmes peuvent traiter les métriques de différentes façons. La première différence concerne la corrélation entre les métriques. Le délai de transmission de bout en bout dépend de la capacité du lien, de la bande passante réservée, du délai de propagation, de la probabilité de perte sur le lien et du modèle de trafic employé. Ma et Steenkiste [MA 97] ont proposé des algorithmes qui se basent sur des métriques dépendantes entre elles, où la difficulté est de définir la relation entre les différents paramètres.

Une autre façon de traiter les métriques est de les considérer indépendantes les unes des autres. Par exemple, le délai de propagation est indépendant de la bande passante disponible. Dans ce cas là, la complexité de calcul de l'algorithme devient le principal défi comme l'explique la sous-section 2.1.

Ce papier suppose l'utilisation de métriques indépendantes. Les propositions trouvées dans la littérature traitent les métriques de trois façons distinctes: individuellement, comme un coût total (une métrique mixte) ou à l'aide d'heuristiques. Les sous-sections qui suivent analysent ces différentes approches pour justifier nos choix dans SMM-LS.

2.1. *Métriques multiples*

La complexité de l'algorithme est fonction du nombre de métriques considérées et de leurs règles de composition. Les besoins de QoS usuels sont le délai de transmission, la gigue, le coût, la probabilité de perte et la bande passante. Ces métriques ont des règles de composition distinctes. Si la métrique du chemin est la valeur minimale entre les métriques de tous les liens, la règle de composition est dite concave. Les règles additive et multiplicative sont évidentes. La bande passante est concave, tandis que le délai, la gigue et le coût sont additifs. La règle de composition pour la probabilité de perte est un peu plus complexe. Soit p un chemin, $lp(p)$ sa métrique, et $lp(i,j)$ la probabilité de perte sur chacun de ses liens, entre les noeuds i et j. La formule de $lp(p)$ est donc

$$lp(p) = 1 - ((1 - lp(i,j)) \times (1 - lp(j,k)) \times ... \times (1 - lp(q,r))). \quad [1]$$

Pourtant, si la probabilité de transmission avec succès remplace la probabilité de perte, la règle de composition devient multiplicative.

Wang et Crowcroft [WAN 96] ont démontré que le problème de trouver un chemin dans le réseau soumis à des contraintes sur n métriques additives et m métriques multiplicatives est NP-difficile si $n + m \geq 2$. Pourvu que la bande passante soit une métrique concave, l'élagage des liens impraticables peut garantir ce besoin de QoS. Par conséquent, des algorithmes pas trop complexes ne pourraient traiter que la bande passante et une autre métrique parmi le délai, la gigue ou la probabilité de perte par exemple.

2.2. *Métrique unique mixte*

Une idée simple pour retomber sur un problème de routage classique avec une seule métrique consisterait à traduire les besoins de QoS de l'utilisateur dans une mesure unique (un coût total) et l'utiliser ensuite comme métrique. Par exemple, soit un chemin p. Une fonction proportionnelle à la bande passante, $B(p)$, et inversement proportionnelle au délai, $D(p)$, et à la probabilité de perte, $L(p)$, peut être définie (Équation 2). Le chemin avec la plus grande valeur de $f(p)$ est la meilleure route.

$$f(p) = \frac{B(p)}{D(p) \times L(p)}. \quad [2]$$

Le coût total présente néanmoins deux problèmes. Le premier, maximiser $f(p)$ n'assure pas tous les besoins de QoS. Le second, la règle de composition de cette fonction n'est pas définie car les règles de composition des différents paramètres sont distinctes. Malgré ces problèmes la métrique unique mixte est utilisée dans plusieurs approches [GUE 97, AND 98]. La métrique unique mixte n'assure pas chaque besoin de QoS individuellement, mais les poids des différentes métriques peut être ajusté dans le calcul de la fonction de coût pour augmenter l'importance d'un besoin de QoS spécifique.

2.3. *Heuristiques*

Plusieurs propositions d'heuristiques existent dans la littérature pour le routage à QoS garantie [VOG 96, LEE 95, AND 98, SAL 97]. Les heuristiques les plus performantes en termes d'utilisation efficace des ressources du réseau supposent le routage centralisé. Un routeur (ou station de gestion) connaît la topologie complète du réseau et l'ensemble des flux à transporter. Cette entité est responsable du calcul des routes et pourvu qu'elle ait une information globale sur le réseau, elle est capable de faire les meilleurs choix. Pourtant, cette approche introduit un élément central qui compromet le fonctionnement de tout le réseau en cas de panne. À ce fait s'ajoutent les besoins en termes de mémoire et capacité de calcul qu'une telle entité exigerait pour des réseaux de taille importante ou avec des flux nombreux.

