

**Titre:** Routage probabiliste dans un réseau IP utilisant la réservation de ressources  
Title: resources

**Auteur:** Kajetan Litwin  
Author:

**Date:** 2003

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Litwin, K. (2003). Routage probabiliste dans un réseau IP utilisant la réservation de ressources [Mémoire de maîtrise, École Polytechnique de Montréal].  
Citation: PolyPublie. <https://publications.polymtl.ca/7135/>

## Document en libre accès dans PolyPublie

Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/7135/>  
PolyPublie URL:

**Directeurs de recherche:** Brunilde Sanso  
Advisors:

**Programme:** Non spécifié  
Program:

**In compliance with  
Canadian Privacy Legislation  
some supporting forms  
may have been removed from  
this dissertation.**

**While these forms may be included  
in the document page count,  
their removal does not represent  
any loss of content from the dissertation.**



UNIVERSITÉ DE MONTRÉAL

ROUTAGE PROBABILISTE DANS UN RÉSEAU IP UTILISANT LA  
RÉSERVATION DE RESSOURCES

KAJETAN LITWIN  
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE ÉLECTRIQUE)  
MARS 2003



National Library  
of Canada

Acquisitions and  
Bibliographic Services

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque nationale  
du Canada

Acquisitions et  
services bibliographiques

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* Votre référence

ISBN: 0-612-86410-3

*Our file* Notre référence

ISBN: 0-612-86410-3

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

# Canadä

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ROUTAGE PROBABILISTE DANS UN RÉSEAU IP UTILISANT LA  
RÉSERVATION DE RESSOURCES

présenté par: LITWIN Kajetan

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. SOUMIS François, Ph.D., président

Mme. SANSO Brunilde, Ph.D., membre et directrice de recherche

M. GIRARD André, Ph.D., membre

à mes parents

## REMERCIEMENTS

J'aimerais exprimer ma reconnaissance aux personnes qui ont contribué à la réalisation de ce projet de recherche.

Je tiens à remercier vivement mon directeur de recherche, Mme Brunilde Sansò, pour m'avoir proposé ce sujet et pour m'avoir conseillé et appuyé durant la réalisation du projet.

Je remercie ma famille et mon épouse pour leur appui et leur patience, ainsi que pour leurs encouragements qui m'ont aidé à persévérer, même pendant les périodes les plus difficiles du projet.

## RÉSUMÉ

Le réseau Internet a connu une croissance très rapide, particulièrement au cours de la dernière décennie, au point où l'on envisage maintenant la convergence des réseaux de données, de la voix et de la télévision. Au début, le réseau Internet a opéré sous le principe du meilleur effort, qui a assuré sa simplicité et sa robustesse. Les nouvelles applications de l'Internet exigent une qualité de service mieux contrôlée. La qualité de service dans les réseaux IP est offerte selon deux stratégies définies par le IETF, les Services Intégrés (Integrated Services, intserv) et les Services Différenciés (Differentiated Services, diffserv).

Le modèle des Services Intégrés prévoit l'établissement de réservations de ressources pour les clients du réseau pour assurer la qualité de service. Le protocole RSVP a été proposé pour établir les réservations de ressources. Le modèle des Services Différenciés propose de diviser le trafic en plusieurs classes à l'entrée du réseau, pour offrir un traitement approprié à chaque classe.

Pour mettre en oeuvre la réservation de ressources dans les noeuds du réseau, de nouvelles disciplines de service ont été développées, dont la plus répandue est le Generalized Processor Sharing (GPS) et son approximation discrète, le Weighted Fair Queuing (WFQ). La discipline de service Weighted Fair Queuing (WFQ) offre un taux de service garanti aux flots de données, même en cas de congestion du réseau. De plus, le délai et le taux de perte de paquets (à cause des files d'attente

pleines) sont bornés lorsque les sources sont contrôlées par des "leaky bucket". Il existe aussi un modèle stochastique de GPS, qui considère plutôt la probabilité de délai et la probabilité de la taille des files d'attente.

Un des éléments importants pour le support de la qualité des service dans les réseaux IP est le routage, dont le rôle est de trouver un chemin qui satisfait les exigences des clients. Actuellement, le routage dans l'Internet est basé surtout sur la topologie du réseau. Dans le contexte de la qualité de service, on a besoin d'un routage plus sophistiqué, pour mieux exploiter les ressources du réseau et satisfaire les exigences des clients. Il existe une littérature abondante sur des algorithmes et des heuristiques de routage avec des contraintes multiples, dont plusieurs basés sur le modèle déterministe des réseaux GPS et WFQ. Cependant, le modèle stochastique de GPS a été peu étudié jusqu'à maintenant.

Dans ce mémoire, nous évaluons, par simulation, l'avantage d'un algorithme de routage basé sur la qualité de service dans un réseau utilisant la discipline de service GPS. Nous considérons les modèles déterministe et stochastique de GPS, en plus du routage basé uniquement sur la topologie.

Nous proposons un algorithme de routage basé sur l'énumération de chemins disjoints. Cet algorithme permet de générer rapidement plusieurs chemins possibles pour une paire de noeuds. De plus, étant donné la nature spécifique du modèle de GPS, il augmente les chances de trouver un chemin satisfaisant les exigences des clients. Cet algorithme sera utilisé avec le modèle déterministe et le modèle

stochastique de GPS pour résoudre le problème de routage avec contrainte de délai ou de probabilité de délai maximum. Le critère de comparaison des différents algorithmes de routage est l'efficacité du réseau, que l'on définit comme le nombre de clients que le réseau peut accepter tout en respectant leurs exigences de qualité de service.

Nous proposons ensuite une modification de l'algorithme de routage basé sur le modèle stochastique de GPS. Dans cet algorithme modifié, le choix d'un chemin est basé sur une combinaison de l'état *courant* du réseau et de son état *moyen*. L'état moyen du réseau peut être obtenu par observation de la charge sur une période assez longue. Nous étudions l'effet de la variation des coefficients de l'état courant et de l'état moyen sur l'efficacité du réseau. De plus, nous étudions trois façons différentes de combiner l'information sur l'état courant et l'état moyen du réseau pour le contrôle d'admission et le routage, afin de trouver la méthode la plus avantageuse, toujours du point de vue de l'efficacité du réseau.

Les principales contributions de ce mémoire sont le développement de l'algorithme d'énumération de chemins disjoints, l'algorithme de routage basé sur le modèle stochastique de GPS et l'algorithme de routage prédictif permettant d'augmenter l'efficacité du réseau.

## ABSTRACT

The fast growth of the Internet communication network and the convergence of multimedia and data services through the Internet create the need for a predictable and controlled quality of service. One of the fundamental principles of the Internet is the "best-effort" packet transmission, which means that the network does not guarantee the delivery of packets. However, this model is no longer satisfactory for new applications of the Internet, such as voice and video and other business-critical transactions.

In order to meet the QoS requirements of multimedia applications, two different but complementary frameworks have been developed by the IETF. The Integrated Services (intserv) service model is based on resource reservation on a per-flow basis. Complementary to intserv, the Differentiated Services model separates the traffic into several classes (levels) of Quality of Service at the entry to the network.

To support the resource reservations, new service disciplines have been developed for packet scheduling in the network nodes. One of the most popular choices to implement resource reservation is the Weighted Fair Queuing service discipline, also called Generalized Processor Sharing. This service discipline has interesting properties which make it suitable to implement resource reservations. It offers a guaranteed transmission rate to registered data flows, protecting them from effects

of network congestion. The delay or probability of delay under GPS are bounded for traffic conforming to the leaky-bucket or EBB envelopes.

An area of research that received a lot of attention recently is the development of routing algorithms taking into account the Quality of Service (QoS) requirements of data flows. Recent studies show that this type of routing can improve the network utilization compared to routing based only on topology. However, most algorithms and heuristics found in litterature address the case of the deterministic model of GPS.

In this work, we focus on the intserv model of resource reservation for individual data flows. Our goal is to provide a measurement of the gain in network efficiency that is achievable using Quality of Service enabled routing, in networks using the Generalized Processor Sharing service discipline to implement resource reservations. We study both the deterministic and the stochastic models of GPS. The quality of service requirements are the end-to-end delay or probability of delay.

We propose a routing algorithm based on enumeration of node-disjoint paths in the network. This algorithm quickly generates several possible paths in the network for a given pair of nodes. Also, given the nature of the GPS service discipline, this algorithm increases the chance to find a suitable path in the network which satisfies the QoS requirements of the traffic. We use this algorithm in conjunction with deterministic and stochastic models of GPS to select the best path for a new data flow.

In addition, we investigate the impact of basing the routing decision partially on future, average state of the network, not only on the current state of the network at the moment when the decision is made. We explore a range of coefficients for both the average and the current network states to determine their effect on the network's utilization. In addition, we examine three different ways of combining the current and the average state of the network in the admission control and routing algorithms, to determine the one that yields the highest utilization of the network.

Using simulation, we compare routing algorithms based on deterministic and stochastic models of GPS with routing algorithms based solely on network topology, which represent the type of routing commonly used in today's Internet. The comparison criterion is the network utilization, i.e. the percentage of sessions that are blocked because their quality of service requirements cannot be satisfied by the network. We also investigate the effect of network density on the performance of these routing algorithms.

The main contributions of this research are: the development of an algorithm to find node-disjoint paths in a network, evaluation of routing algorithms based on the deterministic and stochastic models of the GPS service discipline, and development of a "predictive" routing algorithm based partially on the average state of the network.

**TABLE DES MATIÈRES**

|  |              |
|--|--------------|
| <b>DÉDICACE . . . . .</b>                                      | <b>iv</b>    |
| <b>REMERCIEMENTS . . . . .</b>                                 | <b>v</b>     |
| <b>RÉSUMÉ . . . . .</b>  | <b>vi</b>    |
| <b>ABSTRACT . . . . .</b>                                      | <b>ix</b>    |
| <b>TABLE DES MATIÈRES . . . . .</b>                            | <b>xii</b>   |
| <b>LISTE DES FIGURES . . . . .</b>                             | <b>xvi</b>   |
| <b>LISTE DES NOTATIONS . . . . .</b>                           | <b>xix</b>   |
| <b>LISTE DES TABLEAUX . . . . .</b>                            | <b>xxi</b>   |
| <b>LISTE DES ANNEXES . . . . .</b>                             | <b>xxiii</b> |
| <b>INTRODUCTION . . . . .</b>                                  | <b>1</b>     |
| <b>CHAPITRE 1 QUALITÉ DE SERVICE DANS L'INTERNET . . . . .</b> | <b>7</b>     |
| 1.1 Définitions . . . . .                                      | 7            |
| 1.2 Historique du réseau Internet . . . . .                    | 9            |
| 1.3 Émergence de la qualité de service . . . . .               | 10           |

|   |   |           |
|---|---|-----------|
| 1.4   | Recherches dans le domaine de la qualité de service . . . . . | 12        |
| 1.5   | Modèle des services intégrés . . . . .                        | 15        |
| 1.5.1   | Classification du trafic . . . . .                            | 15        |
| 1.5.2   | Types de service . . . . .                                    | 17        |
| 1.5.3   | Éléments du modèle intserv . . . . .                          | 18        |
| 1.5.3.1   | Classificateur de paquets . . . . .                           | 18        |
| 1.5.3.2   | Ordonnateur de paquets . . . . .                              | 19        |
| 1.5.3.3   | Contrôleur d'admission . . . . .                              | 19        |
| 1.5.3.4   | Protocole de réservations des ressources . . . . .            | 19        |
| 1.6   | Modèle des services différenciés . . . . .                    | 20        |
| 1.6.1   | Principe de fonctionnement . . . . .                          | 21        |
| 1.6.2   | Classes de service . . . . .                                  | 22        |
| 1.6.2.1   | Expedited Forwarding . . . . .                                | 22        |
| 1.6.2.2   | Assured Forwarding . . . . .                                  | 23        |
| <b>CHAPITRE 2 ÉLÉMENTS DE LA QUALITÉ DE SERVICE . . .</b> |   | <b>25</b> |
| 2.1   | Algorithmes de service . . . . .                              | 25        |
| 2.1.1   | Generalized Processor Sharing . . . . .                       | 31        |
| 2.2   | Routage . . . . .   | 35        |
| 2.2.1   | Heuristiques pour le routage par chemins multiples . . . . .  | 42        |
| 2.3   | Caractérisation du trafic . . . . .                           | 44        |

|   |   |           |
|---|---|-----------|
| 2.4   | Bornes de délai dans GPS . . . . .  | 46        |
| 2.4.1   | Borne déterministe . . . . .  | 48        |
| 2.4.2   | Borne stochastique . . . . .  | 50        |
| 2.4.3   | Comparaison des deux bornes . . . . .   | 53        |
| <b>CHAPITRE 3 MODÈLES DE ROUTAGE PROPOSÉS . . . . .</b> |   | <b>55</b> |
| 3.1   | Routage avec critère déterministe de QS . . . . .                                       | 58        |
| 3.2   | Routage avec critère stochastique de QS . . . . .                                       | 59        |
| 3.3   | Routage "stochastique+" avec considération de la charge moyenne sur le réseau . . . . . | 61        |
| 3.3.1   | Algorithme stochastique+ version 1 . . . . .  | 64        |
| 3.3.2   | Algorithme stochastique+ version 2 . . . . .  | 64        |
| 3.3.3   | Algorithme stochastique+ version 3 . . . . .  | 65        |
| 3.4   | Routage par le plus court chemin . . . . .  | 65        |
| 3.5   | Critères de comparaison des algorithmes de routage . . . . .                            | 66        |
| <b>CHAPITRE 4 SIMULATION . . . . .</b>                  |   | <b>68</b> |
| 4.1   | Génération de topologie du réseau . . . . .   | 71        |
| 4.2   | Recherche de chemins . . . . .  | 73        |
| 4.2.1   | Énumération de tous les chemins . . . . .   | 73        |
| 4.2.2   | Énumération des chemins disjoints . . . . .   | 76        |

|  |            |
|--|------------|
| <b>CHAPITRE 5 RÉSULTATS NUMÉRIQUES . . . . .</b>     | <b>85</b>  |
| 5.1 Topologies de réseaux . . . . .                  | 85         |
| 5.2 Algorithmes de routage . . . . .                 | 86         |
| 5.3 Trafic entrant dans le réseau . . . . .          | 88         |
| 5.4 Critère de comparaison des algorithmes . . . . . | 89         |
| 5.5 Validation du simulateur . . . . .               | 91         |
| 5.6 Variabilité des résultats . . . . .              | 92         |
| 5.7 Discussion des résultats . . . . .               | 95         |
| <b>CONCLUSION . . . . .</b>                          | <b>111</b> |
| <b>RÉFÉRENCES . . . . .</b>                          | <b>115</b> |

## LISTE DES FIGURES

|          |  |     |
|----------|--|-----|
| FIG. 2.1 | Aperçu d'algorithmes pour l'implementation de RSVP . . . . .   | 29  |
| FIG. 2.2 | Modèle de l'algorithme de service GPS . . . . .  | 32  |
| FIG. 2.3 | Mécanisme du "token bucket" . . . . .  | 47  |
| FIG. 4.1 | Étapes de l'ajout d'une session dans le réseau . . . . .   | 70  |
| FIG. 4.2 | Réseau 1 . . . . .   | 76  |
| FIG. 4.3 | Exemple de solution sous-optimale trouvée par l'algorithme de chemins disjoints . . . . .                      | 77  |
| FIG. 4.4 | Réseau pour la démonstration des résultats de ENUM et ENUM_D   | 80  |
| FIG. 4.5 | Chemins disjoints entre les noeuds 1 et 4 dans le réseau de la figure 4.4 - algorithme ENUM_D . . . . .        | 81  |
| FIG. 4.6 | Tous les chemins possibles entre les noeuds 1 et 4 dans le réseau de la figure 4.4 - algorithme ENUM . . . . . | 82  |
| FIG. 5.1 | Variabilité des résultats pour un réseau de 150 noeuds, 6500 arcs  | 93  |
| FIG. 5.2 | Variabilité des résultats pour un réseau de 150 noeuds, 6500 arcs  | 94  |
| FIG. 5.3 | Taux de blocage pour un réseau de 50 noeuds, 700 arcs . . .  | 96  |
| FIG. 5.4 | Taux de blocage pour un réseau de 50 noeuds, 1100 arcs . . .   | 97  |
| FIG. 5.5 | Taux de blocage pour un réseau de 100 noeuds, 500 arcs . . .   | 98  |
| FIG. 5.6 | Taux de blocage pour un réseau de 100 noeuds, 3000 arcs . . .  | 99  |
| FIG. 5.7 | Taux de blocage pour un réseau de 100 noeuds, 4500 arcs . . .  | 100 |

|           |  |     |
|-----------|--|-----|
| FIG. 5.8  | Taux de blocage pour un réseau de 150 noeuds, 6500 arcs . . .                | 102 |
| FIG. 5.9  | Taux de blocage pour un réseau de 150 noeuds, 10000 arcs . . .               | 103 |
| FIG. 5.10 | Taux de blocage pour un réseau de 50 noeuds, 100 arcs . . . . .              | 107 |
| FIG. 5.11 | Taux de blocage pour un réseau de 50 noeuds, 1100 arcs . . . . .             | 108 |
| FIG. I.1  | Taux de blocage pour un réseau de 50 noeuds, 100 arcs . . . . .              | 123 |
| FIG. I.2  | Taux de blocage pour un réseau de 50 noeuds, 350 arcs . . . . .              | 124 |
| FIG. I.3  | Taux de blocage pour un réseau de 50 noeuds, 700 arcs . . . . .              | 125 |
| FIG. I.4  | Taux de blocage pour un réseau de 50 noeuds, 1100 arcs . . . . .             | 126 |
| FIG. I.5  | Taux de blocage pour un réseau de 50 noeuds, 100 arcs . . . . .              | 127 |
| FIG. I.6  | Taux de blocage pour un réseau de 50 noeuds, 350 arcs . . . . .              | 128 |
| FIG. I.7  | Taux de blocage pour un réseau de 50 noeuds, 700 arcs . . . . .              | 129 |
| FIG. I.8  | Taux de blocage pour un réseau de 50 noeuds, 1100 arcs . . . . .             | 130 |
| FIG. I.9  | Taux de blocage pour un réseau de 50 noeuds, 100 arcs . . . . .              | 131 |
| FIG. I.10 | Taux de blocage pour un réseau de 50 noeuds, 350 arcs . . . . .              | 132 |
| FIG. I.11 | Taux de blocage pour un réseau de 50 noeuds, 700 arcs . . . . .              | 133 |
| FIG. I.12 | Taux de blocage pour un réseau de 50 noeuds, 1100 arcs . . . . .             | 134 |
| FIG. I.13 | Taux de blocage pour un réseau de 100 noeuds, 4500 arcs, version 1 . . . . . | 135 |
| FIG. I.14 | Taux de blocage pour un réseau de 100 noeuds, 500 arcs, version 2136         |     |
| FIG. I.15 | Taux de blocage pour un réseau de 100 noeuds, 3000 arcs, version 3 . . . . . | 137 |

|           |   |     |
|-----------|---|-----|
| FIG. I.16 | Taux de blocage pour un réseau de 150 noeuds, 10000 arcs,<br>version 1 . . . . .  | 138 |
| FIG. I.17 | Taux de blocage pour un réseau de 150 noeuds, 6500 arcs, ver-<br>sion 2 . . . . . | 139 |
| FIG. I.18 | Taux de blocage pour un réseau de 150 noeuds, 1000 arcs, ver-<br>sion 3 . . . . . | 140 |

## LISTE DES NOTATIONS

**IETF** Internet Engineering Task Force

**W3C** World Wide Web Consortium

**intserv** Integrated Services Working Group

**diffserv** Differentiated Services Working Group

**rfc** Request for Comments

**TCP** Transmission Control Protocol

**IP** Internet Protocol

**FTP** File Transfer Protocol

**QoS** Quality of Service

**RSVP** Resource ReserVation Protocol

**MPLS** Multiprotocol Label Switching

**OSPF** Open Shortest Path First

**CBQ** Class Based Queuing

**WFQ** Weighted Fair Queuing

**GPS** Generalized Processor Sharing

**PGPS** Packet-by-Packet GPS

**RPPS** Rate Proportional Processor Sharing

**FQS** Fair Queuing based on Starting Times

**FQF** Fair Queuing based on Finishing Times

**EBB** Exponentially Bounded Burstiness

**FIFO** First-In First-Out

**UDP** User Datagram Protocol

**RTCP** Real Time Control Protocol

**RTP** Real Time Transport Protocol

**RIP** Routing Information Protocol

**EDF** Earliest Deadline First

**PRR** Packet-by-Packet Round Robin

**WRR** Weighted Round Robin

**RED** Random Early Detection

**WRRED** Weighted Random Early Detection

## LISTE DES TABLEAUX

|          |  |     |
|----------|--|-----|
| TAB. 2.1 | Algorithmes de qualité de service-produits commerciaux . . . . .   | 27  |
| TAB. 3.1 | Sources de trafic et leurs paramètres EBB et "leaky bucket" . . .  | 59  |
| TAB. 3.2 | Sommaire de l'algorithme stochastique+ version 1 . . . . .   | 64  |
| TAB. 3.3 | Sommaire de l'algorithme stochastique+ version 2 . . . . .   | 65  |
| TAB. 3.4 | Sommaire de l'algorithme stochastique+ version 3 . . . . .   | 65  |
| TAB. 3.5 | Algorithmes de routage . . . . .   | 67  |
| TAB. 4.1 | Quelques chemins possibles entre les noeuds 13 et 6 dans le réseau de la figure 4.2 . . . . .                  | 75  |
| TAB. 4.2 | Liste d'adjacence pour le réseau de la figure 4.4 . . . . .  | 79  |
| TAB. 4.3 | Tous les chemins possibles entre les noeuds 1 et 4 pour le réseau de la figure 4.4 - algorithme ENUM . . . . . | 83  |
| TAB. 5.1 | Topologies de réseau étudiées . . . . .  | 85  |
| TAB. 5.2 | Algorithmes de routage . . . . .   | 87  |
| TAB. 5.3 | Sources ON-OFF et leurs caractérisations EBB . . . . .   | 88  |
| TAB. 5.4 | Exemple de la demande pour un réseau de 50 noeuds . . . . .  | 90  |
| TAB. 5.5 | Exemple de calcul du taux de blocage . . . . .   | 91  |
| TAB. 5.6 | Performance des algorithmes de routage . . . . .   | 95  |
| TAB. 5.7 | Taux de blocage en % après 2500 sessions, réseau de 50 noeuds,<br>VER. 1 . . . . .                             | 101 |

|           |   |     |
|-----------|---|-----|
| TAB. 5.8  | Taux de blocage en % après 2500 sessions, réseau de 50 noeuds,<br>VER. 2 . . . . .  | 101 |
| TAB. 5.9  | Taux de blocage en % après 2500 sessions, réseau de 50 noeuds,<br>VER. 3 . . . . .  | 102 |
| TAB. 5.10 | Taux de blocage en % après 2500 sessions, réseau de 100 noeuds,<br>VER. 1 . . . . . | 104 |
| TAB. 5.11 | Taux de blocage en % après 2500 sessions, réseau de 100 noeuds,<br>VER. 2 . . . . . | 104 |
| TAB. 5.12 | Taux de blocage en % après 2500 sessions, réseau de 100 noeuds,<br>VER. 3 . . . . . | 105 |
| TAB. 5.13 | Taux de blocage en % après 2500 sessions, réseau de 150 noeuds,<br>VER. 1 . . . . . | 107 |
| TAB. 5.14 | Taux de blocage en % après 2500 sessions, réseau de 150 noeuds,<br>VER. 2 . . . . . | 108 |
| TAB. 5.15 | Taux de blocage en % après 2500 sessions, réseau de 150 noeuds,<br>VER. 3 . . . . . | 109 |

**LISTE DES ANNEXES**

|                 |   |            |
|-----------------|---|------------|
| <b>Annexe I</b> | <b>Résultats numériques</b>   | <b>122</b> |
| I.1             | Réseau de 50 noeuds, version 1                                      | 122        |
| I.2             | Réseau de 50 noeuds, version 2                                      | 122        |
| I.3             | Réseau de 50 noeuds, version 3                                      | 122        |
| I.4             | Autres résultats pour des réseaux de 100 noeuds et 150 noeuds . . . | 122        |

## INTRODUCTION

Le réseau Internet est en train d'évoluer du modèle de service "meilleur effort" qui prévalait à ses débuts vers un modèle plus sophistiqué, capable d'offrir différents niveaux de Qualité de Service (QS). La commercialisation de l'Internet et la transmission de la voix et vidéo en direct exigent des délais et une fiabilité bien contrôlés. De plus, les fournisseurs de services Internet sont intéressés à pouvoir offrir des services de différents niveaux de qualité et de prix.