Le calcul de routes distribué est plus adapté aux réseaux de grande taille ayant de nombreux flux ou un trafic très dynamique. Ce travail essaie de cibler ce cas, et suppose que les calculs des routes sont faits par le routeur local à la source. La section qui suit présente notre proposition et la Section 4 la compare avec quelques autres approches, elles aussi distribuées.

3. L'algorithme SMM-LS

Dans [COS 00] nous avons proposé et analysé la performance de l'algorithme SMM - *Distance Vector* (SMM-DV) qui se base sur des vecteurs de distance pour réaliser le routage saut par saut. Dans ce papier nous proposons une version améliorée de SMM basé sur l'état des liens (SMM-LS). Cet algorithme suppose que les routes sont fixées pendant la durée de vie des flux grâce à une sorte de routage explicite ou de marquage de routes (*route pinning*) [DAV 98].

L'idée de base de SMM-LS est d'éliminer du graphe tous les liens n'ayant pas suffisamment de bande passante et ensuite d'exécuter une version modifiée de l'algorithme de Dijkstra. Nous supposons que la source (le routeur local à la source) connaît la topologie complète du réseau. SMM-LS utilise une métrique unique mixte qui combine le délai et la probabilité de perte. À la place de la probabilité de perte, l'algorithme emploie la valeur absolue du logarithme de la probabilité de transmission avec succès (Équation 3) pour éviter une règle de composition complexe.

$$slog(p) = |log(1 - L(p))|. \qquad [3]$$

Par ailleurs, nous supposons que les routes ne peuvent pas avoir une probabilité de perte plus grande que 90%. De ce fait, on conclut que $0 < slog < 1$. Nous considérons que l'unité millisseconde est appropriée pour le délai [FEI 98] et nous nous limitons donc au cas où le délai est un numéro entier. Par conséquent, nous avons une métrique unique qui représente le délai et la probabilité de perte, où la partie entière est le délai et la partie décimale représente la probabilité de perte. Cette propriété nous permet de construire une heuristique simple et capable de découvrir des chemins praticables dans le réseau.

Soit $G = (N,A)$ un réseau avec N noeuds et A arêtes. Deux numéros réels sont associés à chaque arête (i,j) : b_{ij} représente la bande passante disponible et $u_{ij} =$

$slog_{ij} + d_{ij}$ la métrique unique mixte, où $slog_{ij}$ est la valeur absolue du logarithme de la probabilité de transmission avec succès et $d_{i,j}$ est le délai de propagation.

Pour une arête (i,j) inexistante nous considérons que $b_{ij} = 0$ et $u_{ij} = \infty$. Soit un chemin $p = (i,j,k,...,q,r)$. La largeur de ce chemin, $w(p)$, égale $min[b_{ij},b_{jk},...,b_{qr}]$ et sa longueur, $l(p)$, égale $u_{ij} + u_{jk} + ... + u_{qr}$. Soient trois constantes, $\mathbf{B}$ la bande passante minimale, $\mathbf{D}$ le délai maximal et $\mathbf{P}$ la valeur maximale pour le logarithme de la probabilité de transmission avec succès ($\mathbf{U} = \mathbf{D} + \mathbf{P}$ représente la métrique unique mixte maximale). Le problème qui se pose est donc de trouver un chemin, p, entre i et r tel que $w(p) \geq \mathbf{B}$ et $l(p) \leq \mathbf{U}$.

Soit un chemin p avec x sauts. Si $slog_{ij}$ pour chaque lien (i,j) dans p est plus petit que $\mathbf{P}/x$, alors $slog(p)$ est plus petit que $\mathbf{P}$. Pourtant, x est inconnu pendant le calcul du chemin. L'idée est donc de limiter $slog_{ij}$ à $\mathbf{P}/C$, où C représente la longueur du chemin le plus long dans le graphe du réseau (en supposant le pire des cas). Si $slog_{ij}$ est limité à $\mathbf{P}/C$ à chaque pas de l'algorithme, alors à la fin $slog(p)$ est plus petit que $\mathbf{P}$. Cette limitation peut être, néanmoins, trop forte car la route peut toujours avoir une longueur en nombre de sauts plus petite que C.