Les recherches sur la qualité de service portent sur plusieurs domaines: routage, protocoles, algorithmes de service, contrôle d'admission, matériel (processeurs plus rapides, parallélisme, réseaux optiques). L'organisme qui oriente le développement de l'Internet, le Internet Engineering Task Force (IETF) a créé deux groupes de travail, le Integrated Services Working Group (intserv) et le Differentiated Services Working Group (diffserv) pour diriger les travaux sur la qualité de service dans les réseaux de communications.

Le groupe intserv propose un modèle basé sur la réservation de ressources pour flots de données individuels. Le protocole Resource Reservation Protocol (RSVP) est utilisé pour établir ces réservations de ressources, permettant de garantir la qualité de service requise par les applications. Ce protocole est en fonction aujourd'hui dans les routeurs commercialisés par les principaux fabricants d'équipements de communication tels que Cisco, Juniper et Nortel.

Le concept de la réservation de ressources a entraîné le développement de nouveaux algorithmes de service dans les noeuds du réseau. Un algorithme de service représente la stratégie utilisée dans un noeud pour servir l'ensemble des liens entrant dans ce noeud et transmettre les paquets vers les liens sortant du noeud. Dans le cas du service "meilleur effort", les algorithmes de service ne tiennent pas compte des contraintes telles que le taux de transmission ou le délai. Ces algorithmes ne sont pas adaptés au contexte de la qualité de service. Les nouveaux algorithmes de service doivent garantir le niveau de performance requis, quel que soit l'état du réseau et la charge sur le noeud.

Le Generalized Processor Sharing (GPS) est un exemple d'algorithme conçu pour offrir un support à la qualité de service. Il garantit la largeur de bande requise par un client, même en situation de congestion du réseau. De plus, cet algorithme possède une caractéristique intéressante qui permet de l'analyser de façon quantitative. En effet, sous l'algorithme GPS, il est possible de calculer des bornes déterministes [5,6] et des bornes stochastiques [46] sur la longueur des files d'attente et le délai subi par un paquet. Ces bornes peuvent être utilisées pour le contrôle d'admission et le routage. L'algorithme GPS est un modèle fluide, qui ne tient pas compte de la division en paquets. Les approximations discrètes de cet algorithme, tenant compte de la transmission par paquets, sont souvent utilisées pour supporter la qualité de service et la réservation de ressources.

Dans le contexte de la qualité de service, on a aussi besoin de protocoles de

routage plus sophistiqués, tenant compte des caractéristiques spécifiques des algorithmes de service et de la nature probabiliste de la demande. La plupart des algorithmes de routage utilisés actuellement dans le réseau Internet ne tiennent pas compte des paramètres de qualité de service. Leur but principal est d'assurer la connectivité. En général, ces protocoles fournissent une seule route pour atteindre la destination, quelquefois avec une ou plusieurs routes de secours en cas de panne et pour mieux répartir la charge. Les routes sont comparées en fonction de métriques, par exemple la distance ou la capacité des liens. Prenons l'exemple du routage par le plus court chemin, par exemple avec le protocole OSPF. Supposons que les sessions entrantes exigent un certain niveau de qualité de service. Si le chemin fourni par OSPF ne peut assurer ce niveau de qualité de service, la session est rejetée, même s'il existe peut-être un autre chemin acceptable dans le réseau. En utilisant un algorithme de routage adapté au contexte de qualité de service, on peut ainsi réduire le taux de blocage des sessions et augmenter également l'efficacité du réseau.

Les objectifs de ce mémoire sont:

- proposer un nouvel algorithme de routage basé sur le modèle stochastique de GPS
- étudier l'effet du routage sur l'efficacité du réseau. Nous voulons déterminer s'il est possible d'augmenter le nombre de sessions admises dans le réseau,

tout en respectant leurs exigences de qualité de service, en utilisant un routage basé sur les conditions de charge du réseau

- comparer les modèles déterministe et stochastique de l'algorithme de service GPS

Nous considérons que dans chaque noeud du réseau, l'algorithme de service Generalized Processor Sharing (GPS) est utilisé pour mettre en oeuvre la réservation de ressources. L'efficacité du réseau est exprimée en termes du taux de blocage de sessions. Une session est bloquée quand on ne peut pas trouver un chemin qui satisfait ses exigences de qualité de service.

Nous étudions la performance de trois types d'algorithmes de routage:

1. routage basé uniquement sur la topologie du réseau
2. routage basé sur le modèle déterministe de l'algorithme GPS
3. routage basé sur le modèle stochastique de l'algorithme GPS

Comme point de référence, nous considérons le routage basé uniquement sur la topologie, sans considération de qualité de service. Un algorithme de routage de type "plus court chemin" est utilisé pour trouver un chemin pour une nouvelle session. Si le chemin trouvé ne satisfait pas les exigences de qualité de service, la session est rejetée.

Les deux autres algorithmes sont basés sur la comparaison de plusieurs chemins possibles pour une session. En théorie, on peut évaluer tous les chemins possibles pour une session et choisir celui qui est le meilleur selon les critères de performance utilisés. Cependant, le nombre de chemins peut être trop grand pour permettre de les évaluer dans un temps raisonnable. Pour limiter le nombre de chemins à évaluer, on a développé un algorithme d'énumération des chemins disjoints.

Dans le cas du modèle déterministe, la décision de routage est basée sur le délai maximum pour chaque chemin considéré. Cette valeur de délai sera toujours respectée, même en cas de congestion du réseau.

Dans le cas du modèle stochastique, la décision de routage est basée sur la probabilité de délai pour chaque chemin considéré. Les garanties de qualité de service sont moins strictes: il y a une faible probabilité que le délai maximum soit dépassé. Toutefois, les applications multimédia peuvent tolérer un certain niveau de délai et de perte de paquets sans que la qualité de la transmission ne soit affectée.

Le chapitre 1 de ce mémoire présente un bref historique du développement du réseau Internet, avec l'accent sur le développement du support pour la qualité de service. Le chapitre 2 décrit plus en détail les différents éléments utilisés pour assurer la qualité de service dans les réseaux de communication par paquets. En particulier, ce chapitre décrit les algorithmes de service, les modèles de trafic et les calculs de paramètres de qualité de service qui serviront de base au travail présenté dans ce mémoire, avec un aperçu des travaux antérieurs dans les domaines connexes.

Le chapitre 3 présente les détails des différents algorithmes de routage conçus et évalués dans le cadre de ce projet. Les détails de la simulation et les algorithmes d'enumération des chemins sont présentés dans le chapitre 4. Le chapitre 5 présente une discussion des résultats obtenus.

## CHAPITRE 1

### QUALITÉ DE SERVICE DANS L'INTERNET

#### 1.1 Définitions

La qualité de service dans les réseaux de communication par paquets est généralement définie par les paramètres suivants:

**Débit** taux de transmission maximal sur un chemin.

**Fiabilité** mesure du nombre d'erreurs de transmission et du taux de perte de paquets. Le taux de perte représente le pourcentage de paquets émis par la source qui n'arrivent pas à destination.

**Délai** temps écoulé entre le moment où le paquet est émis et le moment où il est reçu.

**Gigue** différence entre le délai le plus court et le délai le plus long subi par un paquet. Alternativement, différence maximale entre les délais de deux paquets consécutifs. Cette dernière définition se retrouve dans la spécification du Real Time Control Protocol (RTCP) [14].

La qualité de service peut être appliquée au niveau d'un flot de données ou d'un ensemble de flots de données agrégés. Un flot de données dans le réseau Internet est généralement décrit par l'ensemble des paramètres suivants [40]:

- adresse IP de la source
- numéro de port de la source
- adresse IP de la destination
- numéro de port de la destination
- protocole (UDP, TCP)

Dans le cadre de ce mémoire, la définition d'une session (flot de données) est un peu différente. Les identificateurs de ports et de protocoles ne sont pas utiles, par contre, nous avons besoin de connaître les caractéristiques du trafic émis par la source. Nous allons considérer qu'un flot de données est décrit par les paramètres suivants:

- identité du noeud source
- identité du noeud destination
- caractéristiques du trafic de la source (par exemple: modèle de source, processus stochastique, utilisation d'un régulateur)

## 1.2 Historique du réseau Internet

Le réseau Internet a connu une croissance très rapide, passant de quelques dizaines d'usagers des quatre ordinateurs de l'Arpanet en 1969 à quelques 600 millions d'usagers aujourd'hui. La croissance s'est nettement accélérée au cours de la dernière décennie. En 1990, le premier lien transatlantique à 1.5 Mbits par seconde est mis en service entre le CERN en Suisse et l'université Cornell aux États-Unis. Le développement dans le domaine du logiciel suit cette évolution pour permettre à un plus grand nombre d'usagers de profiter des ressources disponibles sur le réseau. Au début des années 1990, Archie, le premier outil de navigation est mis au point à Montréal. Le premier outil de navigation à interface graphique, appelé Mosaic, est lancé en 1993, facilitant l'accès à l'Internet pour le grand public, peu à l'aise avec l'interface texte. Dès 1994, les compagnies privées proposent des services de connexion à Internet. En 1994, America Online compte un million d'usagers. Dans la dernière moitié des années 90, on assiste à un développement fulgurant dans tous les domaines. Dans le domaine du matériel, il y a les processeurs de plus en plus rapides, les routeurs à très grande capacité, la transmission par fibre optique. Dans le domaine du logiciel, on assiste à la mise au point de logiciels graphiques et multimédia permettant la transmission de la voix et de la vidéo. Les premières expériences multimédia à grande échelle sur Internet sont lancées avec la diffusion des conférences de l'IETF [34]. Vers la fin de la décennie, pour accéder à Internet,

on n'a plus besoin d'un ordinateur personnel. Un téléphone cellulaire ou un ordinateur de poche avec modem sans fil permettent d'accéder au réseau, multipliant du même coup le nombre d'usagers. On se dirige maintenant vers la convergence des réseaux téléphoniques, de la télévision par câble et de l'Internet.

### **1.3 Émergence de la qualité de service**

Dans les premiers temps de l'expansion grand public d'Internet (1994-1997) le réseau était surtout utilisé pour le transfert de fichiers (protocole FTP), la communication par courrier électronique (protocole SMTP) et le visionnement de pages Web. Ce type de trafic est peu sensible au délais de transmission. Les nouvelles applications de l'Internet, telles que le multimédia, la téléphonie par Internet ou encore la télé-médecine, exigent une qualité de service mieux contrôlée. Ces applications à haut débit nécessitent des garanties d'acheminement des paquets et des délais de transmission faibles et prévisibles.

Les principes d'opération du réseau Internet assurent sa relative simplicité et sa robustesse. Le réseau est conçu pour fonctionner en mode non-connecté (data-gramme). Il n'y a pas d'autorité centralisée; tous les noeuds sont au même niveau hiérarchique. Les messages sont divisés en paquets, qui sont transmis de façon indépendante et peuvent prendre des chemins différents pour se rendre à destination. Chaque noeud du réseau prend les décisions de routage pour les paquets qui tran-

sitent par ce noeud. Quand un noeud est en panne, les noeuds voisins acheminent les paquets de façon à contourner la panne.

Ces principes ont permis le développement rapide de l'Internet mais ils rendent le réseau peu adapté pour les applications sensibles au délai. La transmission de données s'effectue la plupart du temps selon le principe du "meilleur effort", ce qui veut dire que les paquets reçoivent le meilleur traitement possible étant donné l'état du réseau, mais qu'on ne peut garantir les critères de performance tels que le délai ou le taux de perte de paquets.

Le protocole de base utilisé dans l'Internet, appelé Internet Protocol (IP) assure une connexion point-à-point sans garanties de performance: les délais de transmission et de traitement peuvent varier, les paquets peuvent être perdus, reçus dans un ordre différent de celui dans lequel ils ont été envoyés ou encore reçus en plusieurs copies. Un autre protocole, appelé Transmission Control Protocol (TCP) est employé pour améliorer la fiabilité de connexions IP. En utilisant des mécanismes de retransmission de paquets, ce protocole assure une transmission sans erreurs, mais au prix d'une augmentation de délais.

Les nouvelles applications du réseau Internet, en particulier la transmission de la voix et de la vidéo en direct, nécessitent des changements au modèle du service "meilleur effort". Ces nouvelles applications exigent une bande passante réservée et un délai prévisible. Même pour les applications traditionnelles de l'Internet, comme le visionnement des pages Web, les utilisateurs sont plus exigeants aujourd'hui

qu'auparavant. Si une page Web prend trop de temps à être chargée, il y a de bonnes chances pour que l'utilisateur abandonne et aille voir ailleurs. Dans les applications commerciales de l'Internet (par exemple, la vente de livres et de disques sur Internet), ceci peut représenter la perte d'un client.

Selon les projections d'un des pionniers de l'Internet, Lawrence Roberts, maintenant à la tête de la compagnie Caspian Networks, le réseau pourrait bientôt offrir une qualité de service garantie pour la voix et la vidéo. De plus, le temps de chargement des pages Web serait inférieur à une seconde dans tous les cas. Les fournisseurs de services pourraient facturer les clients en fonction de la qualité de service désirée. Dans un futur plus lointain, la qualité de service serait suffisante pour que le réseau soit utilisé pour la téléphonie et la télévision haute définition, remplaçant du même coup l'infrastructure du réseau téléphonique et du câble de télévision. Il s'agit seulement de prévisions, mais elles montrent que la qualité de service est un des enjeux les plus importants en ce moment pour le futur du réseau Internet.

#### **1.4 Recherches dans le domaine de la qualité de service**

Les recherches sur la qualité de service sont menées dans plusieurs domaines: matériel électronique et optique, algorithmes de service, protocoles de routage, contrôle d'admission et conditionnement du trafic, applications multimédia.

La technologie des liens physiques a évolué pour permettre d'atteindre un débit

de données très élevé. La fibre optique commercialisée aujourd’hui permet d’atteindre un débit de l’ordre de 10 Gb/s. On cherche maintenant à atteindre des vitesses de l’ordre de 400 Gb/s dans une seule fibre optique (source: Lucent). Les routeurs et les processeurs très rapides permettent de traiter et d'aiguiller les paquets sans ralentir leur débit (*line speed*).

Pour offrir une qualité de service garantie aux usagers, il faut tout d’abord dimensionner le réseau adéquatement pour répondre à la demande. Toutefois, ce n'est pas suffisant pour garantir la performance, car la demande peut quand même dépasser la capacité du réseau, quelle que soit sa taille. Pour contrôler la qualité de service, il faut restreindre le nombre d’usagers qui partagent les ressources du réseau. Le contrôle d’admission (au niveau des "fLOTS" individuels de données) et le conditionnement du trafic (au niveau des paquets d'un flot) permettent de limiter l'accès au réseau et, par conséquent, de maintenir la qualité de service pour les sessions actives.

De nouveaux algorithmes de service pour files d’attente sont développés pour supporter les réservations de largeur de bande et les garanties de performance, par exemple Weighted Fair Queuing (WFQ), Earliest Deadline First (EDF) et Class Based Queuing (CBQ). Les protocoles comme le Resource ReserVation Protocol (RSVP) et le Real Time Transport Protocol (RTP) et les nouvelles applications audio et vidéo, par exemple Quicktime d’Apple et Real Player de RealNetworks aident à exploiter le potentiel des réseaux à haut débit.

Les efforts de recherche, de développement et de standardisation dans le domaine des réseaux de communication sont coordonnés par des organismes internationaux, dont les deux plus importants sont Internet Engineering Task Force (IETF) et le World Wide Web Consortium (W3C). L'IETF maintient les standards décrits par les *Request for Comments (rfc)* et les *internet drafts*. Il est divisé en groupes de travail, spécialisés dans les domaines tels que les applications de l'Internet, le routage, la sécurité et le transport. Deux groupes parmi les plus importants dans le domaine du transport et de la qualité de service sont le Integrated Services Working Group (intserv) et Differentiated Services Working Group (diffserv).

Le groupe intserv, dont le travail a récemment été complété, a développé un modèle basé sur la réservation de ressources qui permet de garantir la qualité de service au niveau des flots de données individuels. Le protocole RSVP [19] a été proposé comme standard pour établir ces réservations de ressources. Le protocole RSVP est déjà implementé dans les équipements de communication actuels. La méthode de réservation de ressources présente tout de même un inconvénient: chaque noeud du réseau doit garder l'information d'état de tous les clients qui ont établi des réservations de ressources. Cette information d'état est utilisée pour servir les clients conformément à leur réservation.

Quand le nombre de réservations augmente, la quantité d'information d'état peut devenir trop grande pour être traitée efficacement. Par exemple, sur un lien OC-48 (2.4 Gbps), il peut y avoir jusqu'à 150 000 flots de données de 16 kbps

chacun. Si chaque flot établit une réservation, la quantité d'information dans les noeuds et la quantité de messages RSVP qui doivent circuler dans le réseau devient exorbitante [35].

Pour remédier à ce problème, le groupe diffserv propose d'agréger les flots individuels dans un certain nombre de "classes" de service. Les classes représentent différents niveaux de priorité en cas de congestion du réseau. Ceci permet de réduire la quantité d'information d'état dans les noeuds du réseau. Cependant, les garanties de qualité de service sont moins strictes que dans le cas de la réservation de ressources.

Les sections suivantes présentent une description plus approfondie des modèles intserv et le diffserv.

## 1.5 Modèle des services intégrés

### 1.5.1 Classification du trafic

Dans le modèle Integrated Services Working Group (intserv), les applications sont divisées en deux classes: les applications temps réel et les applications élastiques.

Les applications temps réel sont du type "playback". Typiquement, la source transmet un signal audio ou vidéo, échantillonné, numérisé et envoyé sous forme de paquets. Après un certain délai, les paquets arrivent au récepteur qui reconstruit le

signal. Considérons l'exemple d'une transmission vidéo de 2 minutes, par exemple un reportage disponible sur les sites des chaînes de télévision comme Radio-Canada ([www.radio-canada.com](http://www.radio-canada.com)) ou ABC ([www.abcnews.com](http://www.abcnews.com)). Lorsqu'on "clique" sur la vidéo pour la télécharger, la transmission commence mais la vidéo ne commence pas à jouer immédiatement car le récepteur doit accumuler assez de données dans une mémoire "tampon" pour éviter des pauses pendant la retransmission. Ce délai est basé sur une estimation du délai maximum qu'un paquet subit dans le réseau. L'estimation du délai peut être basée sur l'expérimentation ou des calculs de bornes déterministes ou stochastiques. Lorsque la vidéo commence à jouer, les paquets qui ne sont pas arrivés au moment où ils doivent être joués sont perdus. Les applications audio et vidéo peuvent tolérer un certain taux de perte de paquets sans que le signal ne devienne inintelligible.

Les applications "élastiques" ne sont pas sensibles au délai mais requièrent une transmission fiable, sans perte de paquets. On retrouve dans cette catégorie les applications comme File Transfer Protocol (FTP) (transfert de fichiers) et le courrier électronique. Ces applications s'accommodeent d'un service "meilleur effort" où la fiabilité est assurée par les protocoles de transport.

### 1.5.2 Types de service

Le modèle intserv offre trois types de service: le service de base sans garantie de performance (meilleur effort), le "partage contrôlé du lien" (Controlled Load) et le service avec garanties de performance (Guaranteed Service).

Dans le modèle "meilleur effort", il n'y a pas de garanties de performance. Tous les flots de données sont au même niveau de priorité. Si le réseau est dimensionné adéquatement, il n'y a pas de congestion et la qualité de service est acceptable. Cependant, les flots de données ne sont pas isolés les uns des autres: si un flot accapare toute la largeur de bande, le taux de perte de paquets et les délais augmentent pour tous les flots de données.

Dans le modèle de "charge contrôlée" où "partage contrôlé du lien" (Controlled Load où Controlled Link Sharing), la largeur de bande disponible est divisée entre plusieurs classes de trafic. En situation de surcharge du réseau, un certain pourcentage de la largeur de bande est garanti à chacune des classes. Lorsque le réseau est faiblement chargé, la largeur de bande inutilisée par les différentes classes est disponible pour tous les usagers. Les différentes classes de trafic peuvent représenter des groupes d'usagers ou du trafic appartenant à un protocole en particulier, par exemple User Datagram Protocol (UDP). Ce modèle de service peut être implementé par exemple par des files First-In First-Out (FIFO) avec un contrôle d'admission basé sur la théorie de la capacité équivalente ("effective bandwidth")

pour éviter la surcharge et la perte de paquets [35]. La discipline de service WFQ est particulièrement appropriée pour implementer ce type de service, car elle assure une largeur de bande garantie à chaque session en cas de surcharge du réseau.

Pour assurer la transmission des données temps réel, le modèle intserv propose le service avec garanties de performance (Guaranteed Service). Dans ce type de service, il n'y a pas de perte de paquets due à la congestion (il peut y avoir des pertes de paquets dues aux erreurs sur les liens physiques) et le délai est borné. Le modèle ne précise pas la nature des garanties; elles peuvent être déterministes ou stochastiques. Pour offrir ces garanties, on propose d'établir des réservations de ressources dans les routeurs sur le chemin emprunté par les paquets de données. Le protocole RSVP est le moyen proposé pour établir ces réservations de ressources.

### 1.5.3 Éléments du modèle intserv

Les paragraphes suivants expliquent les blocs fonctionnels d'un noeud conforme au modèle intserv.

#### 1.5.3.1 Classificateur de paquets

Le classificateur de paquets sépare les paquets entrants dans le noeud en différentes classes de service, selon l'information contenue dans l'en-tête des paquets. Il place les paquets dans les files d'attente appropriées pour le type de service requis (ex. meilleur effort, service garanti).

#### **1.5.3.2 Ordonnateur de paquets**

L'ordonnateur détermine l'ordre dans lequel les paquets doivent être envoyés sur les liens sortant du routeur. L'ordonnancement des paquets dépend de la discipline de service des différentes files d'attente dans le système. Par exemple, la discipline de service WFQ est souvent utilisée pour implanter les réservations de largeur de bande, car elle garantit le partage de la largeur de bande dans des proportions requises par les réservations.

L'ordonnateur de paquets s'assure aussi que les flots de données ne dépassent pas les limites de leurs réservations, par exemple en transmettant avec un débit plus élevé que prévu.

#### **1.5.3.3 Contrôleur d'admission**

Le contrôleur d'admission détermine si le routeur possède des ressources suffisantes pour accepter une nouvelle session et lui offrir les garanties de performance requises, sans affecter les sessions déjà en place. Il assure l'authentification des requêtes en vérifiant la provenance et la légitimité des réservations. Finalement, il accumule des données pour fins de statistiques et de facturation.

#### **1.5.3.4 Protocole de réservations des ressources**

Le groupe intserv a proposé la standardisation du protocole RSVP pour établir des réservations de ressources. Ce protocole, formalisé en 1997, est utilisé aujourd'hui.

d'hui dans un large éventail de routeurs commerciaux [13]. Le nombre d'utilisateurs de RSVP augmente depuis la commercialisation récente de Windows 2000, qui supporte ce protocole [44]. Le protocole RSVP est décrit dans l'annexe A.

## 1.6 Modèle des services différenciés

Contrairement au modèle intserv qui établit une nouvelle réservation de ressources pour chaque flot de données, le modèle des services différenciés diffserv divise le trafic en quelques classes de service. Les noeuds du réseau offrent un traitement privilégié à certaines classes, ce qui a pour effet d'augmenter la qualité de service perçue par les usagers prioritaires. Le concept du modèle diffserv est plus simple que celui de intserv. Il n'y a pas de mécanisme d'établissement et de maintien des réservations, donc pas de protocole de signalisation comme RSVP. Les noeuds du réseau n'ont pas à maintenir les informations de réservation. Cependant, le modèle diffserv n'offre pas de garanties de qualité de service. En fait, ce modèle est une amélioration du service "meilleur effort", permettant d'obtenir une qualité de service satisfaisante au prix d'une complexité un peu plus élevée des routeurs.

Le modèle des services différenciés peut être utilisé pour agréger les flots RSVP dans le coeur du réseau [43]. On bénéficie alors de la fine granularité du modèle intserv dans la périphérie du réseau, là où le volume de trafic est plus faible. Dans le coeur du réseau, on bénéficie de la simplicité du modèle diffserv et on évite les

problèmes potentiels de manque de ressources causés par un trop grand nombre de réservations RSVP.

### 1.6.1 Principe de fonctionnement

L'architecture du modèle diffserv est décrite dans [36] et [21]. Les paquets sont classés et marqués à l'entrée du réseau, selon le niveau de qualité de service désiré par l'application. Un champ de l'en-tête du paquet IP, appelé "Type Of Service" dans IPv4 et "Traffic Class" dans IPv6 [21] identifie la classe de service à laquelle appartient le paquet. Ce champ détermine le traitement, appelé Per-Hop Behavior (PHB), que le paquet va subir dans chaque noeud du réseau. Les noeuds du réseau offrent plusieurs PHB différents pour satisfaire les classes de service définies par l'IETF. Le PHB définit un ensemble de règles à suivre pour transmettre un paquet.

Ces règles peuvent être:

1. la file d'attente à utiliser
2. l'algorithme de service à utiliser
3. le niveau de priorité à accorder aux paquets
4. le chemin à prendre pour arriver à destination (routage).

Les PHB peuvent être implementés de plusieurs façons [7], en utilisant les algorithmes de service tels que WFQ et CBQ ou encore des mécanismes de contrôle de congestion tels que RED et WRED.

Si la correspondance entre le DSCP d'un paquet et les PHB disponibles dans un noeud ne peut être établie, alors le paquet sera traité selon le PHB de défaut, qui est celui du "meilleur effort".

L'architecture du modèle diffserv permet de l'appliquer à des réseaux de très grande taille. Les opérations les plus longues, soit la classification et le marquage des paquets sont effectuées par les noeuds périphériques du réseau, qui traitent un volume de trafic plus faible. Les noeuds au coeur du réseau appliquent un traitement convenu d'avance aux paquets déjà marqués.

### **1.6.2 Classes de service**

Trois classes de service ont été définies par diffserv:

1. Best Effort (BE)
2. Expedited Forwarding (EF)
3. Assured Forwarding (AF)

La première classe correspond au service de base "meilleur effort", alors que les deux autres classes offrent un niveau supérieur de qualité de service.

#### **1.6.2.1 Expedited Forwarding**

Ce service, aussi appelé Premium Class, offre une largeur de bande garantie, un faible taux de perte de paquets et un faible délai [38].

Le principe de base de cette classe de service est d'assurer que dans tous les noeuds du réseau, le débit du trafic EF entrant ne soit pas supérieur au débit du trafic EF sortant du noeud. Ainsi, on minimise les files d'attente et leurs conséquences: délai, gigue et pertes de paquets.

Ce PHB peut être implementé de plusieurs façons: files d'attente avec priorité, CBQ, WFQ. L'implementation comprend deux parties:

- un mécanisme de "scheduling" dans les noeuds, assurant un débit garanti à la classe EF en tout temps. Ce mécanisme doit aussi assurer un faible délai et un faible taux de perte de paquets pour la classe EF.
- un mécanisme de contrôle d'admission et de mise en forme du trafic EF, pour assurer que le débit entrant dans un noeud soit toujours inférieur au débit maximum réservé à cette classe. Ce mécanisme est nécessaire pour maintenir la qualité de service et pour protéger les autres classes de trafic.

#### **1.6.2.2 Assured Forwarding**

Ce service offre quatre niveaux de qualité de service, qui diffèrent par la quantité de ressources (mémoire dans les files d'attente) et la largeur de bande [16] offertes à leurs clients.

Chaque niveau de AF offre en plus trois degrés de priorité de perte (*drop pre-*

*cedence*). En cas de congestion, les routeurs protègent les flots plus prioritaires en laissant tomber d'abord les paquets des flots moins prioritaires.

Si un client dépasse son débit maximum souscrit, les routeurs s'efforcent de transmettre le trafic excédentaire mais avec une probabilité de perte plus élevée.

L'étude présentée dans les prochains chapitres repose sur des travaux antérieurs dans les domaines du routage, des algorithmes de service et de la modélisation du trafic. Le chapitre suivant présente ces concepts importants pour la suite de ce mémoire.

## CHAPITRE 2

### ELÉMENTS DE LA QUALITÉ DE SERVICE

Dans ce chapitre, nous faisons un survol de deux éléments qui sont intimement liés au support de la Qualité de Service (QS) dans les réseaux IP: les algorithmes de service dans les noeuds du réseau et le routage. Nous étudions ensuite deux modèles de trafic fréquemment utilisés: le "leaky bucket" et le Exponentially Bounded Burstiness (EBB), qui permettent de formuler des bornes de Qualité de Service pour certains algorithmes de service, comme l'algorithme Generalized Processor Sharing (GPS).

#### 2.1 Algorithmes de service

Dans le domaine des télécommunications, un algorithme de service représente la politique utilisée pour servir simultanément plusieurs flots de données dans un noeud du réseau. Un algorithme de service doit pouvoir gérer les situations de contention. Dans le contexte de la qualité de service, les algorithmes de service ont pour objectif additionnel d'assurer un partage équitable des ressources dans le cas du service "best effort" ou de respecter les garanties de performance de chaque flot de données dans le cas du service garanti.

Les algorithmes de service sont le résultat d'un compromis entre les trois objec-

tifs suivants: la performance, exprimée en termes de la qualité de service obtenue et de la fiabilité des résultats, l'efficacité du réseau exprimée comme le taux d'utilisation des ressources et la complexité des algorithmes [35,44]. Plus le débit de données est élevé, plus les algorithmes de service doivent être rapides et efficaces. Avec des liens entrant de type OC-48, un routeur a typiquement moins de 100 ns pour analyser le contenu de l'en-tête d'un paquet et prendre des décisions, par exemple pour placer le paquet dans une des files d'attente [40]. Les différents algorithmes de service existants peuvent être évalués selon les critères suivants:

**équité** partage équitable des ressources entre tous les clients

**isolation** protection contre les effets de pointes de trafic: une session ne peut pas affecter le trafic des autres sessions

**efficacité** nombre de flots qui peuvent être supportés pour la quantité de ressources disponibles

**complexité** difficulté d'implementation et temps d'exécution

Selon une enquête réalisée auprès des fabricants de matériel de communication [13], les deux algorithmes de service les plus souvent utilisés pour fournir le support au protocole de réservation de ressources RSVP sont le Class Based Queuing (CBQ) et le Weighted Fair Queuing (WFQ) [1], aussi appelé Generalized Processor Sharing

TAB. 2.1 *Algorithmes de qualité de service-produits commerciaux*

| Fabricant                     | FIFO | Round Robin | CBQ | WFQ |
|-------------------------------|------|-------------|-----|-----|
| 3COM                          | X    | X           | X   | X   |
| ASCEND                        | X    | X           |     | X   |
| Bay Networks                  | X    | X           | X   |     |
| CEFRIEL/Politecnico di Milano | X    |             | X   |     |
| Cisco                         | X    | X           |     | X   |
| CLASS Data Systems            |      |             | X   |     |
| Hewlett-Packard               |      |             | X   |     |
| IBM Research Center           |      |             |     | X   |
| Intel                         | X    | X           |     |     |
| IPHighway                     |      |             | X   | X   |
| Juniper                       |      |             |     |     |
| MICOM/Nortel                  | X    |             |     |     |
| Microsoft                     | X    | X           | X   | X   |
| Newbridge                     | X    | X           | X   | X   |
| Nortel                        |      |             |     | X   |
| RouterWare Inc                | X    |             |     |     |
| Silicon Graphics Inc          |      | X           |     |     |
| SONY                          | X    |             | X   | X   |
| SUN Microsystems              |      |             | X   |     |
| Tektronix                     | X    |             |     |     |

(GPS) [5, 6]. Le tableau 2.1 présente un aperçu des algorithmes de service utilisés par les fabricants de matériel de télécommunications pour implémenter RSVP.

Le plus simple des algorithmes de service est l'algorithme "premier arrivé-premier servi" First-In First-Out (FIFO). Les paquets sont servis dans l'ordre d'arrivée, sans considération de priorité. Lorsque la file d'attente est pleine, les nouveaux paquets qui arrivent sont perdus. Cette méthode ne peut pas être utilisée pour garantir la qualité de service. Un flot de données peut monopoliser toutes

les ressources, empêchant les autres flots de transmettre. De plus, le délai subi par un paquet dans le réseau n'est pas borné.

Des algorithmes de service plus sophistiqués permettent d'offrir un support pour les garanties de qualité de service. Par exemple, chaque flot de données peut avoir sa propre file d'attente. Le processeur sert les files d'attente une après l'autre en transmettant un paquet de chaque file, s'il y a des paquets en attente. Si la file est vide, il passe immédiatement à la file suivante. Ce mécanisme, appelé Packet-by-Packet Round Robin (PRR) isole les flots de données: un flot qui transmet à un taux très élevé ne va pas empêcher les autres de passer. L'algorithme Packet-by-Packet Round Robin (PRR) peut être modifié pour transmettre plus d'un paquet à la fois de chaque file d'attente, en fonction du poids affecté à chaque file: c'est le Weighted Round Robin (WRR). Les algorithmes de service Packet-by-Packet Round Robin (PRR) et Weighted Round Robin (WRR) font partie d'une classe d'algorithmes appelés Class Based Queuing (CBQ). Les principaux algorithmes utilisés pour implementer RSVP sont présentés à la figure 2.1.

Lorsque CBQ est utilisé pour implanter la réservation de ressources, une nouvelle classe de service (représentée par une file d'attente) est créée pour chaque réservation acceptée par un routeur. Pour éviter le problème de mise à l'échelle, on utilise plutôt un nombre fixe de classes de service et on assigne une classe existante à un nouveau flot de données.

Les deux algorithmes qui implementent le service de "charge contrôlée", Ran-

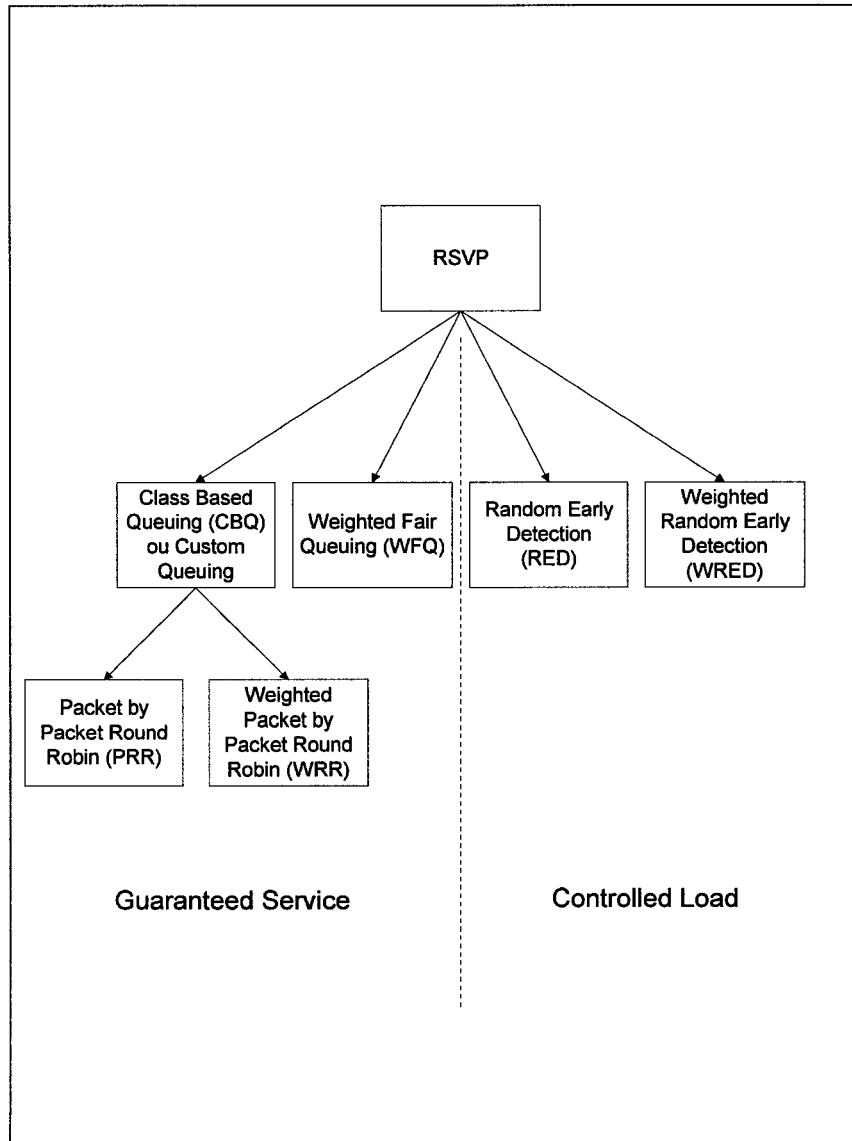


FIG. 2.1 Aperçu d'algorithmes pour l'implementation de RSVP

dom Early Detection (RED) et Weighted Random Early Detection (WRED) (figure 2.1) sont utilisés pour prévenir la congestion dans les files d'attente. Ces algorithmes suppriment des paquets de façon aléatoire à l'entrée de la file d'attente.

L'algorithme Random Early Detection (RED) réduit la congestion dans les files d'attente en rejettant les nouveaux paquets selon une certaine probabilité, qui dépend de la longueur moyenne de la file d'attente. Par exemple, supposons que l'administrateur du réseau a défini deux valeurs  $L_{max}$  et  $L_{min}$  qui représentent deux longueurs de la file d'attente. Soit  $L_{moy}$  la longueur moyenne de la file d'attente sur un certain intervalle de temps. L'algorithme RED peut être programmé de la façon suivante:

1. si  $L_{min} > L_{moy}$ , pas de paquets rejetés
2. si  $L_{max} > L_{moy} > L_{min}$ , les nouveaux paquets sont rejetés avec une certaine probabilité qui est fonction de  $L_{moy}$
3. si  $L_{moy} > L_{max}$ , tous les nouveaux paquets sont rejetés

L'algorithme WRED est semblable, mais il permet en plus de définir un niveau de probabilité de rejet par type de paquets, indiqué par un champ de l'en-tête IP du paquet.

Les algorithmes RED et WRED sont conçus pour être utilisés avec le protocole TCP. Dans TCP, les pertes de paquets causent une diminution du taux de transmission, ce qui a pour effet de réduire la probabilité de perte de paquets dans RED.

D'autres types de protocoles pourraient retransmettre les paquets perdus, ce qui ne ferait qu'augmenter la congestion.

### 2.1.1 Generalized Processor Sharing

La discipline de service la plus répandue pour implementer un service avec différents niveaux de priorité est Weighted Fair Queuing (WFQ), aussi appelée Generalized Processor Sharing (GPS). Dans WFQ, un poids est attribué à chaque flot de données, qui détermine l'importance de ce flot par rapport aux autres et le taux de service qu'il va obtenir. Le taux de service défini par ce poids est garanti, quel que soit le comportement des autres flots. Les sessions sont isolées les unes des autres: si une session transmet à un taux supérieur au taux prévu lors de la réservation, son délai va augmenter alors que les autres sessions ne seront pas affectées. De plus, la capacité excédentaire est répartie en proportion des poids de chaque session.

Le modèle conceptuel de l'algorithme de service GPS est présenté à la figure 2.2. Un noeud ayant une capacité de traitement  $r$ , servant simultanément les processus d'arrivée  $Q_1(t) \dots Q_N(t)$  peut être remplacé par  $N$  noeuds ayant des capacités de traitement  $r_1 \dots r_N$  servant des processus d'arrivée  $d_1(t) \dots d_N(t)$ .

Dans le cas du GPS idéal (fluide), lorsqu'il y a  $k > 0$  files d'attente non-vides, toutes les files sont servies simultanément avec un taux de service proportionnel à  $1/k$ . Les flots de données sont considérés continus, sans division en paquets. Dans

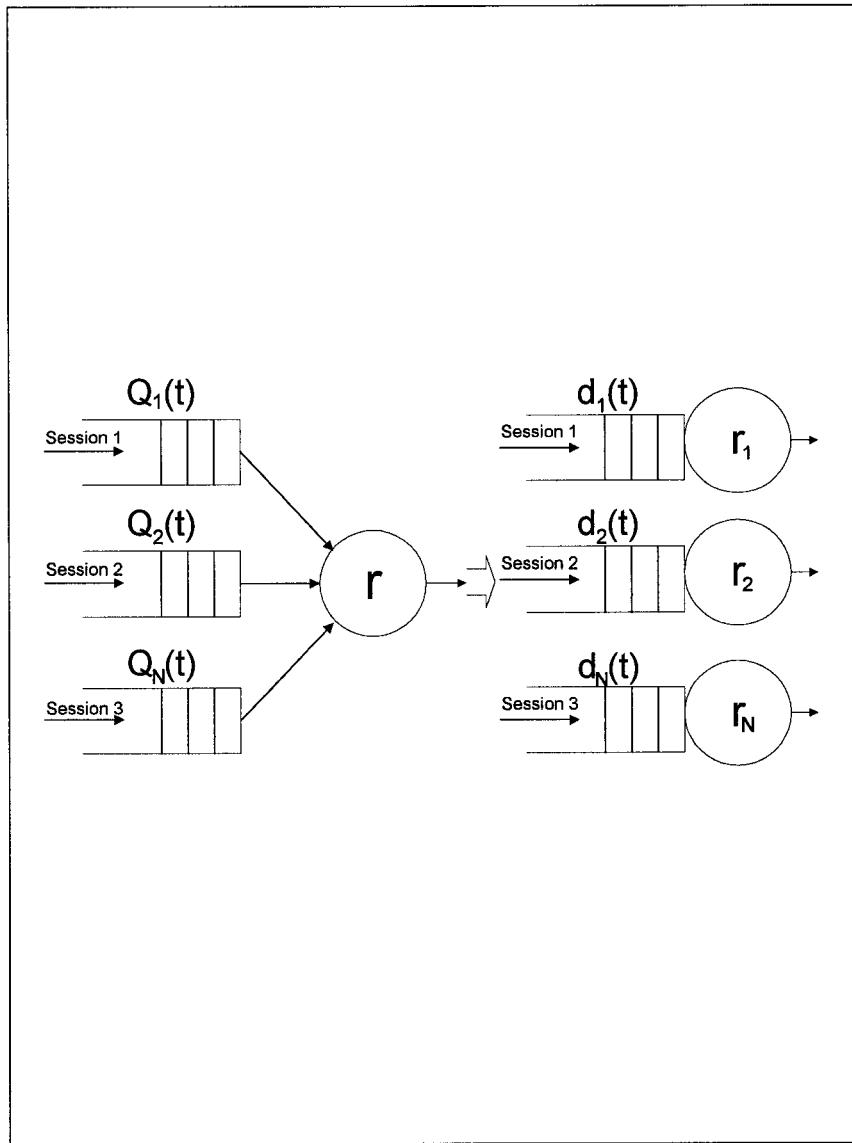


FIG. 2.2 Modèle de l'algorithme de service GPS

le cas réel, il y a plusieurs contraintes supplémentaires par rapport à ce modèle "fluide". Lorsqu'un serveur transmet un paquet, il doit terminer la transmission avant de passer à un paquet dans une autre file d'attente. Les paquets peuvent être de différentes longueurs. La transmission d'un paquet commence seulement après que le paquet ait été reçu en entier par le serveur. Les algorithmes WFQ tentent de reproduire les caractéristiques de l'algorithme GPS idéal en tenant compte de toutes ces contraintes. Deux variantes de WFQ [3] ont été proposées sous les noms suivants:

- Fair Queuing based on Starting Times (FQS): le serveur choisit le paquet qui aurait commencé le service en premier si l'algorithme idéal était utilisé
- Fair Queuing based on Finishing Times (FQF): le serveur choisit le paquet qui aurait terminé le service en premier si l'algorithme idéal était utilisé

La différence entre la quantité de bits transmise en utilisant FQS ou FQF et celle qui aurait été transmise dans le cas du GPS idéal est bornée. La borne supérieure de cette différence est égale à la longueur maximale d'un paquet dans le réseau ([3], [5]).

L'algorithme qui approxime le GPS idéal est aussi appelé Packet-by-Packet GPS (PGPS) [5]. Les disciplines FQS et le FQF sont des variantes du Packet-by-Packet GPS (PGPS).

Une autre variante du WFQ est le Worst-Case Weighted Fair Queuing ( $WF^2Q$ )

[20]. Dans  $WF^2Q$ , le processeur choisit un paquet qui aurait terminé le service le premier si l'algorithme idéal était utilisé, à condition que son temps "idéal" de début de service se situe dans le passé.

Il existe d'autres variantes du WFQ, comme le Deficit Round Robin [24] et le Frame Based Fair Queueing [10]. La première prend en compte des paquets de taille variable, alors que la deuxième offre des performances comparable à celles de PGPS tout en étant plus simple à implémenter.