Un schéma alternatif est donc proposé pour alléger cette contrainte. À chaque pas de l'algorithme un solde de sauts (SS_i) et un solde de probabilité logarithmique (SP_i) sont associés à chaque noeud i. Le solde de sauts est la différence entre la longueur du chemin le plus long dans le graphe et la longueur du chemin candidat entre la source et le noeud i. Le solde de probabilité est la différence entre la valeur demandée et la valeur actuelle du logarithme de la probabilité de transmission avec succès pour le chemin candidat entre la source et le noeud i. La quantité de probabilité logarithmique qui peut être additionnée à chaque pas de l'algorithme est limitée à SP_i/SS_i au lieu de $\mathbf{P}/C$. Cette valeur peut être plus grande que $\mathbf{P}/C$, si SP_i est plus petit que $SP_i \times (\mathbf{P}/C)$. Alors la probabilité de trouver un chemin dans le graphe du réseau augmente. Supposons que le noeud 1 soit la source, l'algorithme SMM-LS est le suivant :

Pas 1: $\forall i,j : u_{ij} = \infty$ si $b_{ij} < \mathbf{B}$.

Pas 2: $L = \{(1,1)\}$; $SS_1 = C$; $SP_1 = \mathbf{P}$.

$\forall i, i \neq 1$:

$U_i = u_{1i}$ et $ph_i = 1$

si $(U_i == \infty)$, $SS_i = C$, $SP_i = \mathbf{P}$;

sinon $SS_i = C - 1$, $SP_i = \mathbf{P} - (u_{1i} - \lfloor u_{1i} \rfloor)$.

Pas 3: Trouver $k \notin L \mid (U_k = \min_{i \notin L} U_i$ et $U_i - \lfloor U_i \rfloor \leq \mathbf{P})$.

Si $U_k > \mathbf{U}$, il n'y a pas de chemin possible. Arrêter.

$L := L \cup \{(k,ph_k)\}$.

Si L contient le noeud m, un chemin a été trouvé. Arrêter.

Pas 4: $\forall i \notin L$:

```
Si ((U_k + u_ki < U_i) et (u_ki − ⌊u_ki⌋ < SP_i/SS_i)),
  ou si ((U_k + u_ki < U) et ((SP_k − ⌊u_ki⌋)/(SS_k − 1) < SP_i/SS_i)),
  faire
   { U_i := U_k + u_ki.
     ph_i := k.
     SS_i := SS_k − 1.
     SP_i := SP_k − ⌊u_ki⌋. }
Aller au pas 3.
```

Le Pas 4 a été modifié par rapport à la proposition originelle de SMM [COS 99]. Ce pas vérifie si le noeud k constitue un meilleur “saut précédent” pour les noeuds i qui n'ont pas encore été inclus dans le chemin en construction, L. Si en choisissant k la longueur du chemin devient plus petite, alors k est choisi selon la version originelle de l'algorithme de Dijkstra. Le test qui suit dans la deuxième ligne du Pas 4 garantit que la raison entre les soldes proposées dans notre heuristique est respectée. La troisième ligne constitue une deuxième possibilité de choisir k comme saut précédent : même si la longueur du chemin est plus grande en choisissant k, ce noeud peut encore être choisi si la nouvelle longueur ne rend pas la route impraticable *et* si la raison des soldes est meilleure. Cette heuristique augmente la probabilité de trouver un chemin praticable.

Le Pas 1 comprend $O(N^2)$ opérations et est exécuté une seule fois. Le Pas 2 comprend $O(N)$ opérations et est également exécuté une seule fois. Les Pas 3 et 4 comprennent $O(N)$ opérations et sont répétés $(N-1)$ fois dans le pire des cas. La complexité de l'algorithme est donc $O(N^2)$, c'est à dire la même complexité de l'algorithme de Dijkstra original.

L'heuristique de SMM-LS peut être améliorée en ajoutant une certaine complexité. L'idée c'est de réaliser la recherche de Dijkstra à plusieurs reprises avec des valeurs différentes de C (soit $\mathbf{C} = \{C_i\}$), de façon à augmenter la probabilité de trouver un chemin praticable. Ça veut dire que les Pas 2 à 4 sont répétés $|\mathbf{C}|$ fois. La complexité de l'algorithme devient donc $O(|\mathbf{C}| \times N^2)$. Pourvu que $|\mathbf{C}|$ soit petit, la complexité de SMM-LS est acceptable et l'algorithme résiste au facteur d'échelle.

4. La Simulation de SMM-LS

Nous avons implémenté l'algorithme SMM-LS dans NS [FAL 00]. NS possède une implémentation dynamique de l'algorithme de routage de Bellman-Ford sur laquelle nous nous sommes basés dans notre travail précédent (SMM-DV) [COS 00]. Pour simuler SMM-LS, nous avons dû implémenter un mécanisme d'état des liens similaire à celui utilisé par OSPF (*Open Shortest-Path First*) [MOY 98].

L'implémentation du schéma de routage à QoS consiste d'un nouveau protocole de routage et d'une implémentation des liens modifiée pour admettre les trois paramètres de QoS utilisés: le délai de propagation, la bande passante disponible et la probabilité de perte. Nous avons utilisé le module de Services Intégrés disponible dans NS pour simuler le contrôle d'admission, l'établissement de connexion et la réservation de bande passante. Nous avons mesuré la probabilité d'admission de connexions pour analyser l'efficacité de SMM-LS dans la recherche de chemins praticables et l'utilisation des ressources du réseau.

Nous avons comparé SMM-LS avec cinq autres algorithmes. Le premier est tout simplement l'algorithme de Dijkstra classique en utilisant le nombre de sauts comme métrique (*Dijkstra's Shortest-Path algorithm* - DSP). En théorie, DSP devrait fournir une probabilité plus faible d'admission d'une connexion puisqu'il ne prend pas en compte les besoins de QoS des applications. Pour cette raison on l'a choisi comme référence dans la comparaison des algorithmes les uns avec les autres. BWDD (*BandWidth pruning and Delay Dijkstra search*) implémente la proposition de Wang et Crowcroft [WAN 96], c'est à dire l'élagage des liens impraticables suivi d'une recherche de Dijkstra avec le délai comme métrique. BWMD (*BandWidth pruning and Mixed metric Dijkstra search*) est le même algorithme que BWDD, sauf que le délai est remplacé par une métrique unique mixte qui combine le délai et la probabilité de perte. SMM-LS est l'algorithme proposé. OC (*Overall Cost*) implémente une recherche de Dijkstra basée sur un coût total comme celui présenté dans l'Équation 2. SOC (*Scaled OC*) est une version modifiée de l'approche du coût total, où les poids des trois métriques sont modifiés de façon à que les trois besoins de QoS aient la même importance dans la fonction de coût. Les facteurs de multiplication ont été choisis selon les valeurs des paramètres de QoS des liens du réseau.

4.1. *Expérimentations*

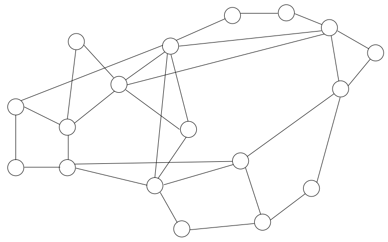


Figure 1. *La topologie ISP.*

La Figure 1 présente la topologie utilisée dans nos expérimentations. Cette topologie est proche de la topologie utilisée dans [APO 98] et peut être considérée comme typique du réseau d'un grand ISP (*Internet Service Provider*). Le délai de propagation de même que la probabilité de perte des liens sont uniformément distribués dans les intervalles $[min,max]$ présentés dans le Tableau 1. La bande passante des liens est homogène et égale à $10Mbps$.

Tableau 1. *Caractéristiques de la topologie simulée.*

Bande passante des liens	10Mbps
Quantité de noeuds	18
Quantité de liens	30

	min	max
Délai des liens	1 ms	10 ms
Perte des liens	0.001	0.005

Chaque connexion dans notre modèle comprend une paire source-destination et une demande avec trois besoins de QoS : en délai, en bande passante et en probabilité de perte. Les connexions sont censées arriver indépendamment dans chaque noeud selon une distribution de Poisson. L'intervalle entre l'arrivée des connexions est exponentiellement distribué avec moyenne $0.4s$. La durée de chaque connexion est exponentiellement distribuée avec $3s$ de moyenne. Les besoins de QoS des connexions sont variés dans les différentes expérimentations. La quantité de bande passante disponible dans les liens est simplement définie comme la capacité du lien moins la somme des réservations sur ce lien, puisque nous ne sommes pas intéressés par le transfert de données. Chaque simulation a une durée de 1500 secondes et est répétée dix fois. Cela veut dire que chaque configuration dans les expérimentations décrites par la suite a généré à peu près 35.000 requêtes de connexion.

Nous avons conduit trois expérimentations où un besoin de QoS est varié et les deux autres sont gardés fixes et négligeables. L'objectif est d'évaluer la sensibilité de chaque algorithme à chaque besoin de QoS séparément. Le Tableau 2 résume les caractéristiques des différentes expérimentations.