Un algorithme de service plus élaboré combine des sessions GPS ayant des garanties de performance avec le trafic de moindre priorité, auquel est assigné un seul flot GPS [9]. A l'intérieur de ce dernier flot, il y a des classes de priorité, et pour chaque classe, le service est effectué en suivant une variante de l'algorithme FIFO, appelé FIFO+. Dans FIFO+, un paquet peut être avancé dans la file d'attente si son délai augmente et s'éloigne trop de la moyenne de délais subis par les autres paquets de sa classe de priorité.

En dernier, il faut mentionner l'algorithme de service Earliest Deadline First (EDF) [22,35], qui offre des garanties indépendantes de délai et de largeur de bande. Dans cet algorithme, lorsqu'un paquet arrive au routeur, celui-ci lui assigne un délai limite de traitement. Ce délai est la somme du temps d'arrivée du paquet et du délai maximum garanti pour le flot de données auquel appartient le paquet. Les paquets sont classés selon la valeur du "délai limite". Le paquet ayant la valeur la plus faible du "temps limite" est choisi pour la transmission. Ainsi, la priorité du

paquet augmente avec le temps que celui-ci passe dans le noeud. L'algorithme EDF permet de garantir le délai pour une session à faible taux de transmission.

Dans le cadre de ce mémoire, nous allons considérer un réseau qui utilise une variante de la discipline de service GPS, appellée Rate Proportional Processor Sharing (RPPS) où les poids des flots de données sont proportionnels à leurs taux de transmission. Pour cette discipline de service, des bornes peuvent être formulées pour le délai et la longueur des files d'attente. Ces bornes peuvent être déterministes [6] ou stochastiques [28].

## 2.2 Routage

L'objectif principal des protocoles de routage dans le réseau Internet est d'assurer la connectivité entre les noeuds du réseau. Les protocoles de routage utilisent l'algorithme de Bellman-Ford (Distance-Vector Protocols) ou l'algorithme des plus courts chemins de Dijkstra (Link State Protocols) pour calculer les meilleurs chemins. Dans la première catégorie de protocoles, la métrique la plus simple est le nombre de noeuds jusqu'à la destination. Le chemin le plus court est choisi pour acheminer le trafic. Plusieurs chemins ayant le même nombre de noeuds peuvent co-exister pour assurer la redondance et permettre la répartition du trafic. Un exemple de protocole à vecteur de distance est le Interior Gateway Routing Protocol (IGRP) de la compagnie Cisco, dont la métrique est une combinaison de plusieurs paramètres tels la bande passante, le délai de propagation, la charge sur le lien et sa

fiabilité. La métrique de IGRP permet d'adapter le routage en fonction des exigences de qualité de service des clients.

Dans la seconde catégorie de protocoles, le protocole Open Shortest Path First (OSPF) [8, 12, 17] utilise l'algorithme des plus courts chemins de Dijkstra pour calculer les meilleurs chemins. La métrique, appelée coût dans OSPF, est une fonction de la bande passante des liaisons. Les liaisons ayant les coûts les moins élevés sont celles qui sont les plus sollicitées pour transmettre des paquets. De plus, OSPF permet d'adapter le routage au type de trafic (selon le type de service IP) et de partager la charge entre plusieurs chemins. Des extensions du protocole OSPF [17] permettent de prendre en compte les paramètres de qualité de service, comme le délai ou la longueur des chemins.

Un seul protocole de routage, appelé Hello [11] utilise le délai comme métrique pour choisir la meilleure route. Le délai représente le temps nécessaire à un paquet pour faire un aller-retour entre la source et la destination. Le protocole Hello a été utilisé dans le réseau Distributed Computer Network (DCN) dans les années 1980, mais il a maintenant été remplacé par une version du protocole IS-IS, qui utilise l'algorithme du plus court chemin.

Pour supporter la qualité de service, les algorithmes de routage doivent fournir des chemins qui répondent aux exigences des clients. Les algorithmes de routage traditionnels ne sont pas adaptés dans ce contexte. Quand un algorithme de routage traditionnel trouve un chemin, la session est rejetée s'il ne satisfait pas les critères

de performance requis. Un algorithme de routage plus sophistiqué pourrait trouver un autre chemin qui satisfait les contraintes et éviter ainsi de rejeter la session.

Le rôle du routage dans le contexte de la qualité de service est d'assurer la connectivité, mais aussi d'augmenter le taux d'utilisation du réseau [37]. L'efficacité d'un réseau peut-être définie comme le nombre de sessions acceptées dans le réseau par rapport au nombre total de sessions entrantes.

Au cours des dernières années, il y a eu plusieurs travaux publiés dans le domaine du routage pour la qualité de service [31–33, 37, 41, 45]. Les nouveaux algorithmes de routage présentés dans ces travaux sont généralement au stade de propositions et ne sont pas encore déployés dans les réseaux de communication.

Les algorithmes de routage qui supportent la qualité de service doivent composer avec des contraintes supplémentaires, par rapport aux algorithmes de routage actuels qui sont basés sur la topologie du réseau. Ces contraintes supplémentaires sont:

- les exigences de qualité de service de chaque flot ou ensemble de flots agrégés
- les ressources disponibles dans les noeuds et les liens du réseau
- les contraintes administratives (par exemple, contrats de service avec les usagers).

Les exigences de qualité de service peuvent être de plusieurs types, par exemple: délai, gigue maximum tolérés par les usagers, largeur de bande minimale garantie,

probabilité de perte de paquets. Les ressources sont la capacité des noeuds à traiter l'information (capacité de la mémoire, vitesse du processeur) et les capacités des liens physiques. Une contrainte administrative peut être par exemple une limite de largeur de bande qu'il faut respecter en acceptant une nouvelle connexion.

Les problèmes de routage dans le contexte de la qualité de service peuvent être classés en fonction de l'objectif à atteindre [37]:

- problème d'optimisation de largeur de bande: trouver le chemin offrant la plus grande largeur de bande. La largeur de bande d'un chemin est déterminée par le lien le plus faible du chemin. Une solution à ce problème a été présentée sous le nom de l'algorithme "widest path" [45].
- les problèmes d'optimisation de coût et de délai
- les problèmes qui combinent plusieurs contraintes, comme le délai, la largeur de bande et la gigue, par exemple: trouver le chemin offrant une largeur de bande donnée et le plus faible délai possible.

Les problèmes de routage avec plusieurs contraintes sont généralement NP-complets quand les contraintes sont indépendantes [32,45]. Lorsque la discipline de service WFQ est utilisée, le délai dans les files d'attente est fonction de la largeur de bande allouée à la session [32,37]. Cette relation permet de trouver des algorithmes polynomiaux pour résoudre le problème de routage avec plusieurs contraintes. Ces

algorithmes sont basés sur des formules déterministes de délai pour les algorithmes de service de type GPS [6, 29].

Wang et Crawcroft [45] ont présenté un algorithme en deux étapes pour trouver un chemin qui respecte à la fois des contraintes de largeur de bande et de délai. Dans un premier temps, on élimine tous les liens ayant une largeur de bande insuffisante. Ensuite, l'algorithme de Dijkstra permet de trouver le chemin ayant la somme de délais la plus faible. Le chemin est accepté s'il respecte la contrainte de délai.

Les auteurs présentent aussi une autre version de cet algorithme dans laquelle la contrainte de largeur de bande est considérée prioritaire. La *largeur* d'un chemin est déterminée par le lien ayant la largeur de bande minimale. La somme des délais de propagation est appelée la *longueur* du chemin. L'algorithme trouve d'abord le chemin ayant la largeur de bande la plus élevée (chemin le plus *large*), et sélectionne ensuite le chemin ayant la somme des délais de propagation la plus faible (chemin le plus *court*). Cet algorithme est connu dans la littérature sous le nom de "shortest-widest". Quand les largeurs de bande de tous les liens sont identiques, cet algorithme est équivalent à l'algorithme des plus courts chemins.

Les performances de plusieurs heuristiques pour le routage avec contraintes de qualité de service ont été comparées dans [31, 32]. Les algorithmes suivants ont été évalués:

**widest-shortest path** choisit le chemin le plus court possible (en termes de nombre

de liens) tout en satisfaisant les contraintes de qualité de service. S'il y a plusieurs chemins de la même longueur, le chemin offrant la largeur de bande la plus élevée est choisi.

**shortest-widest path** contrairement au cas précédent, cet algorithme choisit le chemin offrant la largeur de bande la plus élevée tout en satisfaisant les contraintes de qualité de service. S'il y a plusieurs choix équivalents, le chemin le plus court est sélectionné. Cet algorithme est le même que celui décrit dans [45]

**dynamic-alternative path** commence par considérer les chemins ayant le nombre minimal de liens. Si aucun chemin ne satisfait les contraintes, les chemins ayant un lien de plus sont considérés.

**shortest-distance(P,1)** choisit le chemin qui minimise la somme suivante:

$$\sum_{i=1}^k \frac{1}{R_i} \quad (2.1)$$

ou les  $R_i$  sont les largeurs de bande des différents liens qui composent le chemin entre l'origine et la destination.

Ces algorithmes présentent un compromis entre la qualité de service offerte et l'efficacité du réseau, exprimée en termes de taux de blocage à l'entrée du réseau. Par exemple, l'algorithme "shortest-widest path" se montre quelquefois moins per-

formant car il choisit des chemins plus longs, monopolisant ainsi plus de ressources dans le réseau. Ceci se traduit par un objectif important du routage pour la qualité de service: celui de trouver un chemin qui satisfait les contraintes tout en évitant de gaspiller des ressources [4]. Le réseau pourra ainsi accomoder plus d'usagers, augmentant son efficacité.

L'algorithme "shortest-distance(P,1)" offre la meilleure performance dans le contexte de la discipline de service "max-min fair share" [33], dont le principe est semblable à celui de GPS. Dans cette discipline de service, les sessions se partagent la largeur de bande en fonction de leurs poids respectifs. Quand une session n'utilise pas toute la largeur de bande qui lui est attribuée, la partie inutilisée est partagée entre les autres sessions. La largeur de bande disponible varie dans le temps, car des sessions se terminent et des nouvelles sessions arrivent. Plus il y a de liens dans un chemin, plus il y a de chances que la largeur de bande allouée à une session diminue au cours de sa durée de vie. Ainsi, le chemin offrant la largeur de bande la plus élevée n'est pas nécessairement le meilleur, car la longueur du chemin est aussi un facteur important. L'algorithme "shortest-distance(P,1)" offre le meilleur compromis entre les objectifs suivants: la recherche du chemin le plus "large" (chemin ayant la largeur de bande la plus élevée) et du chemin le plus "court" (chemin avec le plus petit nombre de liens possible).

Dans [2], les auteurs présentent des modèles d'optimisation basés sur la formule de délai déterministe pour les réseaux GPS. Le modèle présenté considère

le problème d'allocation de taux de service aux clients, de façon à optimiser un certain objectif. Deux catégories de fonctions objectif sont présentées. En l'absence des contraintes de délai maximum, l'objectif est de minimiser le délai maximum sur l'ensemble des flots présents dans le réseau. S'il y a des contraintes de délai maximum, l'objectif peut être de maximiser le revenu généré par le réseau. Selon les auteurs, il est difficile de trouver des algorithmes efficaces pour résoudre ces problèmes à cause de la nature de la fonction objectif. De plus, ces modèles supposent que l'on connaît, en tout temps, toutes les sessions qui composent le trafic dans le réseau. En réalité, la quantité de sessions dans le réseau est trop grande et leur variation trop rapide pour qu'il soit possible d'appliquer la solution optimale à chaque fois que la charge varie.

### 2.2.1 Heuristiques pour le routage par chemins multiples

Plusieurs auteurs ont exploré l'idée du routage par chemins multiples [15, 25, 33]. Les objectifs de cette approche sont d'augmenter la probabilité de trouver un chemin qui respecte les contraintes et de réduire le temps de calcul nécessaire pour trouver ce chemin. Lorsque l'algorithme trouve plusieurs solutions, les données peuvent être acheminées sur un seul chemin [15] ou sur plusieurs chemins en parallèle [25, 33].

Cidon et al [15] proposent d'envoyer les requêtes de réservation de ressources simultanément sur plusieurs chemins, pour augmenter la probabilité d'établir la

réservation. Cette méthode permet de réduire le temps nécessaire pour trouver un chemin qui accepte la réservation de ressources. Même si la réservation réussit sur plusieurs chemins, seul le plus court chemin est ensuite utilisé pour acheminer les données. Pour éviter le gaspillage de ressources, la largeur de bande réservée sur les chemins qui ne sont pas choisis est utilisée par les autres sessions.

D'autres auteurs [33] proposent d'utiliser plusieurs chemins en parallèle pour transférer des données entre la source et la destination. Les auteurs montrent que les algorithmes qui utilisent deux chemins en parallèle offrent des performances intéressantes. Cependant, ce type de routage peut avoir un effet adverse sur le trafic sensible au délai. Par exemple, un signal audio ou vidéo pourrait être affecté par le temps nécessaire pour reconstruire le signal à partir des paquets qui arrivent en désordre.

Dans [42], les auteurs étudient un mécanisme original de réservation de ressources pour le routage par chemins multiples. Dans la méthode proposée, la source contrôle l'étendue de la recherche de chemins en envoyant des paquets "sondes" avec une quantité variable de jetons. Ces paquets se propagent vers la destination, en réservant les ressources dans les noeuds. Les jetons servent à marquer les noeuds visités. Lorsque le paquet épouse ses jetons, la recherche s'arrête. Tous les chemins sur lesquels la réservation a été établie sont utilisés pour transmettre les données.

Récemment, Taft-Plotkin et al ont suggéré [26] qu'il est avantageux de rechercher des chemins disjoints (chemins qui n'ont pas de noeuds en commun) ou des

chemins ayant le moins possible de noeuds en commun. Une explication intuitive de ce résultat est que ceci diminue la probabilité qu'un lien indésirable du point de vue de la qualité de service se retrouve à la fois dans plusieurs chemins.

### 2.3 Caractérisation du trafic

Pour calculer les bornes sur le délai ou sur la longueur des files d'attente, il faut connaître l'algorithme utilisé pour servir les files d'attente (FIFO, GPS), mais il faut aussi être en mesure de décrire le trafic entrant dans le réseau. Le trafic peut être caractérisé de deux façons: par un modèle qui décrit les caractéristiques d'une source de données, ou encore par un mécanisme de régulation appliqué à la source de données. Un modèle de trafic décrit les caractéristiques du flot de données de la source, le taux d'émission des paquets (taux fixe, processus stochastique), la distribution de la taille des paquets et celle de l'intervalle de temps entre les paquets. Un modèle peut représenter le trafic de la source à l'aide de bornes, formées par des fonctions connues. Un mécanisme de régulation de la source, quant à lui, modifie le profil du trafic, mais produit à la sortie du régulateur un flot de données aux caractéristiques connues.

Nous présentons ici un modèle de trafic et un mécanisme de régulation de source fréquemment utilisés pour la formulation de bornes de qualité de service pour l'algorithme GPS, soit le modèle Exponentially Bounded Burstiness (EBB) [27] et le mécanisme du "leaky bucket" [18].

Le modèle Exponentially Bounded Burstiness (EBB), introduit dans [27], fournit une borne sur les pointes de trafic d'une source dont le taux de transmission est aléatoire. Soit  $R(t)$  le taux de transmission instantané d'une source.  $R(t)$  est un processus stochastique  $(\rho, \Lambda, \alpha)$ -EBB si la quantité de données produite par la source dans l'intervalle de temps  $[s, t]$  possède la propriété suivante:

$$\Pr \left\{ \int_s^t R(u)du \geq \rho(t-s) + \sigma \right\} \leq \Lambda \exp^{-\alpha\sigma} \quad (2.2)$$

pour tout  $\sigma \geq 0$  et  $0 \leq s < t$ .

Cette équation signifie que la quantité de données transmise lors des pointes de trafic est bornée par une fonction exponentielle décroissante. Le paramètre  $\rho$  est appelé le taux supérieur de transmission de la source. Soit  $\mu$  la limite supérieure de l'espérance du taux de transmission  $R(t)$ , sur un intervalle de temps très long. On peut montrer [27] que:

$$\mu = \limsup_{(t-s) \rightarrow \infty} \frac{\mathbb{E} \int_s^t R(u)du}{t-s} \leq \rho \quad (2.3)$$

Ainsi,  $\rho$  est une borne supérieure sur le taux de transmission moyen de la source, d'où son appellation de taux supérieur de transmission.  $\Lambda$  est appelé le pré-facteur et  $\alpha$  est appelé le taux de décroissance de la fonction exponentielle. Le modèle EBB permet de décrire un grand nombre de processus stochastiques [27, 46]. De plus, ce modèle de source se prête bien à la formulation de bornes sur les valeurs comme la

longueur de files d'attente ou le délai subi dans un noeud ou un réseau de noeuds. Plus loin, nous présentons la formulation des bornes qui seront utilisés dans la suite de ce mémoire.

Le "token bucket", aussi appelé "leaky bucket" est un mécanisme de contrôle du trafic, permettant de réduire la congestion dans le réseau. Le principe de fonctionnement d'un "token bucket" est montré à la figure 2.3. Le trafic entrant est composé de paquets. Des permis appelés jetons (tokens) sont générés à un taux fixe  $\rho$ . Un paquet entrant dans la queue peut être transmis seulement si un nombre suffisant de jetons est disponible dans le contenant de taille  $\sigma$ . Le nombre de jetons nécessaire pour transmettre un paquet est proportionnel à la taille du paquet. Quand un paquet est transmis, le nombre de jetons correspondant est retiré du contenant.

Dans le cadre de ce mémoire, nous considérons le "leaky bucket" dans le cas des bornes déterministes sur le délai et le modèle Exponentially Bounded Burstiness (EBB) dans le cas des bornes stochastiques sur le délai. Les bornes de délai, dans le contexte des réseaux utilisant la discipline de service GPS, sont présentées dans la section suivante.

## 2.4 Bornes de délai dans GPS

L'analyse mathématique d'une discipline de service permet d'exprimer, sous forme d'équations, le comportement des files d'attente dans le temps, ainsi que

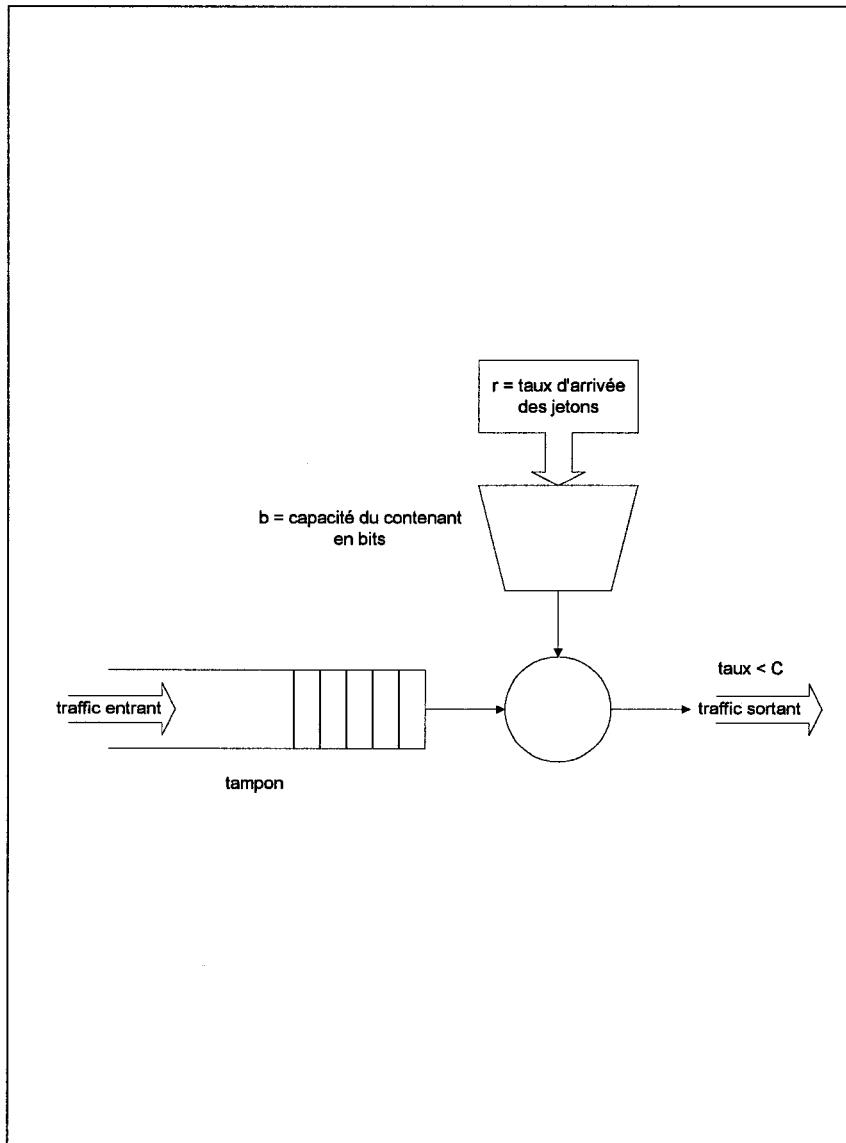


FIG. 2.3 Mécanisme du "token bucket"

le délai subi par un paquet en attente de service. Ces informations peuvent être utilisées pour le contrôle d'admission et le routage. Par exemple, les différentes routes possibles dans un réseau peuvent être classées en fonction de la borne de délai calculée pour chaque route. On peut décider d'accepter une session seulement si on trouve un chemin offrant une borne de délai inférieure à une certaine limite.

Pour la discipline de service GPS, les bornes de délai peuvent être de nature déterministe ou stochastique. Les bornes déterministes [6, 29] offrent des garanties absolues, mais au prix d'une faible utilisation des ressources. Par exemple, sur les liens qui transportent uniquement des sessions de type "Guaranteed Service" du modèle intserv, le taux d'utilisation est de seulement 40% [35]. Etant donné la nature aléatoire du trafic, les bornes déterministes sont très larges car elles doivent prendre en compte le pire cas, qui est celui où toutes les sources transmettent à leur taux maximum. L'utilisation des bornes stochastiques permet d'améliorer le taux d'utilisation du réseau, car ces bornes nécessitent moins de ressources pour chaque flot de données [39].