Tableau 2. *Caractéristiques des requêtes de connexion.*

	Besoins de QoS								
	Délai (ms)			Bande passante ($x \times 64kbps$)			Probabilité de perte		
Exp.	dl_i	dl_f		bw_i	bw_f		lp_i	lp_f	
		min	max		min	max		min	max
1	1	5	100	$fixe$			$fixe$		
2	$fixe$			1	10	160	$fixe$		
3	$fixe$			$fixe$			0.001	0.005	0.1
4	1	5	100	1	10	160	0.001	0.005	0.1

La Figure 2 présente les résultats obtenus dans les trois premières expérimentations. L'axe X représente la limite supérieure de l'intervalle de QoS demandée. L'axe Y représente la probabilité d'admission d'une connexion dans les graphes à gauche, et la différence par rapport à l'algorithme DSP dans les graphes à droite dans la Figure 2.

Dans l'expérimentation 1, les besoins en bande passante et probabilité de perte demandés par les connexions sont très faibles de façon à que le délai domine le choix de la route. Le délai demandé est uniformément distribué dans l'intervalle $[dl_i,dl_f]$.

dl_i est gardé constant et la limite supérieure dl_f est variée entre min et max comme le présente le Tableau 2. Les demandes de QoS sont ainsi plus faibles quand cet intervalle est plus long. Le plus dl_f est petit, les plus exigeantes sont les requêtes car les connexions admettent un délai maximum plus petit.

Dans l'expérimentation 2, le délai et la probabilité de perte demandés sont fixes. La bande passante demandée par les connexions est uniformément distribuée dans l'intervalle $[bw_i \times 64kbps, bw_f \times 64kpbs]$. bw_f est varié, mais dans ce cas, les intervalles plus longs se traduisent dans des demandes de QoS plus fortes. Ce fait peut être observé dans le côté gauche de la Figure 2(b). La charge du réseau est relativement faible puisque la probabilité d'admission d'une connexion est au dessus de 70% la plupart du temps.

Dans l'expérimentation 3 les requêtes ont des besoins en délai et bande passante constants. La probabilité de perte demandée est uniformément distribuée dans $[lp_i, lp_f]$, comme le présente le Tableau 2. Dans ce cas, une autre fois, les demandes de QoS sont moins exigeantes quand lp_f est plus grand et par conséquent la probabilité d'admission d'une connexion est plus importante (Figure 2(c)).

Dans la quatrième expérimentation, les trois séries de requêtes de QoS sont combinées, comme le présente le Tableau 2. Nous avons inversé l'ordre des requêtes en bande passante de manière à avoir une corrélation entre les différents essais et le niveau de QoS demandé. Les niveaux de QoS représentés sur l'axe horizontal du graphe de la Figure 3 sont caractérisés par des demandes de QoS plus faibles pour les niveaux plus à droite (Le niveau 0 présente les trois besoins de QoS les plus forts tandis que le niveau 10 présente les trois demandes les moins exigeantes.).

4.2. *Analyse des résultats*

Besoin en délai

La Figure 2(a) présente les résultats obtenus lors de la première expérimentation. DSP présente la pire performance comme prévu. BWDD, BWMD et SMM-LS ont à peu près le même comportement. Ces résultats sont attendus puisque les trois approches se traduisent par une recherche de Dijkstra basée sur le délai dans ce scénario. Néanmoins, SMM-LS présente un léger désavantage car l'heuristique des soldes risque d'ignorer certains chemins dans le calcul des routes. OC et SOC présentent les pires performances, ce qui démontre qu'au moins pour cette topologie et paramètres de QoS l'approche de coût total est inefficace. OC et SOC souffrent tous les deux de l'effet des demandes de QoS déséquilibrées - la valeur minimale demandée pour $f(p)$ (Équation 2) ne réussit pas à bien exprimer la QoS demandée. La raison est simple, la même valeur de $f(p)$ peut être obtenue avec un $D(p)$ grand et un $L(p)$ petit, ou avec un $D(p)$ petit et un $L(p)$ grand. La fonction de coût total cache l'importance relative des différents besoins de QoS.

Besoin en bande passante

La Figure 2(b) présente l'expérimentation 2, où le besoin en bande passante est varié. DSP présente la pire performance car il ne considère pas la bande passante dans le calcul de routes. BWDD, BWMD et SMM-LS ont une performance équivalente. Ce résultat est attendu puisque les trois algorithmes traitent le besoin en bande passante de la même façon - à travers l'élagage des liens impraticables. Une fois encore, les algorithmes OC et SOC présentent la pire performance entre les algorithmes qui prennent en compte la QoS. La raison de cette mauvaise performance est la même que pour la première expérimentation. Les besoins de QoS des requêtes ont des échelles très hétérogènes mais ce fait est ignoré par la fonction de coût total.

Besoin en probabilité de perte

La Figure 2(c) présente les résultats issus de la troisième expérimentation, où les besoins en délai et bande passante des requêtes sont faibles et constants. Le besoin en probabilité de perte est varié. Ce graphe démontre l'efficacité de l'heuristique employée par SMM-LS en assurant le besoin en probabilité de perte. Quand ce besoin est trop fort (lp_f est petit) tous les algorithmes présentent une mauvaise performance, même si comparés à DSP qui ignore la probabilité de perte dans le calcule de routes. BWDD et BWMD ont à peu près la même performance, ce qui veut dire que l'utilisation de la métrique unique mixte dans BWMD n'est pas efficace pour assurer le besoin en probabilité de perte. La raison est que le délai domine la probabilité de perte dans la métrique mixte. Encore une fois OC et SOC ont des mauvaises performances, pourtant légèrement meilleures que celles des expérimentations 1 et 2. Ce comportement est attendu puisque le besoin en probabilité de perte est numériquement petit, ce qui rend le déséquilibre entre les différents besoins moins important et par conséquent la fonction de coût total présente une meilleure performance.

Requêtes avec trois besoins de QoS

La Figure 3 présente les résultats de l'expérimentation 4. La Figure 3(a) démontre que les niveaux de QoS de gauche sont trop forts, ce qui augmente la probabilité de refus d'une connexion. Les niveaux de QoS moins exigeants causent une probabilité d'admission de connexions aux alentours de 80%. Quand les niveaux de QoS sont très lâches (par exemple, les niveaux 9 et 10 dans la Figure 3(b)), BWDD et BWMD présentent une performance similaire et proche de SMM-LS. Cela s'explique par le faible besoin en probabilité de perte. Cependant, quand les niveaux de QoS sont trop forts (niveaux 0 et 1) les performances de tous les algorithmes se ressemblent. En fait, la plupart des requêtes sont impraticables dans ce scénario.

SMM-LS présente sa meilleure performance à l'intérieur de l'intervalle des niveaux de QoS. Dans cette plage SMM-LS est capable d'accepter un plus grand nombre de connexions que toutes les autres approches, assurant un meilleur niveau d'utilisation des ressources du réseau. Il faut noter que cette probabilité d'admission de connexions plus forte est obtenue au détriment d'une complexité additionnelle par rapport aux autres approches. Cette complexité de traitement additionnelle n'est pas importante, heureusement. Dans nos expérimentations avec la topologie ISP, la taille de $\mathbf{C}$ égale 9. Néanmoins, $|\mathbf{C}|$ peut être réduit puisque les plus grands C_i's sont rarement utilisés. Nous en concluons que la réduction de $\mathbf{C}$ aurait un faible impact sur la probabilité d'admission de connexions.

5. Conclusion

Le routage à QoS fournit des garanties de QoS pour les applications et améliore l'utilisation des ressources du réseau. Pourtant le routage à QoS risque d'être un mécanisme coûteux. Ce papier a proposé SMM-LS, un algorithme capable de calculer des routes contraintes par trois besoins de QoS - le délai de propagation, la bande passante disponible et la probabilité de perte de paquets. SMM-LS est basé sur une version modifiée de l'algorithme du plus court chemin de Dijkstra alliée à l'utilisation d'une heuristique basée sur des soldes. Nous avons démontré que la complexité de SMM-LS est compensée par le gain sur la probabilité d'admission de connexions, ce qui conduit à une utilisation plus efficace des ressources du réseau. Les améliorations apportées à la première version de SMM [COS 99] ont été cruciales pour atteindre une probabilité d'admission plus importante.