#### **2.4.1 Borne déterministe**

Dans le cas d'un seul noeud GPS avec toutes les sessions entrantes conformes au modèle du "leaky bucket", il est possible d'obtenir une borne supérieure déterministe sur le délai subi par le trafic [5]. Dans [6], ce résultat est appliqué au cas d'un réseau de plusieurs noeuds GPS pour obtenir des bornes sur le délai et la longueur

de files d'attente. Les auteurs montrent que dans le cas du GPS idéal (fluide), la borne sur le délai ne dépend pas du nombre de liens traversés sur le chemin emprunté par une session. De plus, cette borne ne dépend pas du comportement des autres sessions, mais seulement du poids attribué à cette session, en autant que la condition générale de stabilité soit satisfaite (équation 2.4). Cette condition stipule que la somme des taux de transmission de toutes les sessions qui entrent dans un noeud doit être inférieure au taux de service maximal du noeud [6, 28].

$$\sum_{j \in I(n)} \rho_j < r^n \quad (2.4)$$

où:

$I(n)$  est l'ensemble des sessions entrantes dans le noeud  $n$

$\rho_j$  est le taux de transmission de la session  $j$

$r^n$  est le taux de service maximal du noeud  $n$

Pour la discipline de service GPS fluide, la borne sur le délai est la suivante [6]:

$$D_i^{GPS} \leq \frac{\sigma_i}{\rho_i} \quad (2.5)$$

où:

$\sigma_i$  profondeur du "token bucket"

$\rho_i$  taux d'arrivée des jetons dans le "token bucket"

Ce résultat est ensuite appliqué au cas de réseaux utilisant la discipline de service Packet-by-Packet GPS (PGPS), dans laquelle les flots de données ne sont pas continus mais divisés en paquets. Les auteurs étudient un cas particulier de PGPS, le Rate Proportional Processor Sharing (RPPS), où le poids de chaque flot de données est proportionnel à son taux de transmission. Le délai subi par un paquet d'une session  $i$  dans RPPS est borné de façon déterministe [6] par:

$$D_i^{PGPS} \leq \frac{\sigma_i + 2(K - 1)L_i}{\rho_i} + \sum_{m=1}^K \frac{L_{max}}{r^m} \quad (2.6)$$

Ce délai maximum est fonction des paramètres suivants, en plus des  $\sigma_i$  et  $\rho_i$  du token bucket:

$K$  nombre de liens dans le chemin entre la source et la destination

$r^m$  taux de service du noeud  $m$

$L_i$  longueur d'un paquet de la session  $i$

$L_{max}$  longueur maximale d'un paquet dans le réseau

#### 2.4.2 Borne stochastique

Lorsque l'algorithme de service GPS est utilisé dans un réseau et le trafic entrant est caractérisé par le modèle EBB, on peut calculer des bornes stochastiques

sur la longueur des files d'attente et le délai [28, 30, 46]. Ces bornes sont moins larges que les bornes déterministes de Parekh et Gallager [5, 6]. Elles permettent de multiplexer plus de sessions, donc de passer plus de trafic dans le réseau. Ces bornes stochastiques sont applicables lorsque les sessions peuvent tolérer une certaine probabilité de dégradation de service. Le multimédia est un exemple d'application qui exige une grande largeur de bande mais qui peut tolérer une certaine probabilité de délai et de perte de paquets, dont les effets seront imperceptibles pour l'usager.

Dans [46], les auteurs présentent une borne stochastique pour un réseau utilisant l'algorithme de service Rate Proportional Processor Sharing (RPPS), variante de Generalized Processor Sharing (GPS) dans laquelle le poids  $\phi_i$  de chaque flot de données  $i$  est égal à son taux de transmission  $\rho_i$ :

$$\phi_i = \rho_i \quad (2.7)$$

La borne stochastique est valide sous les hypothèses suivantes:

- toutes les sessions  $i$  sont des processus  $(\rho_i, \Lambda_i, \alpha_i)$ -EBB
- dans tous les noeuds du réseau, la somme des taux de transmission de toutes les sessions entrantes dans le noeud est inférieure au taux de service maximal

du noeud

$$\sum_{j \in I(n)} \rho_j < r^n \quad (2.8)$$

où:

$I(n)$  est l'ensemble des sessions entrantes dans le noeud  $n$

$\rho_j$  est le taux de transmission de la session  $j$

$r^n$  est le taux de service maximal du noeud  $n$

Cette condition de stabilité est la même que celle pour la borne déterministe (équation 2.4). Cependant, le taux de transmission dans le cas déterministe est borné par le taux d'arrivée des jetons dans le leaky bucket, alors que dans le cas stochastique, il est un paramètre du modèle EBB de la source.

La borne sur le délai total subi par une session  $i$  dans le réseau est donnée [46]

par la formule:

$$P\{D_i^{net}(t) \geq d\} \leq \lambda_i^{net} e^{-\alpha_i g_i^{net} d} \quad (2.9)$$

Le paramètre  $g_i^{net}$  est le taux de service minimum alloué à la session  $i$  le long du chemin  $P(i)$  de l'origine à la destination. Ce taux dépend du poids de la session  $i$  mais aussi des poids des autres sessions présentes au noeud.

$$g_i^{net} = \min_{n \in P(i)} g_i^n \quad (2.10)$$

$$g_i^n = \frac{\phi_i^n}{\sum_{j \in I(n)} \phi_j^n} r^n \quad (2.11)$$

La fonction  $\lambda_i^{net}$  dépend des paramètres de source ( $\rho_i, \Lambda_i, \alpha_i$ ) ainsi que des coefficients GPS attribués à la session  $i$  dans chaque noeud du chemin de l'origine à la destination.

$$\lambda_i^{net} = \frac{\lambda_i e^{\alpha_i \rho_i \xi}}{1 - e^{-\alpha_i (g_i^{net} - \rho_i) \xi}} \quad (2.12)$$

où  $\xi$  est un réel positif qui satisfait la condition suivante:

$$0 < \xi < \frac{\ln(\Lambda_i + 1)}{\alpha_i (g_i^{net} - \rho_i)} \quad (2.13)$$

#### 2.4.3 Comparaison des deux bornes

Dans le cadre de ce mémoire, nous allons considérer la borne de délai déterministe (équation 2.5) et la borne de délai stochastique (équation 2.9). Les deux formules sont applicables dans le cas du trafic fluide, lorsque la taille des paquets est négligeable.

Pour permettre la comparaison des deux bornes, le trafic entrant doit être représenté de façon équivalente dans les deux cas. La formule déterministe est ap-

plicable lorsque les sources sont contrôlées par un "token bucket", alors que la formule stochastique suppose que les sources sont des processus EBB. Pour établir une comparaison, on utilise des sources de type ON-OFF, pour lesquelles il existe une caractérisation EBB [46]. Cependant, il faut choisir les paramètres du "token bucket" de façon à ne pas changer le comportement de la source, ce qui le rend en fait inutile du point de vue du contrôle du débit de données. Cette hypothèse est nécessaire pour pouvoir effectuer la comparaison; cependant, la borne déterministe sur le délai risque d'être plus large que dans le cas où on laisse agir le "token bucket" pour réduire les pointes de trafic.

## CHAPITRE 3

### MODÈLES DE ROUTAGE PROPOSÉS

Le routage basé sur les paramètres de qualité de service permet d'améliorer la performance et l'efficacité des réseaux de communications par paquets [40]. Les travaux précédents qui combinent les aspects de routage et de la qualité de service ont porté surtout sur les heuristiques de routage avec contraintes multiples [31–33, 37, 41, 45]. Ces travaux ont considéré des contraintes liées à la topologie du réseau, par exemple le nombre de liens dans un chemin et le délai de propagation sur le chemin, et des contraintes liées à la charge sur le réseau, comme le délai de traitement dans les noeuds. Les heuristiques proposées sont basées la plupart du temps sur le modèle déterministe de la discipline de service GPS et de ses variantes, comme WFQ [6, 29, 32, 45]. Le modèle déterministe offre une garantie stricte de délai, au prix d'une faible utilisation des ressources [35]. L'utilisation d'un modèle stochastique pour l'algorithme de service WFQ permet d'augmenter l'efficacité du réseau [39]. Toutefois, à notre connaissance, il n'existe pas de travaux qui étudient le routage basé sur la qualité de service en relation avec les disciplines de service dans les noeuds du réseau. Dans ce travail, nous proposons un algorithme de routage basé sur le modèle stochastique de la discipline de service GPS. Nous allons

nous intéresser au problème suivant:

*Etant donné un réseau et un ensemble de sessions déjà présentes dans le réseau, choisir un chemin pour une nouvelle session de façon à optimiser sa qualité de service*

Le critère de qualité de service est utilisé pour le contrôle d'admission et pour le routage. Une session est rejetée si les chemins disponibles ne satisfont pas le critère de QS: c'est la partie *contrôle d'admission* de l'algorithme. Si la session est acceptée, elle est placée sur le chemin qui offre la meilleure qualité de service: c'est la partie *routage* de l'algorithme.

La critère de qualité de service dépend du modèle utilisé pour décrire le trafic et la stratégie de service des files d'attente. Dans le cas de la discipline de service GPS, le critère de qualité de service peut être de nature déterministe (délai maximum) ou stochastique (seuil acceptable de la probabilité de dépassement de ce délai). Selon la nature de ce critère, nous allons considérer les algorithmes de routage suivants:

**routage avec critère déterministe de QS :** le critère de qualité de service est basé sur la borne de délai *déterministe* de GPS [6]

**routage avec critère stochastique de QS :** le critère de qualité de service est basé sur la borne de probabilité de délai du modèle *stochastique* de GPS [46]

**routage "stochastique+" basé sur la charge moyenne :** comme le routage **stochastique**, mais le calcul est basé en partie sur l'état présent du réseau et en partie sur un état moyen (sur une longue période par exemple)

Comme point de référence, nous allons utiliser un algorithme de routage basé uniquement sur la topologie du réseau (algorithme de plus court chemin), car c'est ce type de routage qui prévaut aujourd'hui dans le réseau Internet.

Dans le cas du modèle stochastique de GPS, les calculs de routage sont basés seulement sur l'état courant du réseau au moment où arrive une nouvelle session. Nous nous sommes demandé s'il est avantageux de considérer la charge "moyenne" sur le réseau au moment d'acheminer les sessions. Pour répondre à cette question, nous avons créé une variante de l'algorithme "stochastique", appelée "stochastique+". Cet algorithme peut être qualifié de "prédictif", car il tient compte de l'état *présent* du réseau et de son état *moyen*. L'état moyen, qui est une donnée du problème, représente une charge typique sur le réseau. Cet état moyen peut être basé sur l'observation de la charge sur tous les noeuds du réseau, sur une période de temps assez longue. L'algorithme "stochastique+", qui représente une des contributions importantes de ce mémoire, permettra de déterminer si la connaissance de la charge (présente et future) du réseau influence le choix des chemins et l'efficacité du réseau.

### 3.1 Routage avec critère déterministe de QS

L'algorithme de routage basé sur le modèle déterministe de GPS choisit le chemin qui minimise le *délai* de la nouvelle session, dans un ensemble de chemins "candidats". L'ensemble de chemins peut contenir un ou plusieurs chemins générés par les algorithmes d'énumération des chemins qui seront présentés dans le chapitre suivant.

Dans le modèle déterministe de GPS, la source de trafic est contrôlée par un "leaky bucket" (LB) qui a pour effet de limiter les pointes de trafic. Pour pouvoir établir une comparaison avec le modèle stochastique, les paramètres du "leaky bucket" doivent être choisis de façon à ne pas modifier le comportement aléatoire de la source. Le tableau 3.1 présente quelques exemples de sources ON-OFF avec les caractérisations EBB correspondantes, ainsi que le choix des paramètres du "leaky bucket". Les caractéristiques de sources ON-OFF sont tirées d'exemples présentés dans [46]. Le taux d'arrivée des jetons dans le "bucket" ( $LB_{taux}$ ) est égal au taux de transmission de la source dans l'état "ON" ( $\lambda_i$ ). La profondeur du "bucket" est choisie de façon à respecter le délai maximum acceptable pour la session (équation ??). Ainsi, le profil du trafic d'une source ON-OFF n'est pas modifié à l'entrée du réseau.

Le délai maximum permis pour la session est respecté quel que soit le chemin choisi, puisqu'il ne dépend que des paramètres du "leaky bucket". Cependant, le

TAB. 3.1 *Sources de trafic et leurs paramètres EBB et "leaky bucket"*

| Source | $p_i$ | $q_i$ | $\lambda_i$ | $\bar{\lambda}_i$ | $\rho_i$ | $\Lambda_i$ | $\alpha_i$ | $LB_{taux}$ |
|--------|-------|-------|-------------|-------------------|----------|-------------|------------|-------------|
| 1      | 0.3   | 0.7   | 0.5         | 0.15              | 0.2      | 1.0         | 1.74       | 0.5         |
| 2      | 0.4   | 0.4   | 0.4         | 0.2               | 0.25     | 0.92        | 1.76       | 0.4         |
| 3      | 0.3   | 0.3   | 0.3         | 0.15              | 0.2      | 0.84        | 2.13       | 0.3         |
| 4      | 0.4   | 0.6   | 0.5         | 0.2               | 0.25     | 1.0         | 1.62       | 0.5         |

chemin choisi doit respecter la condition générale de stabilité (équation 2.4), qui exige que la somme des taux de transmission de toutes les sessions qui passent par un noeud soit inférieure au taux de service du noeud [6].

L'algorithme de routage dans le cas du modèle déterministe se réduit donc à examiner les chemins disponibles et trouver un chemin où la condition de stabilité est respectée dans tous les noeuds. Le premier chemin qui respecte la condition de stabilité est choisi. Si aucun des chemins ne respecte cette condition, la session est rejetée.

### 3.2 Routage avec critère stochastique de QS

L'algorithme de routage basé sur le modèle stochastique de GPS choisit le chemin qui minimise la *probabilité de délai* de la nouvelle session.

Dans le cas du modèle stochastique de GPS, une source de trafic  $i$  est décrite par les paramètres EBB ( $\rho_i, \lambda_i, \alpha_i$ ) . Le paramètre  $\rho_i$  représente le taux de transmission de la source, alors que les autres paramètres sont une mesure de son caractère aléatoire [27, 28]. Dans le chapitre précédent, nous avons présenté le calcul de la

borne de probabilité de délai basée sur le modèle de trafic EBB (équations 2.7, 2.8 et 2.9).

Le choix d'un chemin pour une nouvelle session est effectué en deux étapes: tout d'abord, on trouve un ensemble de chemins possibles entre l'origine et la destination. Ensuite, on calcule la probabilité de délai de la nouvelle session pour tous les chemins possibles. Les chemins qui ont une probabilité de délai supérieure à la limite sont rejétés: c'est la composante *contrôle d'admission* de cet algorithme. On choisit le chemin qui offre la probabilité de délai la plus faible pour la nouvelle session: c'est la composante *routage* de l'algorithme.

Le contrôle d'admission est ajouté à l'algorithme de routage proprement dit pour assurer que le chemin choisi offre des performances raisonnables aux usagers du réseau. Il faut remarquer que l'algorithme de routage peut fonctionner sans le contrôle d'admission. Dans ce cas, il choisit le chemin offrant la probabilité de délai minimale. Cependant, cette probabilité de délai pourrait être élevée, ce qui augmenterait les chances de dépasser le délai maximum. Comme le but du modèle stochastique est d'améliorer l'efficacité du réseau tout en offrant une qualité de service satisfaisante, la composante de contrôle d'admission est essentielle pour assurer cette qualité.

### 3.3 Routage "stochastique+" avec considération de la charge moyenne sur le réseau

Afin de chercher à améliorer l'efficacité du réseau, nous proposons une modification de l'algorithme de routage stochastique. Cette modification consiste à baser la décision de routage non seulement sur l'état présent du réseau mais, aussi, sur son état *moyen*. Cet état moyen est une donnée du problème. Dans la réalité, il peut être basé sur l'observation de la charge sur le réseau sur une période assez longue, par exemple quelques jours. Considérons un exemple du calcul du taux d'occupation moyen  $S_{moy}$  dans un noeud  $k$ , initialement vide. Supposons que 3 sessions de même durée sont ajoutées au noeud  $k$ , à intervalles de temps égaux, avec des taux de transmission correspondant à 20%, 40% et 10% de la capacité de traitement maximale du noeud. Lorsque la session 1 arrive, le taux d'occupation  $S$  est égal à 20%. Lorsque la session 2 arrive,  $S$  devient égal à  $20\% + 40\% = 60\%$ . Supposons que lorsque la session 3 arrive, seule la session 2 est encore présente, alors  $S$  est égal à  $40\% + 10\% = 50\%$ . Ainsi, la moyenne des taux d'occupation  $S_{moy}$ , sur les trois intervalles de temps considérés, est égale à:

$$(20\% + 60\% + 50\%)/3 = 43.33\%$$

La charge moyenne sur un noeud, calculée selon l'exemple ci-haut, sera utilisée dans la présente version de l'algorithme en combinaison avec la charge courante sur le noeud. Le poids relatif de l'état présent et de l'état moyen du réseau dans le

calcul est un paramètre de l'algorithme. En variant ce paramètre, on peut déterminer l'effet de la connaissance de l'état moyen du réseau sur les performances et l'efficacité du réseau.

Le principe de fonctionnement de cet algorithme peut être illustré par l'exemple suivant. Prenons un noeud  $n$  du réseau et supposons qu'il soit le noeud ayant le taux de service minimal sur un chemin donné. Supposons que la charge moyenne sur ce noeud, sur une période assez longue, est de 70% de sa capacité maximale. Supposons que l'on veuille acheminer une session  $i$  par ce noeud. Supposons aussi qu'au moment où la session  $i$  entre dans le réseau, la charge sur le noeud  $n$  est de 20%. Si l'on considère uniquement l'état courant du réseau, la charge de 20% paraît assez faible. Etant donné les caractéristiques de GPS, une faible charge sur ce noeud garantit un taux de service élevé à la session, ce qui rend le chemin attrayant. Supposons maintenant que dans les calculs, on tient compte de la charge moyenne sur le noeud  $n$ , en plus de sa charge instantanée, avec des coefficients de 0.6 et 0.4 respectivement. Au moment d'acheminer la session, la charge sur le noeud  $n$  devient:

$$(70\% * 0.6) + (20\% * 0.4) = 42\% + 8\% = 50\%$$

Maintenant, la charge calculée sur le noeud  $n$  est de 50% de sa capacité maximale, au lieu de 20% auparavant, ce qui rend le chemin moins attrayant. L'algorith-

rithme de routage va peut-être trouver un autre chemin pour la session, ce qui peut contribuer à réduire la charge moyenne sur le noeud  $n$  et le taux de blocage des sessions.

Le modèle stochastique+ est lui-même subdivisé en trois algorithmes, qui diffèrent par la façon d'utiliser l'information sur l'état courant et l'état moyen du réseau dans les composantes "contrôle d'admission" et "routage" de chaque algorithme. Rappelons que dans le cadre de ce mémoire, l'algorithme de choix d'un chemin est constitué de trois étapes:

1. recherche d'un ensemble de chemins possibles;
2. contrôle d'admission;
3. recherche du meilleur chemin selon les critères de qualité de service.

Définissons les termes suivants: soit  $S_{cour}$  l'ensemble des valeurs *courantes* de charge sur les noeuds et  $S_{moy}$  l'ensemble des valeurs de charge sur les noeuds dans l'état *moyen* du réseau. Soit  $0 \leq c \leq 1$ , le poids accordé à l'état courant dans les calculs. Alors,  $(1 - c)$  représente le poids accordé à l'état moyen. Soit  $P(d)$  la probabilité de dépasser le délai maximum permis. Avec ces définitions, nous pouvons maintenant présenter les différents algorithmes de type "stochastique+".

### 3.3.1 Algorithme stochastique+ version 1

Dans la première version, on utilise une combinaison linéaire de la charge courante  $S_{cour}$  et de la charge moyenne  $S_{moy}$  de chaque noeud pour calculer la probabilité de délai:

$$P(d) = f(cS_{cour} + (1 - c)S_{moy})$$

Cette valeur de probabilité de délai sert ensuite pour le calcul du contrôle d'admission et pour le routage.

TAB. 3.2 Sommaire de l'algorithme stochastique+ version 1

|  |
|--|
| $P(d) = f(cS_{cour} + (1 - c)S_{moy})$   |
| <b>Contrôle d'admission:</b> basé sur $P(d)$<br><b>Routage:</b> $\min P(d)$ sur l'ensemble des chemins possibles |

### 3.3.2 Algorithme stochastique+ version 2

Dans la deuxième version, on calcule la probabilité de délai basée sur l'état courant  $P_1(d)$ , suivie de la probabilité de délai basée *uniquement* sur l'état moyen  $P_2(d)$ . Les deux probabilités sont ensuite combinées dans les proportions dictées par le coefficient  $k$ , pour obtenir  $P(d)$ .

Le contrôle d'admission est basé sur l'état courant  $P_1(d)$ , alors que le routage est basé sur  $P(d)$ . Ainsi, on a la certitude que les exigences de qualité de service de chaque session sont respectées au moment où la session arrive dans le réseau.

TAB. 3.3 *Sommaire de l'algorithme stochastique+ version 2*

|  |
|--|
| $P_1(d) = f(S_{cour})$<br>$P_2(d) = f(S_{moy})$<br>$P(d) = cP_1(d) + (1 - c)P_2(d)$                                |
| <b>Contrôle d'admission:</b> basé sur $P_1(d)$<br><b>Routage:</b> $\min P(d)$ sur l'ensemble des chemins possibles |

### 3.3.3 Algorithme stochastique+ version 3

Dans cette version, le contrôle d'admission et le routage sont tous les deux basés sur la valeur de  $P(d)$  qui est une combinaison de la probabilité de délai basée sur l'état courant et de celle basée sur l'état moyen.

TAB. 3.4 *Sommaire de l'algorithme stochastique+ version 3*

|  |
|--|
| $P_1(d) = f(S_{cour})$<br>$P_2(d) = f(S_{moy})$<br>$P(d) = cP_1(d) + (1 - c)P_2(d)$                              |
| <b>Contrôle d'admission:</b> basé sur $P(d)$<br><b>Routage:</b> $\min P(d)$ sur l'ensemble des chemins possibles |

### 3.4 Routage par le plus court chemin

Le routage utilisé aujourd'hui dans le réseau Internet est basé généralement sur l'algorithme de plus court chemin, avec des métriques variées. Par exemple,

dans le protocole Open Shortest Path First (OSPF), le coût d'un lien, couramment appelé "interface", est inversement proportionnel à sa capacité exprimée en bits par seconde [8]. Dans ce mémoire, ce type d'algorithme sera évalué seulement dans le but d'obtenir une base de comparaison pour les algorithmes plus sophistiqués, basés sur la qualité de service.

Dans le cadre de ce mémoire, nous allons considérer le chemin le plus court en termes de nombre de liens, ce qui revient à dire que tous les liens ont le même coût. Les plus courts chemins pour toutes les paires de noeuds sont obtenus en utilisant l'algorithme de Floyd. Puisqu'il n'y a qu'un seul chemin disponible pour une session, si ce chemin n'offre pas une qualité de service satisfaisante, la session est bloquée.

### 3.5 Critères de comparaison des algorithmes de routage

Lorsqu'une session est placée dans le réseau, elle occupe une partie des ressources disponibles. A mesure que de nouvelles sessions s'ajoutent, le réseau se remplit, et le taux de blocage de sessions augmente. Le critère de comparaison des différents algorithmes de routage est le taux de blocage de sessions, soit la fraction de toutes les sessions entrantes qui sont rejetées. On peut parler aussi d'*efficacité* du réseau, qui est une mesure inversement proportionnelle au taux de blocage: plus le taux de blocage est faible, plus le réseau est efficace car il utilise mieux les ressources disponibles.

Dans ce chapitre, nous avons présenté les différents algorithmes de routage qui sont l'objet de ce mémoire. Chaque algorithme évalue un ensemble de chemins possibles pour une nouvelle session pour trouver le chemin qui offre les meilleures caractéristiques de qualité de service. Pour mesurer la qualité de service offerte sur un chemin, on utilise le modèle déterministe et le modèle stochastique de GPS. Dans le cas du modèle stochastique, nous proposons de baser les calculs en partie sur l'état moyen du réseau. Le tableau 3.5 présente un résumé des différents algorithmes de routage qui seront comparés dans la suite de ce mémoire.