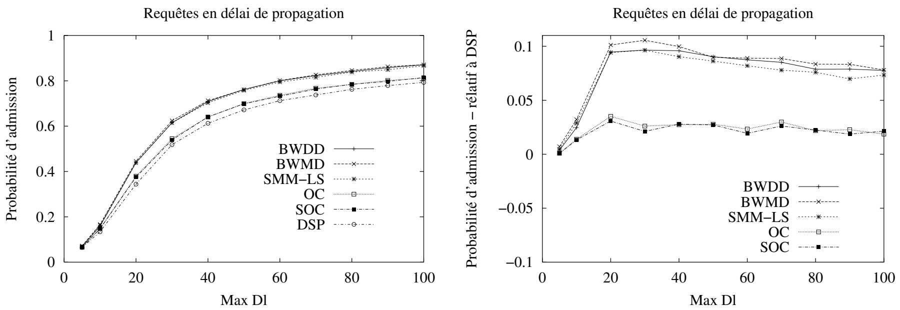
Notre travail futur comprend la réduction de l'information d'état des liens propagée par le protocole. L'état des liens ne serait diffusé que quand des flux demandant une quantité de ressources au-dessus d'un certain seuil arrivent dans le routeur. Notre objectif est d'identifier le compromis entre la précision de l'information d'état du réseau et la probabilité d'admission de connexions. Nous avons aussi comme but d'analyser la pertinence de SMM-LS dans la construction d'arbres de distribution multipoint. Pour cela nous allons comparer SMM-LS à l'heuristique utilisée dans [AND 98, LEE 95]. SMM-LS pourrait remplacer cette heuristique dans une architecture comme celle présentée dans [BIS 99].

Remerciements

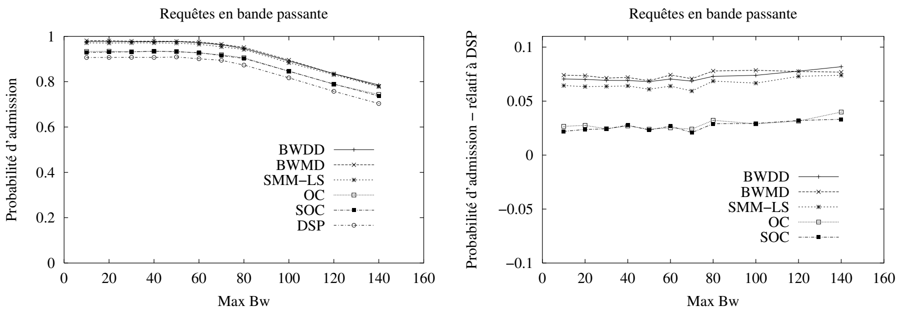
Nous tenons à remercier Mme. Anne Fladenmuller pour son aide précieuse dans la préparation de cet article, ainsi que le CAPES/COFECUB pour le financement de Luís Costa.

6. Bibliographie

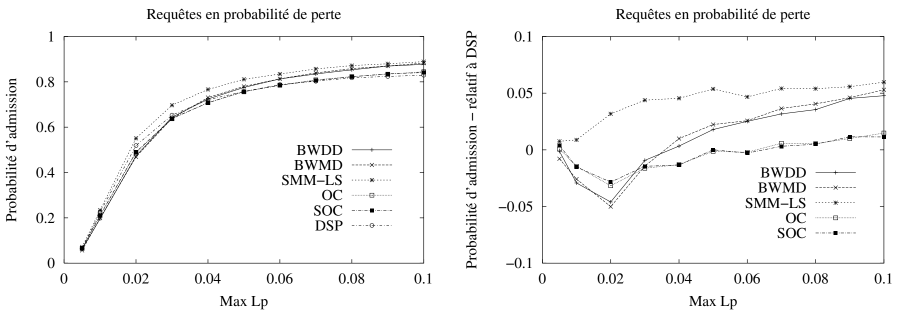
[AND 98] ANDREW L. L. H., KUSUMA A. A. N. A., « Generalised Analysis of a QoS-Aware Routing Algorithm », *IEEE GLOBECOM'98*, 1998, p. 118-123.



(a) variation du délai maximum demandé.

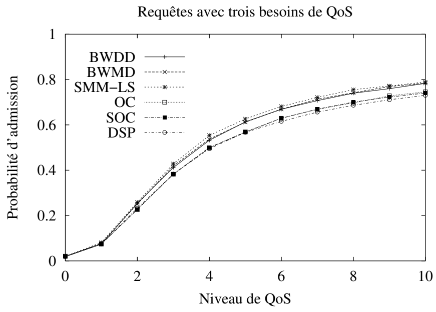


(b) variation de la bande passante minimum demandée.

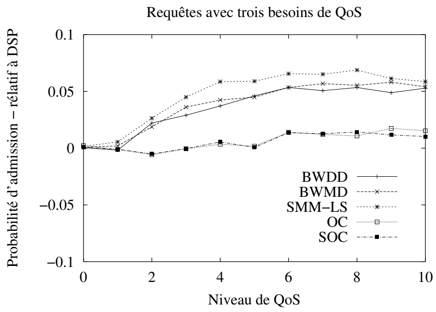


(c) variation de la probabilité de perte maximum demandée.