TAB. 3.5 *Algorithmes de routage*

| Algorithme            | Description   |
|-----------------------|---|
| (1) PCC, dét          | plus court chemin, délai déterministe   |
| (2) dét               | énumération de chemins disjoints, délai déterministe  |
| (3) stoch             | énumération de chemins disjoints, délai stochastique  |
| (4) stoch+, $c = 0.9$ | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 90% et l'état moyen à 10% |
| (5) stoch+, $c = 0.7$ | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 70% et l'état moyen à 30% |
| (6) stoch+, $c = 0.5$ | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 50% et l'état moyen à 50% |
| (7) stoch+, $c = 0.3$ | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 30% et l'état moyen à 70% |
| (8) stoch+, $c = 0.1$ | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 10% et l'état moyen à 90% |

Dans le but d'évaluer les performances de ces algorithmes de routage, nous avons développé un programme de simulation, dont les différents éléments sont présentés dans le chapitre suivant.

## CHAPITRE 4

### SIMULATION

Le problème de routage, défini dans le chapitre précédent, est basé sur les formules de délai [6] et de probabilité de délai [46] pour la discipline de service GPS. Le problème d'optimisation ne peut être résolu de façon analytique à cause de la nature de la formule de probabilité de délai dans le cas du modèle stochastique de GPS. Dans l'expression de la probabilité de délai, il y a un élément qui rend le problème difficile à résoudre par les méthodes mathématiques. Cet élément est le minimum des taux de service alloués à une session dans les noeuds du chemin. À cause de la présence de ce minimum, la fonction est non-linéaire et non-différentiable. Ainsi, pour résoudre le problème de routage, nous avons développé un programme de simulation plutôt que de résoudre directement le problème d'optimisation. Le simulateur est composé d'éléments suivants:

- un réseau
- un ensemble de sessions qui composent le trafic entrant dans le réseau
- les algorithmes de recherche de chemins pour générer un ensemble de chemins possibles
- les algorithmes de routage pour choisir un des chemins possibles

Dans le cadre de ce mémoire, une session de communication est définie par les paramètres suivants:

1. noeud origine
2. noeud destination
3. paramètres du trafic de la source
4. durée de vie
5. exigences de qualité de service

Au début de la simulation, le réseau ne contient aucune session. Les sessions arrivent dans le réseau à un taux fixe, spécifié avant le début de la simulation. Les demandes de qualité de service des sessions sont exprimées en termes de délai maximum ou de probabilité de délai maximale que l'usager est prêt à tolérer. Le simulateur évalue un ensemble des chemins possibles pour une session qui entre dans le réseau, choisissant le chemin qui offre la meilleure qualité de service selon l'algorithme de routage utilisé. La figure 4.1 montre les étapes exécutées à l'arrivée d'une nouvelle session. Les sessions ont une durée de vie limitée. Lorsqu'une session se termine, l'état du réseau est mis à jour pour prendre en compte les ressources libérées par la session. Le programme récolte aussi les statistiques comme le nombre de sessions acceptées dans le réseau et le nombre total d'arrivées pour mesurer l'efficacité du réseau.

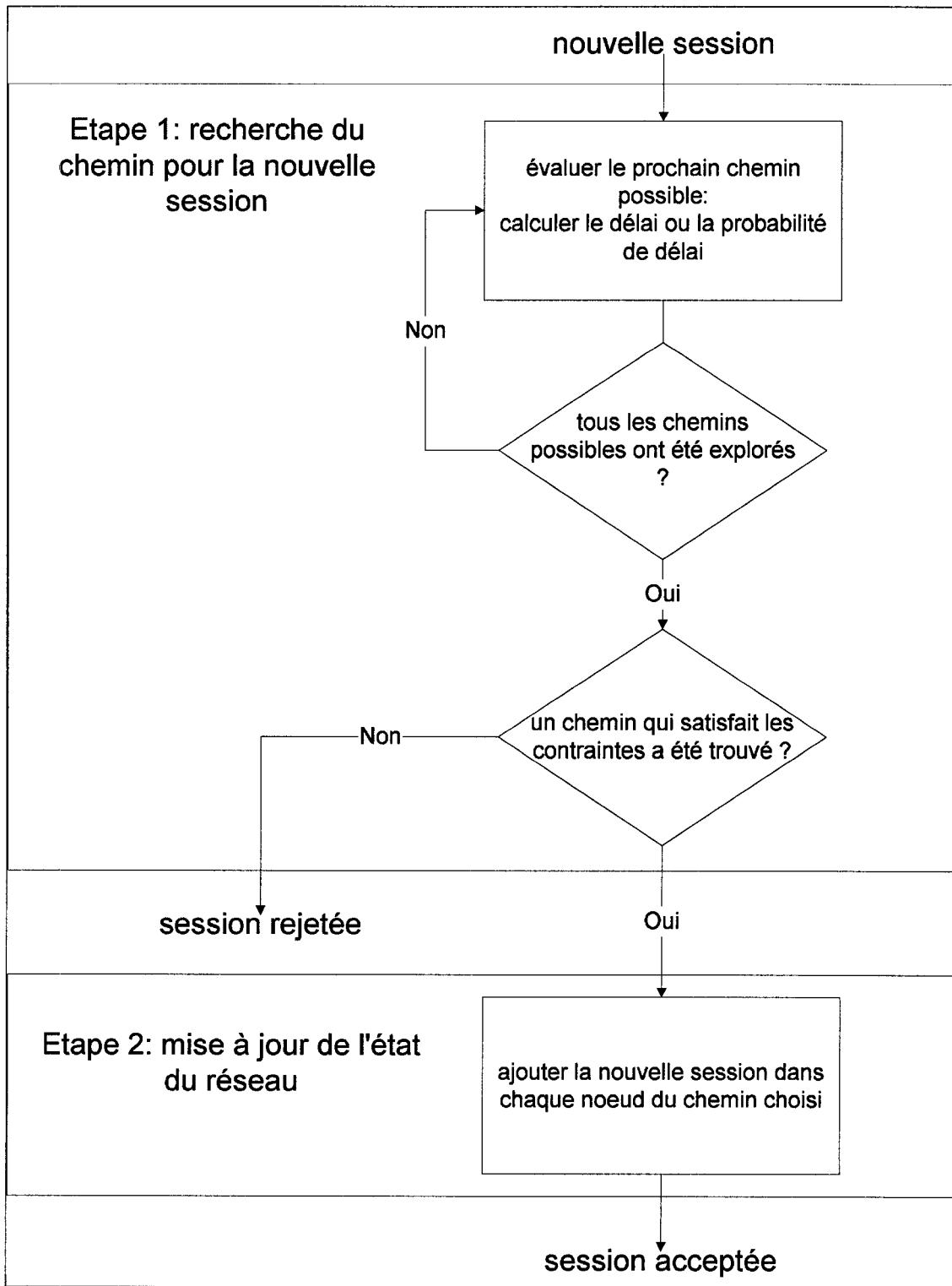


FIG. 4.1 *Étapes de l'ajout d'une session dans le réseau*

L'ensemble des chemins candidats pour une session qui arrive dans le réseau est un élément clé de la simulation. Le contenu de cet ensemble détermine en grande partie les performances et l'efficacité du réseau. L'approche la plus simple est de considérer *tous* les chemins possibles pour une paire de noeuds. Cette approche garantit aussi le choix du meilleur chemin pour une nouvelle session. Cependant, l'énumération de tous les chemins n'est possible que pour les réseaux de petite taille (moins d'une vingtaine de noeuds). Pour les réseaux plus grands, le nombre de chemins pour une paire de noeuds est trop grand pour permettre leur analyse. Pour des réseaux de quelques dizaines de noeuds, on peut avoir des centaines de milliers de chemins possibles pour une paire origine-destination. Pour pouvoir effectuer les calculs dans un temps raisonnable, il faut restreindre le nombre de chemins à évaluer en considérant un sous-ensemble de tous les chemins possibles. Un tel sous-ensemble des chemins doit offrir un bon compromis entre le nombre de chemins à considérer et la performance obtenue. La suite du présent chapitre présente la méthode utilisée dans ce mémoire pour résoudre ce problème de recherche de chemins.

#### 4.1 Génération de topologie du réseau

Le programme de simulation est basé sur les paramètres d'entrée suivants:

1. la topologie du réseau (les noeuds, les liens, la capacité de traitement de chaque noeud)

## 2. les caractéristiques du trafic de chaque session entrant dans le réseau

La topologie du réseau est représentée sous forme d'une liste d'arêtes, produite automatiquement par un programme de génération de graphes. Ce programme permet de créer des graphes orientés ou non-orientés de différentes tailles. Afin de pouvoir contrôler la densité du graphe, on peut spécifier séparément le nombre de noeuds et le nombre de liens. Le générateur attribue de façon aléatoire une valeur de capacité de traitement à chaque noeud du réseau, à l'intérieur de limites spécifiées.

La topologie du réseau et les capacités de traitement de chaque noeud ont une influence déterminante sur le routage. Afin de faciliter les modifications de la topologie et des capacités des noeuds, le graphe généré est sauvegardé dans un fichier texte. Le fichier contient la liste de noeuds avec leurs capacités de traitement, suivie du graphe représenté sous la forme d'une liste d'arêtes.

Le fichier représentant le graphe peut servir de point de départ pour générer un nouveau graphe. Le nouveau graphe peut avoir plus d'arêtes que le graphe d'origine. Ceci va être utilisé pour étudier l'effet de la densité du réseau sur la performance des algorithmes de routage.

Une fois la topologie du réseau générée, on peut appliquer les algorithmes de recherche des chemins qui sont présentés dans la section suivante.

## 4.2 Recherche de chemins

Les algorithmes de routage basés sur les considérations de qualité de service ont pour objectif de trouver la *méilleure* route pour une nouvelle session parmi toutes les routes possibles. Par exemple, dans le cas du modèle déterministe de WFQ, la meilleure route est celle qui minimise le délai pour la nouvelle session. Il est clair que si l'on examine tous les chemins possibles, on est assuré de pouvoir choisir le meilleur chemin en termes de Qualité de Service.

### 4.2.1 Énumération de tous les chemins

Dans une première étape, nous avons développé une version récursive de l'algorithme de recherche de chemins de Migliore et al [23], afin de trouver tous les chemins sans cycle pour une paire origine-destination.

Cet algorithme permet de trouver le *mieux* chemin pour une nouvelle session, étant donné les contraintes de Qualité de Service. Cet algorithme est utilisable pour les réseaux de petite taille ( $<20$  noeuds). Pour des réseaux plus grands, il faut trouver une alternative plus rapide mais offrant quand même de bons résultats. Pour cela, on peut exploiter une propriété de la discipline de service GPS. En effet, la probabilité de délai dans un réseau utilisant GPS ne dépend pas du nombre de noeuds dans le chemin, mais seulement du noeud offrant le taux de service le plus faible. Ainsi, parmi tous les chemins générés, tous les chemins contenant un noeud

---

**Algorithme 1 Algorithme ENUM: *Tous\_les\_chemins(courant, orig, dest)***

---

```

1: si tous les adjacents du noeud courant ont été explorés alors
2:   si noeud courant = noeud origine alors
3:     fin du programme
4: sinon
5:   sélectionner le prochain noeud adjacent pas encore exploré
6:   répéter
7:     si le prochain noeud n'a pas encore été visité alors
8:       si prochain noeud = destination alors
9:         imprimer la solution courante {chemin origine-destination}
10:      sinon si prochain noeud = origine alors
11:        cycle
12:      sinon
13:        inclure le prochain noeud dans la solution courante
14:        explorer les adjacents du prochain noeud:
15:          Tous_les_chemins(prochain, orig, dest)
16:          si noeud courant = origine alors
17:            enlever la marque "visité" sur tous les noeuds
18:            avancer au prochain noeud adjacent
19:          jusqu'à tous les noeuds dans la liste d'adjacence ont été explorés

```

---

TAB. 4.1 *Quelques chemins possibles entre les noeuds 13 et 6 dans le réseau de la figure 4.2*

| Chemin            |
|-------------------|
| 13 12 9 8 7 6     |
| 13 12 9 8 7 5 6   |
| 13 12 9 8 5 7 6   |
| 13 12 9 8 5 6     |
| 13 12 9 4 5 8 7 6 |

"faible" du point de vue de la qualité de service sont en fait redondants. On peut illustrer cette propriété par l'exemple suivant. Examinons un sous-ensemble des chemins (table 4.1) trouvés par l'algorithme ENUM pour le réseau de la figure 4.2:

Tous ces chemins contiennent le noeud 12. Supposons que le noeud 12 est celui qui offre le taux de service le plus bas, il est alors inutile d'évaluer tous les chemins qui contiennent ce noeud. En effet, tous ces chemins auront la même probabilité de délai. Il faut plutôt trouver un chemin qui ne contient pas le noeud 12 pour essayer d'obtenir un meilleur taux de service minimum.

Cette propriété de la formule de probabilité de délai permet de penser qu'il serait intéressant d'évaluer des chemins *disjoints*, c'est à dire des chemins qui n'ont pas de noeuds en commun autres que l'origine et la destination. Cela réduit la quantité de chemins à évaluer, et les chemins sélectionnés permettent d'obtenir de bonnes performances. La section suivante présente une modification de l'algorithme ENUM 1 qui permet d'obtenir des chemins disjoints.

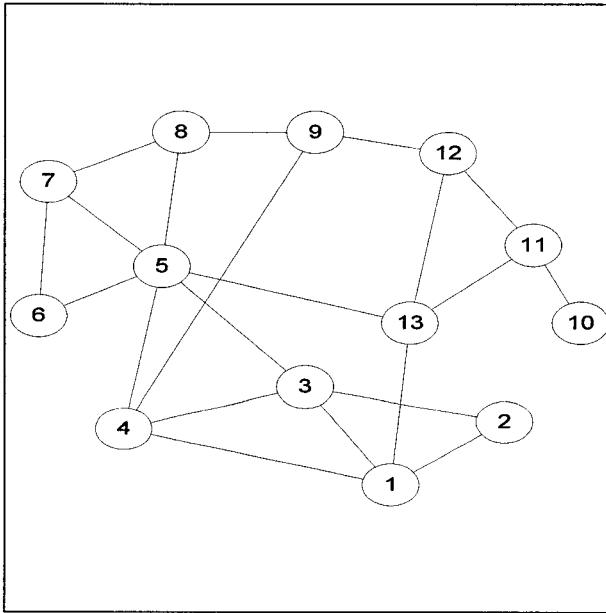


FIG. 4.2 Réseau 1

#### 4.2.2 Énumération des chemins disjoints

L'algorithme de chemins disjoints ENUM\_D permet de générer rapidement plusieurs chemins possibles, même pour des réseaux de grande taille. Il faut remarquer que l'algorithme des chemins disjoints ne trouve pas nécessairement le chemin optimal dans le contexte du modèle stochastique de WFQ, c'est à dire le chemin qui offre la probabilité de délai la moins élevée. Toutefois, il permet d'obtenir rapidement des résultats même pour des réseaux de grande taille, pour lesquels il est impossible de calculer la solution optimale dans un temps raisonnable.

L'exemple de la figure 4.3 illustre le fait que l'algorithme ENUM-D ne trouve

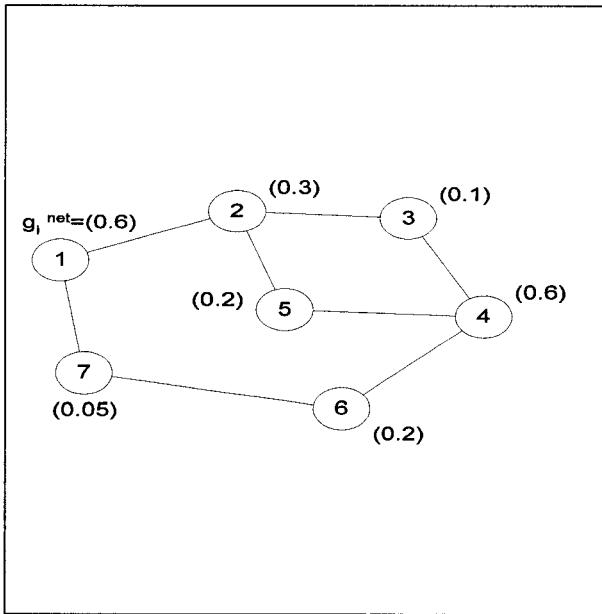


FIG. 4.3 Exemple de solution sous-optimale trouvée par l'algorithme de chemins disjoints. Les chemins possibles pour aller de 1 à 4 sont: 1-2-3-4, 1-2-5-4 et 1-7-6-4. Si l'algorithme des chemins disjoints trouve le chemin 1-2-3-4 en premier, il devra rejeter le chemin 1-2-5-4, pourtant plus avantageux

pas nécessairement le chemin optimal en termes de taux de service et de probabilité de délai.

On cherche les chemins possibles pour la paire origine-destination (1, 4). Il y a deux résultats possibles pour l'algorithme des chemins disjoints, dépendant de l'ordre dans lequel sont évalués les noeuds adjacents au noeud 2. Dans le cas où le noeud 3 figure devant le noeud 5 dans cette liste d'adjacence, l'algorithme de chemins disjoints trouve les chemins suivants:

1-2-3-4 (taux de service minimal = 0.1)

### 1-7-6-4 (taux de service minimal = 0.05)

Le chemin 1-2-5-4, qui offre le taux de service minimal de 0.2, ne sera pas considéré car le noeud 2 est déjà visité. Dans ce cas, la solution n'est pas optimale. Par contre, si le noeud 5 figure devant le noeud 3 dans la liste d'adjacence, le chemin trouvé pour cette paire origine-destination sera optimal.

Afin de trouver tous les chemins disjoints pour une paire origine-destination, une modification de l'algorithme d'énumération des chemins permet d'obtenir rapidement une solution même pour des réseaux de grande taille, qui comptent des milliers de noeuds et des centaines de milliers d'arêtes. L'algorithme est modifié de la façon suivante: au lieu d'explorer tous les adjacents d'un noeud, on explore seulement les adjacents qui n'ont pas encore été "marqués". Un noeud est marqué s'il fait partie d'un chemin qui est déjà dans l'ensemble solution. Cette modification permet d'énumérer seulement des chemins disjoints. Ceci a l'avantage de réduire, de façon importante, le temps de calcul de l'algorithme de routage.

Les figures 4.5 et 4.6 illustrent les chemins obtenus en utilisant l'algorithme ENUM\_D et ENUM pour le réseau de la figure 4.4.

Une autre solution possible pour générer plusieurs chemins pour une paire origine-destination est d'utiliser l'algorithme du  $k$ -ième plus court chemin. En utilisant cet algorithme, on trouve le plus court chemin, le deuxième plus court

---

**Algorithme 2 Algorithme ENUM\_D: *Chemins\_disjoints(courant, orig, dest)***

---

```

1: si tous les adjacents du noeud courant ont été explorés alors
2:   si noeud courant = noeud origine alors
3:     fin du programme
4: sinon
5:   chercher le noeud destination dans la liste d'adjacence
6:   si noeud destination trouvé alors
7:     sélectionner le noeud destination
8:   sinon
9:     sélectionner le prochain noeud adjacent pas encore exploré
10: répéter
11:   si le prochain noeud n'a pas encore été visité, il ne fait pas partie de
    l'ensemble solution et on n'est pas en train de revenir au noeud
    origine pour commencer une nouvelle recherche alors
12:     si prochain noeud = destination alors
13:       imprimer la solution courante {chemin origine-destination}
14:       marquer les noeuds qui font partie du chemin
15:       commencer le retour vers le noeud origine
16:     sinon si prochain noeud = origine alors
17:       cycle
18:     sinon
19:       inclure le prochain noeud dans la solution courante
20:       explorer les adjacents du prochain noeud:
          Chemins_disjoints(prochain, orig, dest)
21:       si noeud courant = origine alors
22:         enlever la marque "visité" sur tous les noeuds
23:         avancer au prochain noeud adjacent
24: jusqu'à tous les noeuds dans la liste d'adjacence ont été explorés

```

---

TAB. 4.2 *Liste d'adjacence pour le réseau de la figure 4.4*

| Noeud | Noeuds adjacents |
|-------|------------------|
| 1     | 2 6              |
| 2     | 1 3 6            |
| 3     | 2 4 5            |
| 4     | 2 3 5 7          |
| 5     | 3 4 6            |
| 6     | 1 2 5 7          |
| 7     | 4 6              |

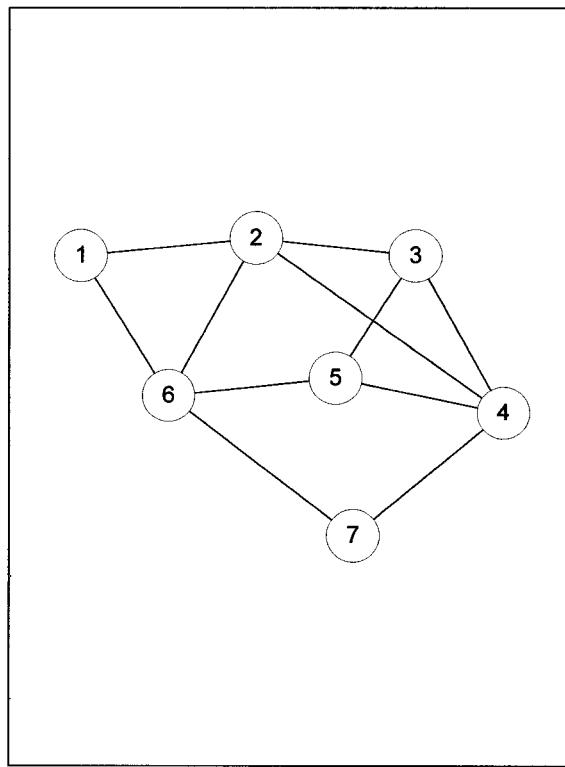


FIG. 4.4 Réseau pour la démonstration des résultats de ENUM et ENUM\_D

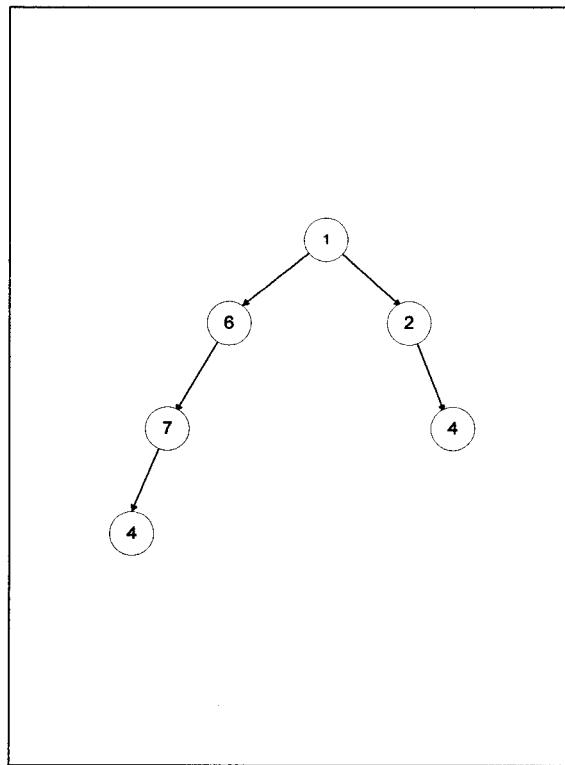


FIG. 4.5 *Chemins disjoints entre les noeuds 1 et 4 dans le réseau de la figure 4.4 - algorithme ENUM\_D*

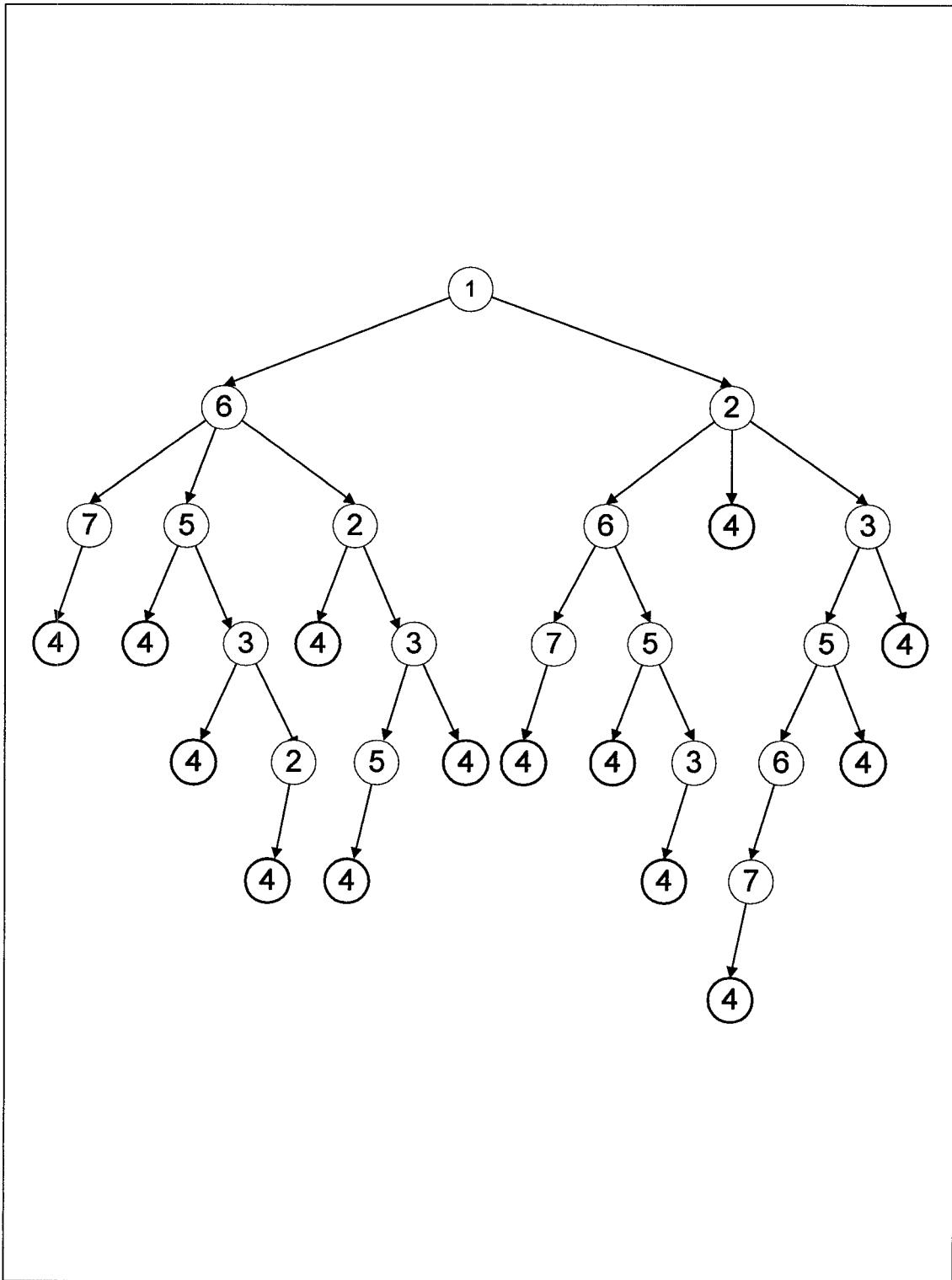


FIG. 4.6 Tous les chemins possibles entre les noeuds 1 et 4 dans le réseau de la figure 4.4 - algorithme ENUM

TAB. 4.3 *Tous les chemins possibles entre les noeuds 1 et 4 pour le réseau de la figure 4.4 - algorithme ENUM*

| Chemin        |
|---------------|
| 1 6 7 4       |
| 1 6 5 4       |
| 1 6 5 3 4     |
| 1 6 5 3 2 4   |
| 1 6 2 4       |
| 1 6 2 3 5 4   |
| 1 6 2 3 4     |
| 1 2 6 7 4     |
| 1 2 6 5 4     |
| 1 2 6 5 3 4   |
| 1 2 4         |
| 1 2 3 5 6 7 4 |
| 1 2 3 5 4     |
| 1 2 3 4       |

chemin, le troisième plus court chemin etc. Cependant, il est préférable d'utiliser l'algorithme des chemins disjoints, pour les raisons suivantes:

1. l'algorithme récursif est simple et compact
2. il n'est pas nécessaire, à priori, de *trier* les chemins trouvés, quelle que soit la métrique utilisée (coût des liens ou longueur du chemin). En effet, sous la politique de service GPS, le délai pour une session ne dépend pas du nombre de noeuds traversés par une session, mais seulement du noeud ayant le taux de service le moins élevé [46]
3. il est possible que les chemins trouvés par l'algorithme du  $k$ -ième plus court chemin contiennent tous le même noeud. Si ce noeud est celui qui offre le

taux de service le moins élevé, la solution ainsi trouvée peut être très loin de la solution optimale.

Dans ce chapitre, nous avons présenté un algorithme exhaustif d'énumération de chemins dans un graphe, ainsi qu'une modification de cet algorithme qui permet d'obtenir des chemins disjoints, ce qui diminue de beaucoup le nombre de chemins à considérer. Pour chacun des chemins ainsi obtenus, on peut calculer le délai ou la probabilité de délai, selon le modèle déterministe ou stochastique de la discipline de service GPS. Les résultats de ces calculs déterminent le choix du meilleur chemin pour une session. La suite de ce mémoire contient les résultats des simulations pour les différents algorithmes de routage présentés dans le chapitre précédent.

## CHAPITRE 5

### RÉSULTATS NUMÉRIQUES

#### 5.1 Topologies de réseaux

Afin de comparer la performance des algorithmes de routage décrits dans les chapitres précédents, nous avons étudié leur comportement dans différentes topologies de réseaux. Les réseaux étudiés diffèrent par leur taille (nombre de noeuds) et leur densité, qui correspond au nombre de liens présents dans le réseau par rapport au nombre de liens d'un réseau complètement connexe. Nous avons étudié des réseaux de 50, 100 et 150 noeuds, avec quatre densités différentes dans chaque cas, correspondant à environ 10%, 30%, 60% et 90% du nombre maximum de liens possible. Le tableau 5.1 présente une vue d'ensemble des topologies de réseaux étudiées. Ce tableau précise le nombre de noeuds et le nombre de liens pour chaque topologie étudiée. Au total, il y a 12 topologies différentes.

TAB. 5.1 *Topologies de réseau étudiées*

| Noeuds | Densité d'arcs |      |      |       |
|--------|----------------|------|------|-------|
|        | 10%            | 30%  | 60%  | 90%   |
| 50     | 100            | 350  | 700  | 1100  |
| 100    | 500            | 1500 | 3000 | 4500  |
| 150    | 1000           | 3000 | 6500 | 10000 |

Etant donné la nature de l'algorithme de recherche des chemins utilisé, la densité

du réseau a une influence importante sur le nombre de chemins générés et, par conséquent, sur le choix du meilleur chemin. Lorsqu'on augmente le nombre de liens dans un réseau, on augmente potentiellement le nombre de chemins (disjoints ou non) entre chaque paire de noeuds. Ceci a pour effet d'augmenter les chances de trouver un chemin qui respecte les contraintes de qualité de service. Lorsqu'on ajoute des liens à un réseau pour augmenter sa densité, les conditions suivantes s'appliquent:

- les capacités de traitement des noeuds restent les mêmes
- les liens existants dans un réseau moins dense sont préservés lorsque le nombre de liens augmente

## 5.2 Algorithmes de routage

Pour chacune des topologies mentionnées ci-haut, nous avons exécuté les algorithmes de routage présentés dans le tableau 5.2. Rappelons que ces algorithmes diffèrent par les éléments suivants:

- l'ensemble des chemins possibles pour une session
- la formule utilisée pour le contrôle d'admission
- la formule utilisée pour le routage

TAB. 5.2 *Algorithmes de routage*

| Algorithme          | Description   |
|---------------------|---|
| (1) PCC, dét        | plus court chemin, délai déterministe   |
| (2) dét             | énumération de chemins disjoints, délai déterministe  |
| (3) stoch           | énumération de chemins disjoints, délai stochastique  |
| (4) stoch+, c = 0.9 | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 90% et l'état moyen à 10% |
| (5) stoch+, c = 0.7 | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 70% et l'état moyen à 30% |
| (6) stoch+, c = 0.5 | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 50% et l'état moyen à 50% |
| (7) stoch+, c = 0.3 | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 30% et l'état moyen à 70% |
| (8) stoch+, c = 0.1 | énumération de chemins disjoints, délai stochastique, basé sur l'état courant à 10% et l'état moyen à 90% |

Dans les algorithmes (4) à (8), le coefficient de l'état moyen du réseau dans le calcul de la qualité de service varie de 0.1 à 0.9, par incrément de 0.2. Les valeurs numériques de l'état moyen sont obtenues en calculant la moyenne des taux d'occupation de chaque noeud lors d'une expérience de plus longue durée. L'expérience (3) a été utilisée pour obtenir un ensemble des valeurs moyennes du taux d'occupation, qui est utilisé comme entrée pour chacune des expériences (4) à (8). Il faut remarquer que n'importe laquelle des expériences aurait pu être utilisée pour générer un ensemble des valeurs moyennes. Par contre, il est important d'utiliser le même ensemble des valeurs moyennes pour les expériences (4) à (8) afin de pouvoir établir des comparaisons.

### 5.3 Trafic entrant dans le réseau

Les paramètres du trafic de la source sont les paramètres EBB dans le cas stochastique et les paramètres du "leaky bucket" dans le cas du modèle déterministe de GPS. Nous avons utilisé les processus d'arrivée discrets de type ON-OFF dont les caractérisations EBB ont été citées dans [46]. Le tableau 5.3 présente ces processus stochastiques et leurs modèles EBB. La probabilité de transition de l'état OFF à l'état ON est  $p_i$  et la probabilité de la transition inverse est  $q_i$ . Le taux de transmission dans l'état ON est noté  $\lambda_i$ . Le taux de transmission moyen  $\bar{\lambda}_i$  est égal à:

$$\frac{p_i \lambda_i}{p_i + q_i}$$

TAB. 5.3 *Sources ON-OFF et leurs caractérisations EBB*

| session | $p_i$ | $q_i$ | $\lambda_i$ | $\bar{\lambda}_i$ | $\rho_i$ | $\Lambda_i$ | $\alpha_i$ |
|---------|-------|-------|-------------|-------------------|----------|-------------|------------|
| 1       | 0.3   | 0.7   | 0.5         | 0.15              | 0.2      | 1.0         | 1.74       |
| 2       | 0.4   | 0.4   | 0.4         | 0.2               | 0.25     | 0.92        | 1.76       |
| 3       | 0.3   | 0.3   | 0.3         | 0.15              | 0.2      | 0.84        | 2.13       |
| 4       | 0.4   | 0.6   | 0.5         | 0.2               | 0.25     | 1.0         | 1.62       |

La demande de qualité de service est exprimée en termes de probabilité de délai maximale dans le cas stochastique, ou de délai maximal dans le cas déterministe. Nous avons utilisé une probabilité de délai de 0.3 et un délai maximal acceptable de 4 s. Les valeurs numériques de la demande utilisée pour un réseau de 50 noeuds sont présentées dans le tableau 5.4 en annexe. Les paramètres EBB sont tirés d'exemples de caractérisations de processus stochastiques de type ON-OFF

[46]. Les paramètres du "leaky bucket" sont choisis de façon à ne pas affecter le processus d'arrivée. Par exemple, pour les processus ON-OFF montrés ci-haut, le taux d'arrivée des jetons dans le "bucket" est égal au taux de transmission dans l'état "ON", ce qui assure que le trafic ne sera jamais en attente à cause d'un manque de jetons. On utilise les mêmes valeurs de la demande pour tous les réseaux de 50 noeuds, quelle que soit leur densité. Cependant, la demande est différente pour les réseaux de 50, 100 et 150 noeuds, pour répartir le trafic de façon à atteindre un plus grand nombre de noeuds. Par exemple, dans le cas d'un réseau de 50 noeuds, le tableau contient des sessions ayant pour origine un noeud entre 1 et 50 et un noeud destination entre 1 et 50. Dans le cas d'un réseau de 100 noeuds, le même tableau est repris mais il est augmenté avec les sessions ayant pour origine et destination un noeud entre 51 et 100.

#### **5.4 Critère de comparaison des algorithmes**

Pour chaque topologie de réseau et pour chaque type d'algorithme de routage, nous avons mesuré le taux de blocage de sessions, qui est défini comme suit:

$$\text{taux de blocage} = \frac{\text{nombre de sessions rejetées}}{\text{nombre total de sessions arrivées dans le réseau}}$$

Le calcul du taux de blocage est effectué après chaque arrivée de session. La nouvelle session est acceptée ou rejetée. Si elle est rejetée, le total des sessions rejetées est mis à jour. Le taux de blocage est calculé en divisant le nombre de sessions rejetées par le nombre total d'arrivées. Le tableau 5.5 montre un exemple

TAB. 5.4 Exemple de la demande pour un réseau de 50 noeuds

| Orig | Dest | Paramètres de la source                |           |              | Durée(s) |     |
|------|------|--|-----------|--------------|----------|-----|
|      |      | Exponentially Bounded Burstiness (EBB) |           | Leaky bucket |          |     |
|      |      | $\rho$                                 | $\Lambda$ | $\alpha$     |          |     |
| 1    | 7    | 0.2                                    | 1         | 1.74         | 0.5      | 100 |
| 4    | 11   | 0.2                                    | 0.84      | 2.13         | 0.3      | 100 |
| 6    | 11   | 0.25                                   | 1         | 1.62         | 0.5      | 100 |
| 2    | 7    | 0.2                                    | 1         | 1.74         | 0.5      | 100 |
| 5    | 11   | 0.25                                   | 0.92      | 1.62         | 0.4      | 100 |
| 3    | 10   | 0.2                                    | 1         | 1.74         | 0.5      | 100 |
| 2    | 13   | 0.25                                   | 0.92      | 1.76         | 0.4      | 100 |
| 10   | 2    | 0.2                                    | 0.84      | 2.13         | 0.3      | 100 |
| 10   | 4    | 0.25                                   | 1         | 1.62         | 0.5      | 100 |
| 5    | 4    | 0.2                                    | 1         | 1.74         | 0.5      | 200 |
| 5    | 11   | 0.25                                   | 0.92      | 1.76         | 0.4      | 200 |
| 1    | 12   | 0.2                                    | 0.84      | 2.13         | 0.3      | 200 |
| 10   | 15   | 0.25                                   | 1         | 1.62         | 0.5      | 200 |
| 15   | 20   | 0.25                                   | 0.92      | 1.62         | 0.4      | 200 |
| 15   | 21   | 0.2                                    | 1         | 1.74         | 0.5      | 200 |
| 25   | 12   | 0.2                                    | 0.84      | 2.13         | 0.3      | 200 |
| 30   | 44   | 0.2                                    | 1         | 1.74         | 0.5      | 200 |
| 35   | 49   | 0.25                                   | 0.92      | 1.76         | 0.4      | 300 |
| 40   | 11   | 0.2                                    | 1         | 1.74         | 0.5      | 300 |
| 3    | 7    | 0.25                                   | 0.92      | 1.76         | 0.4      | 300 |
| 3    | 8    | 0.25                                   | 1         | 1.62         | 0.5      | 300 |
| 4    | 13   | 0.2                                    | 1         | 1.74         | 0.5      | 300 |
| 6    | 13   | 0.25                                   | 0.92      | 1.62         | 0.4      | 400 |
| 4    | 7    | 0.2                                    | 1         | 1.74         | 0.5      | 400 |
| 5    | 13   | 0.25                                   | 0.92      | 1.76         | 0.4      | 400 |
| 3    | 12   | 0.2                                    | 0.84      | 2.13         | 0.3      | 400 |
| 4    | 13   | 0.25                                   | 1         | 1.62         | 0.5      | 400 |
| 12   | 4    | 0.2                                    | 1         | 1.74         | 0.5      | 400 |
| 5    | 4    | 0.25                                   | 0.92      | 1.76         | 0.4      | 400 |
| 5    | 13   | 0.25                                   | 1         | 1.62         | 0.5      | 400 |
| 3    | 14   | 0.2                                    | 1         | 1.74         | 0.5      | 400 |
| 12   | 17   | 0.25                                   | 0.92      | 1.62         | 0.4      | 500 |

de calcul du taux de blocage. La simulation s'arrête lorsque le nombre d'arrivées spécifié au début de la simulation a été atteint.

TAB. 5.5 *Exemple de calcul du taux de blocage*

| Sessions arrivées | Sessions rejetées | Taux de blocage           |
|-------------------|-------------------|---------------------------|
| 99                | 0                 | 0                         |
| 100               | 1                 | $1/100 = 0.01 (1\%)$      |
| 101               | 1                 | $1/101 = 0.0099 (0.99\%)$ |
| 102               | 2                 | $2/102 = 0.0196 (1.96\%)$ |
| 103               | 3                 | $3/103 = 0.029 (2.91\%)$  |
| 104               | 3                 | $3/104 = 0.028 (2.88\%)$  |
| 105               | 3                 | $3/105 = 0.028 (2.88\%)$  |
| 106               | 3                 | $3/106 = 0.028 (2.88\%)$  |

## 5.5 Validation du simulateur

Le programme de simulation qui a été développé pour les besoins de ce mémoire a fait l'objet d'une validation sur des réseaux de test, de taille suffisamment réduite pour pouvoir suivre les calculs du simulateur (une dizaine de noeuds). Un exemple de réseau de test est le réseau de la figure 4.4. Pendant la validation, les résultats du simulateur sont examinés à chaque étape de calcul. Par exemple, pour l'ajout d'une session, les étapes suivantes sont vérifiées:

- calcul du taux de service dans chaque noeud d'un chemin
- choix du noeud ayant le taux de service minimum
- calcul de la probabilité de délai

Ces étapes sont effectuées pour tous les chemins candidats pour une nouvelle session. Le meilleur chemin est ensuite choisi et la session est placée sur ce chemin, ce qui signifie que les taux d'occupation des noeuds sont mis à jour. De la même manière, lorsqu'une session se termine, elle est enlevée du réseau et l'état du réseau est mis à jour. Toutes ces étapes sont vérifiées à l'aide d'un rapport détaillé de chaque étape de calcul et de prise de décision, généré par le simulateur. Cette vérification a été effectuée sur des réseaux de petite taille (moins de 20 noeuds). Pour assurer l'intégrité du simulateur pour des réseaux plus grands, on a ajouté au simulateur de multiples points de vérification. Ces vérifications ont pour but d'assurer que les calculs effectués sont corrects, les conditions de validité des formules sont respectées et que les résultats sont plausibles. On vérifie entre autres les conditions de stabilité des équations 2.4 et 2.8. Ailleurs, on vérifie par exemple que lorsqu'on enlève une session du réseau, la probabilité de délai des autres sessions diminue ou reste inchangée. Si elle augmentait, le simulateur signalerait une erreur.

## 5.6 Variabilité des résultats

Pendant la simulation, chaque nouvelle session ajoutée au réseau est générée en allant chercher, de façon aléatoire, un élément dans un tableau de types de sessions. L'index pour accéder à un élément de ce tableau est obtenu par un générateur de nombres pseudo-aléatoires. Si on ne change pas le nombre initial (ang. *seed*) du générateur, on obtient la même suite de nombres, ce qui permet de tester le

réseau qui opère avec différents algorithmes de routage mais soumis à la même demande. Lorsque l'on change le nombre initial du générateur, on obtient une séquence différente d'arrivées de sessions. Le graphique 5.1 présente un aperçu des résultats obtenus pour 7 valeurs initiales différentes du générateur de nombres pseudo-aléatoires. Les taux de blocage affichées sur le graphique sont ceux obtenus après l'arrivée de 2500 sessions. La topologie du réseau testé est de 150 noeuds et 6500 arcs, avec la version 3 de l'algorithme de routage "stochastique+".

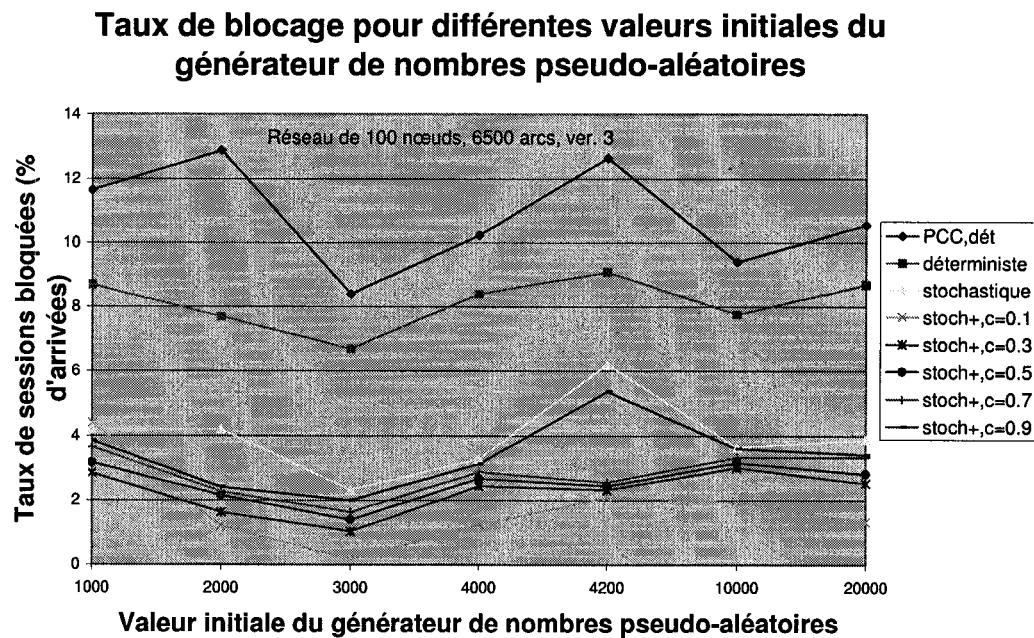


FIG. 5.1 Variabilité des résultats pour un réseau de 150 noeuds, 6500 arcs

Sur un échantillon de 20 expériences, la moyenne du taux de blocage pour l'algorithme "déterministe" est de 8.11% avec un intervalle de confiance à 95% de 0.34. On peut alors être sûr à 95% que la moyenne du taux de blocage pour ce type d'algorithme se trouve dans l'intervalle  $8.11\% \pm 0.34\%$ , c'est-à-dire entre 7.77% et 8.45%. Sur le même échantillon d'expériences, pour l'algorithme "stochastique+,c=0.7", le taux de blocage moyen est de 2.24% avec un intervalle de confiance de 0.29, ce qui place la moyenne du taux de blocage pour cet algorithme entre 1.95% et 2.53%.