Figure 2. *Probabilité d'admission de connexions selon la variation d'un seul besoin de QoS.*



(a)



(b)

Figure 3. *Variation des niveaux de QoS.*

[APO 98] APOSTOLOPOULOS G., GUERIN R., KAMAT S., TRIPATHI S. K., « Quality of Service Based Routing: A Performance Perspective », *ACM SIGCOMM'98*, septembre1998, p. 17-28.

[BIS 99] BISWAS S., IZMAILOV R., RAJAGOPALAN B., « A QoS-Aware Routing Framework for PIM-SM Based IP-Multicast », juin1999, Internet-draft: draft-biswas-pim-sm-qos-00.txt.

[BLA 98] BLAKE S., BLACK D., CARLSON M., DAVIES E., WANG Z., WEISS W., « RFC2475: An Architecture for Differentiated Services », Network Working Group, décembre1998.

[BRA 94] BRADEN R., CLARK D., SHENKER S., « RFC1633: Integrated Services in the Internet Architecture: An Overview », Network Working Group, juin1994.

[COS 99] COSTA L. H. M. K., DUARTE O. C. M. B., « A Scalable QoS-based Routing Mechanism for Supporting Multimedia Applications », *IEEE International Conference on Multimedia Computing and Systems - ICMCS'99*, vol. 2, juin1999, p. 346-351.

[COS 00] COSTA L. H. M. K., FDIDA S., DUARTE O. C. M. B., « Distance-vector QoS-based Routing with Three Metrics », *IFIP Networking'2000 / HPN - High Performance Networking*, mai2000.

[CRA 98] CRAWLEY E., NAIR R., RAJAGOPALAN B., SANDICK H., « RFC2386: A Framework for QoS-Based Routing », Network Working Group, août1998.

[DAV 98] DAVIE B., GIBSON J., « Enabling Explicit Routing in IP Networks », *IEEE GLOBECOM'98 - Internet Mini-Conference*, 1998, p. 125-130.

[FAL 00] FALL K., VARADHAN K., « NS notes and documentation », University of California at Berkeley/Lawrence Berkeley National Laboratory, février2000, Disponible sur http://www-mash.cs.berkeley.edu/ns.

[FAY 99] FAYET V., KHOTIMSKY D. A., PRZYGIENDA T., « Hop-by-Hop Routing with Node-Dependent Topology Information », *IEEE INFOCOM'99*, mars1999.

[FEI 98] FEI A., PEI G., LIU R., ZHANG L., « Measurements on Delay and Hop-count of the Internet », *IEEE GLOBECOM'98 - Internet Mini-Conference*, 1998.

[GUE 97] GUERIN R., ORDA A., WILLIAMS D., « QoS Routing Mechanisms and OSPF Extensions », *IEEE GLOBECOM'97*, novembre1997.

[LEE 95] LEE W. C., HLUCHYJ M. G., HUMBLET P. A., « Routing Subject to Quality of Service Constraints in Integrated Communication Networks », *IEEE Network Magazine*, vol. 9, n° 4, 1995, p. 46-55.

[LOR 98] LORENZ D. H., ORDA A., « QoS Routing in Networks with Uncertain Parameters », *IEEE INFOCOM'98*, 1998.

[MA 97] MA Q., STEENKISTE P., « Quality of Service Routing for Traffic with Performance Guarantees », *IFIP Fifth International Workshop on Quality of Service*, New York, mai1997, p. 115-126.

[MOY 98] MOY J., « OSPF version 2 », avril1998.

[RAJ 98] RAJAGOPALAN B., NAIR R., « Multicast Routing with Resource Reservation », *Journal of High Speed Networks*, vol. 7, n° 2, 1998.

[SAL 97] SALAMA H. F., REEVES D. S., VINIOTIS Y., « A Distributed Algorithm for Delay-Constrained Unicast Routing », *IEEE INFOCOM'97*, 1997.

[TRA 99] TRAFT-PLOTKIN N., BELLUR B., OGIER R., « Quality-of-Service Routing using Maximally Disjoint Paths », *IEEE/IFIP IWQoS'99*, juin1999.

[VOG 96] VOGEL R., HERRTWICH R. G., KALFA W., WITTIG H., WOLF L. C., « QoS-based Routing of Multimedia Streams in Computer Networks », *IEEE Journal on Selected Areas in Communications*, vol. 14, n° 7, 1996, p. 1235-1244.

[WAN 96] WANG Z., CROWCROFT J., « Quality of Service Routing for Supporting Multimedia Applications », *IEEE Journal on Selected Areas in Communications*, vol. 14, n° 7, 1996, p. 1228-1234.