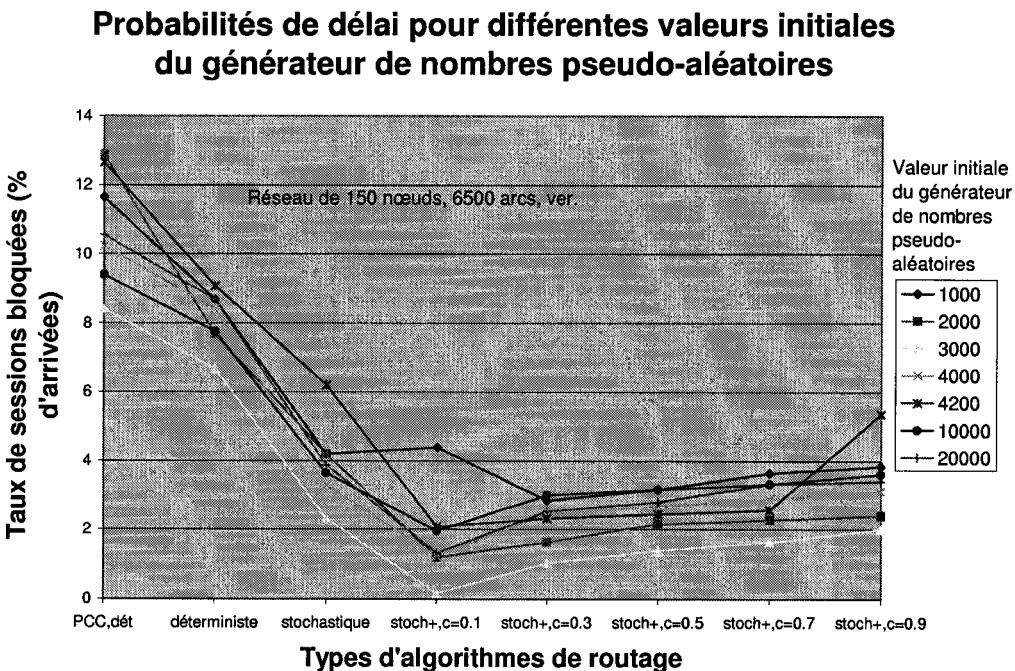


FIG. 5.2 Variabilité des résultats pour un réseau de 150 noeuds, 6500 arcs

Le graphique 5.2 montre les résultats de l'expérience pour tous les algorithmes de routage testés. La plupart du temps, on obtient le taux de blocage minimum pour l'algorithme "stochastique+, c=0.1", peu importe la valeur initiale donnée au générateur de nombres aléatoires.

### 5.7 Discussion des résultats

Les résultats des simulations effectuées sur les différents réseaux montrent que les performances des différents algorithmes de routage varient beaucoup, permettant dans certains cas de réduire de moitié le taux de blocage des sessions ou même d'éliminer complètement le blocage (voir les résultats pour les algorithmes "stoch+",  $c=0.1$ " et "stoch+,  $c=0.3$ " pour le réseau de 100 noeuds 5.10). L'analyse des résultats permet de classer les algorithmes en fonction de la performance (tableau 5.6), dans l'ordre qui apparaît le plus fréquemment lors des simulations.

TAB. 5.6 *Performance des algorithmes de routage*

| Rang | Description       |
|------|-------------------|
| 1    | stoch+, $c = 0.1$ |
| 2    | stoch+, $c = 0.3$ |
| 3    | stoch+, $c = 0.5$ |
| 4    | stoch+, $c = 0.7$ |
| 5    | stoch+, $c = 0.9$ |
| 6    | stoch             |
| 7    | dét               |
| 8    | PCC, dét          |

On retrouve cet ordre, par exemple, dans les graphiques pour le réseau de 50

noeuds et 700 arcs, version 2 (fig. 5.3) et dans celui pour le réseau de 50 noeuds et 1100 arcs, version 3 (fig. 5.4).

### Taux de blocage de sessions, ver. 2

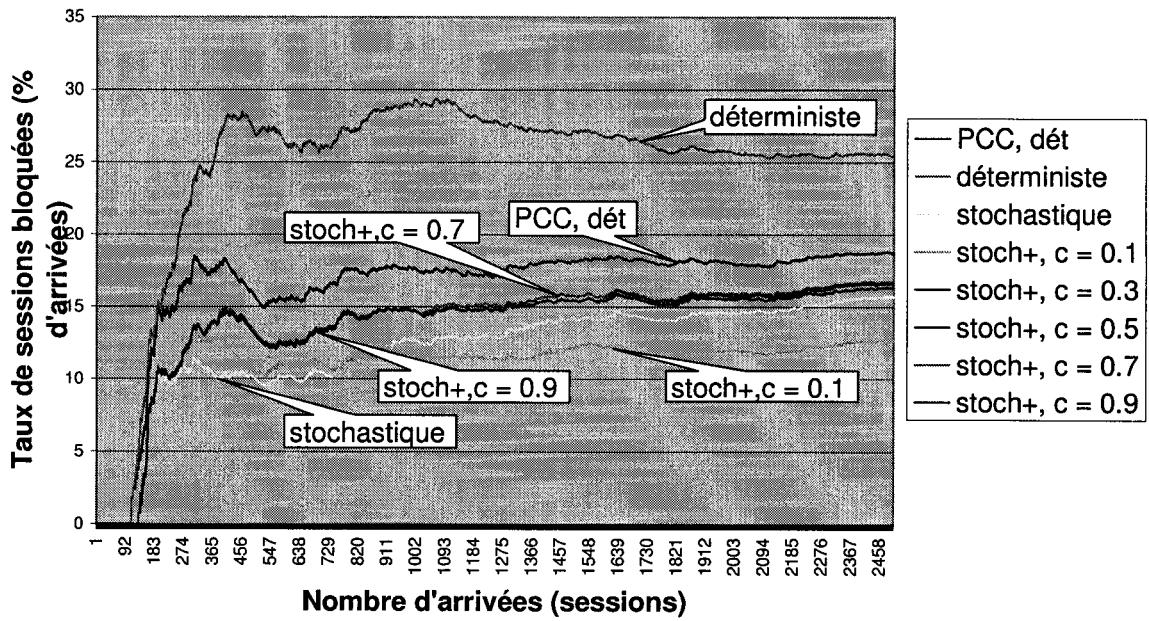


FIG. 5.3 *Taux de blocage pour un réseau de 50 noeuds, 700 arcs*

Pour les réseaux de 100 noeuds, on retrouve ce classement d'algorithmes, entre autres, dans les graphes suivants: 100 noeuds et 500 arcs (fig. 5.5), 100 noeuds et 3000 arcs (fig. 5.6), 100 noeuds et 4500 arcs (fig. 5.7).

### Taux de blocage de sessions, ver. 3

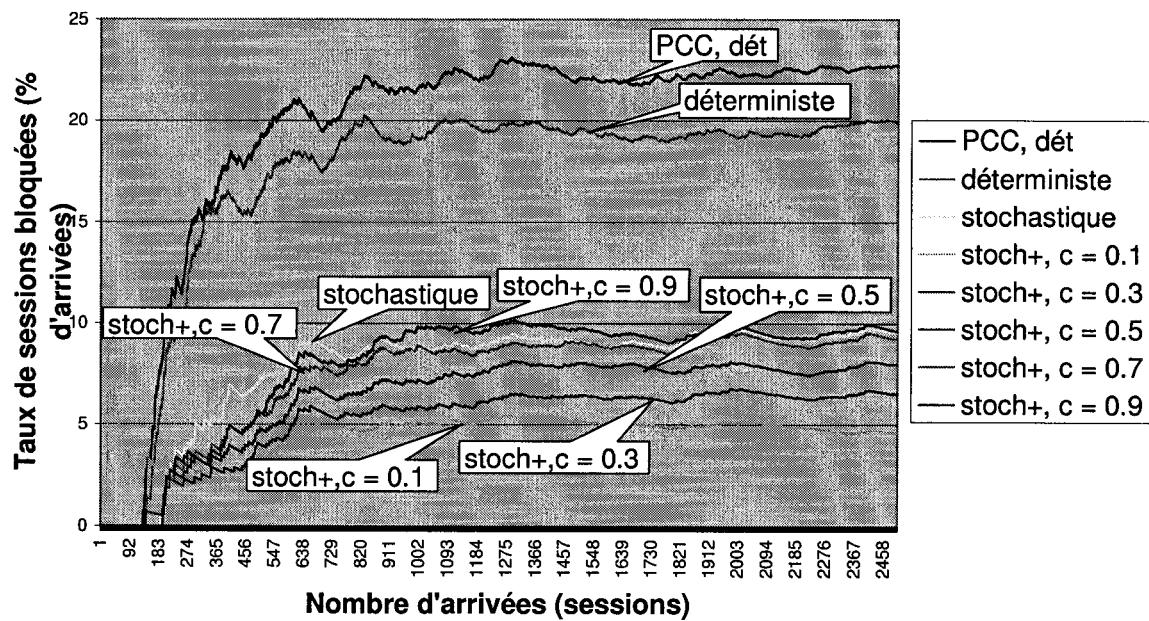


FIG. 5.4 Taux de blocage pour un réseau de 50 noeuds, 1100 arcs

### Taux de blocage de sessions, ver. 2

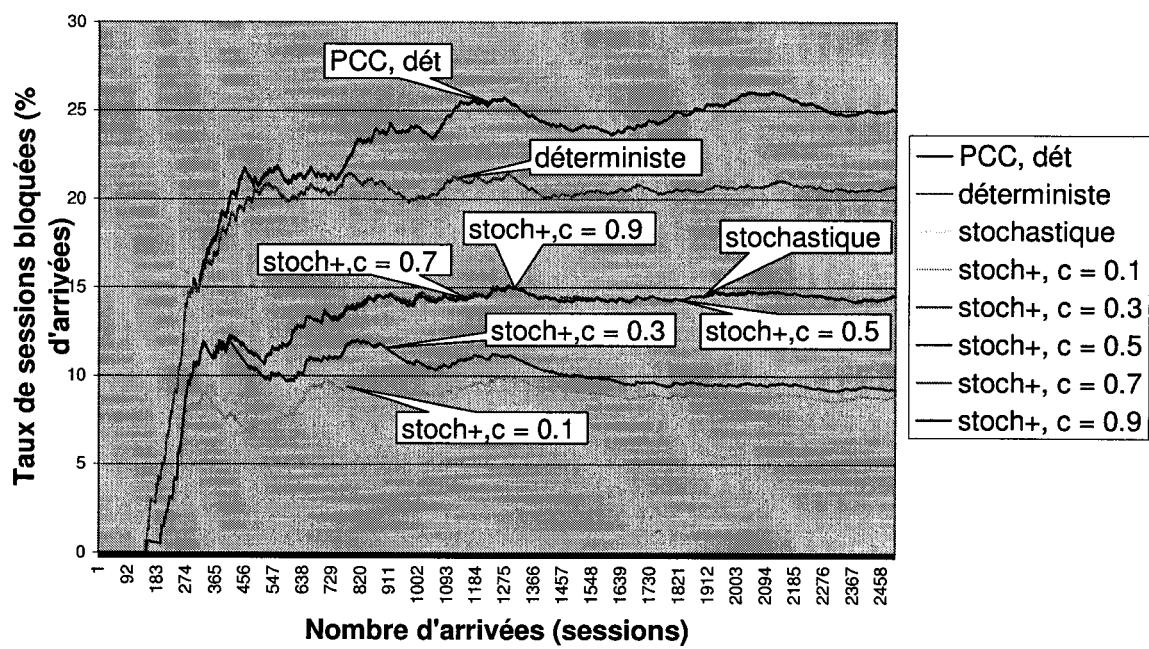


FIG. 5.5 Taux de blocage pour un réseau de 100 noeuds, 500 arcs

### Taux de blocage de sessions, ver. 3

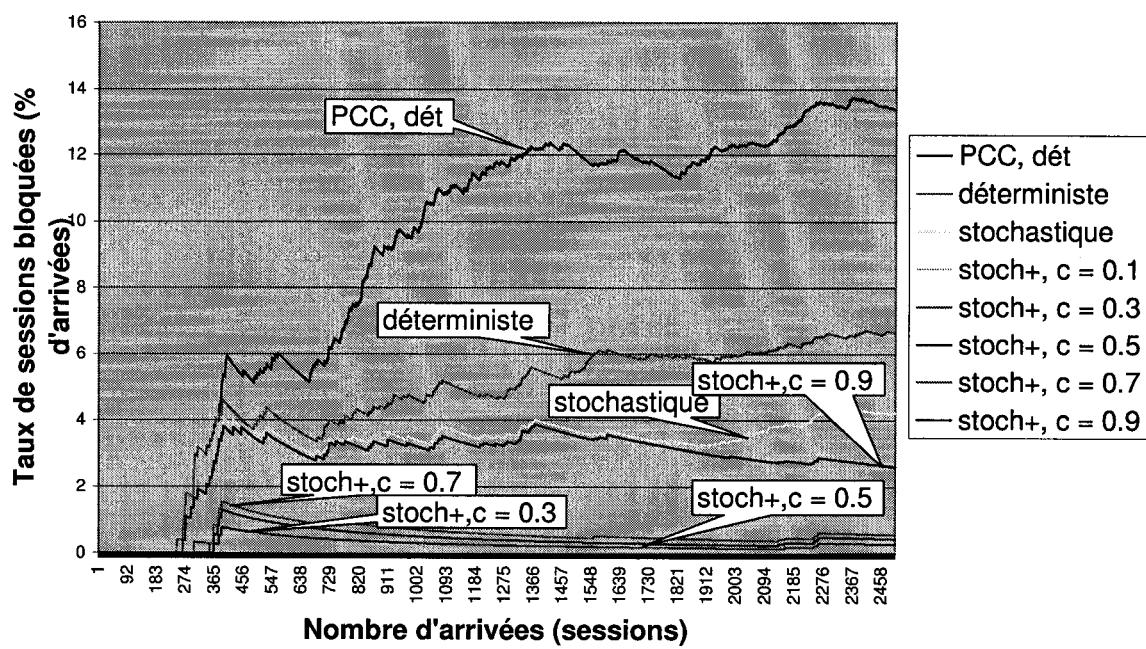


FIG. 5.6 Taux de blocage pour un réseau de 100 noeuds, 3000 arcs

### Taux de blocage de sessions, ver. 1

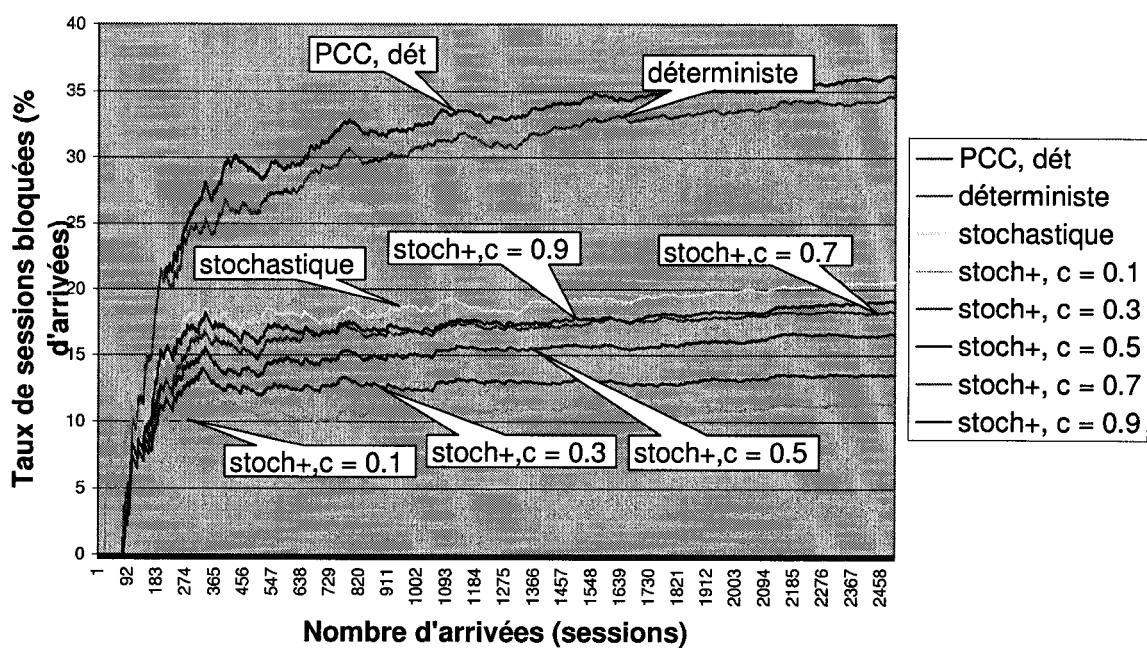


FIG. 5.7 Taux de blocage pour un réseau de 100 noeuds, 4500 arcs

Enfin, pour les réseaux de 150 noeuds, ce classement apparaît dans les graphes suivants: 150 noeuds et 6500 arcs (fig. 5.8), 150 noeuds et 10000 arcs (fig. 5.9).

Les tableaux 5.7, 5.8 et 5.9 présentent un sommaire des taux de blocage pour les réseaux de 50 noeuds, avec différentes densités de liens. Les taux de blocage correspondent aux valeurs atteintes après ajout de 2500 sessions dans le réseau. Les algorithmes de routage sont ceux décrits dans le chapitre 3. Les résultats des algorithmes les plus performants sont affichés en caractères gras.

TAB. 5.7 *Taux de blocage en % après 2500 sessions, réseau de 50 noeuds, VER. 1*

| Densité | Arcs | Types d'algorithmes de routage |                         |       |              |                          |       |       |       |       |
|---------|------|--------------------------------|-------------------------|-------|--------------|--------------------------|-------|-------|-------|-------|
|         |      | PCC                            | Énumération des chemins |       |              |                          |       |       |       |       |
|         |      |                                | dét                     | dét   | stoch        | stochastique+, VERSION 1 |       |       |       |       |
|         |      |                                |                         |       |              | c=0.1                    | c=0.3 | c=0.5 | c=0.7 | c=0.9 |
| 10%     | 100  | 10.68                          | 27.28                   | 23.32 | <b>2.88</b>  | 7.84                     | 14.20 | 17.12 | 18.64 |       |
| 30%     | 350  | 13.20                          | 17.21                   | 4.40  | <b>0.00</b>  | 1.52                     | 2.72  | 3.32  | 4.16  |       |
| 60%     | 700  | 18.68                          | 22.04                   | 14.28 | <b>10.00</b> | 11.40                    | 12.60 | 13.12 | 13.76 |       |
| 90%     | 1100 | 22.44                          | 21.00                   | 15.72 | <b>9.36</b>  | 11.72                    | 13.44 | 15.16 | 15.32 |       |

TAB. 5.8 *Taux de blocage en % après 2500 sessions, réseau de 50 noeuds, VER. 2*

| Densité | Arcs | Types d'algorithmes de routage |                         |       |              |                          |       |       |       |       |
|---------|------|--------------------------------|-------------------------|-------|--------------|--------------------------|-------|-------|-------|-------|
|         |      | PCC                            | Énumération des chemins |       |              |                          |       |       |       |       |
|         |      |                                | dét                     | dét   | stoch        | stochastique+, VERSION 2 |       |       |       |       |
|         |      |                                |                         |       |              | c=0.1                    | c=0.3 | c=0.5 | c=0.7 | c=0.9 |
| 10%     | 100  | 15.32                          | 36.64                   | 24.24 | <b>15.20</b> | 23.44                    | 23.52 | 23.00 | 23.64 |       |
| 30%     | 350  | 13.28                          | 18.80                   | 5.84  | <b>3.20</b>  | 4.92                     | 4.92  | 5.00  | 4.96  |       |
| 60%     | 700  | 18.72                          | 25.40                   | 15.68 | <b>12.64</b> | 16.68                    | 16.56 | 16.68 | 16.28 |       |
| 90%     | 1100 | 28.96                          | 23.92                   | 19.04 | <b>18.92</b> | 19.84                    | 19.68 | 19.72 | 19.76 |       |

Les résultats pour les réseaux de 100 noeuds sont présentés dans les tableaux

### Taux de blocage de sessions, ver. 2

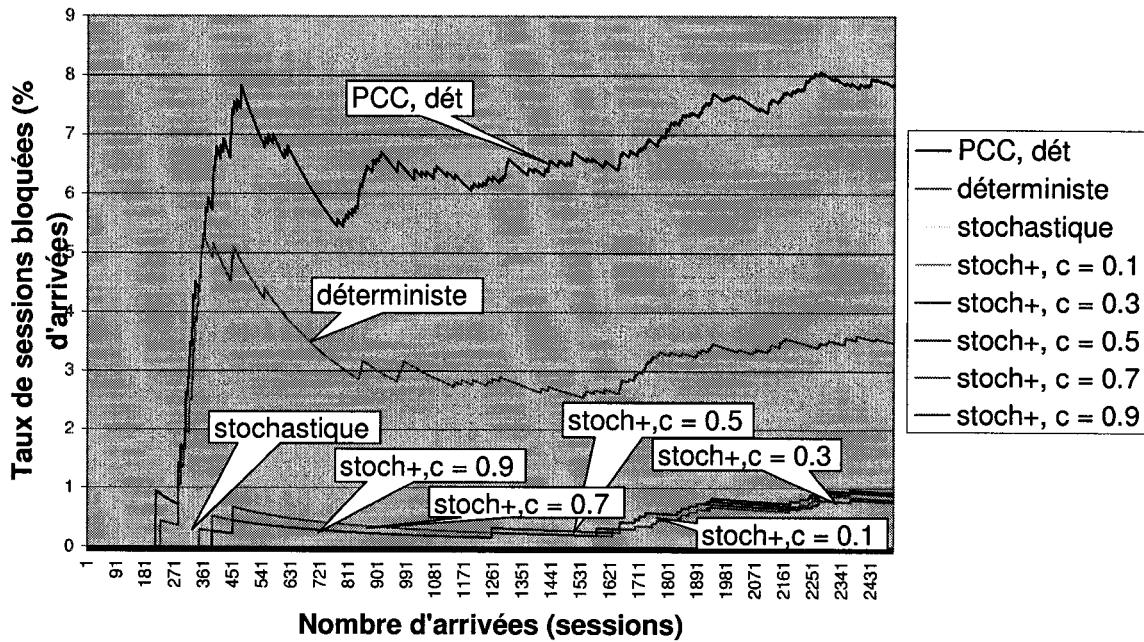


FIG. 5.8 Taux de blocage pour un réseau de 150 noeuds, 6500 arcs

TAB. 5.9 Taux de blocage en % après 2500 sessions, réseau de 50 noeuds, VER. 3

| Densité | Arcs | Types d'algorithmes de routage |                         |       |       |                          |       |       |       |       |  |
|---------|------|--------------------------------|-------------------------|-------|-------|--------------------------|-------|-------|-------|-------|--|
|         |      | PCC                            | Énumération des chemins |       |       |                          |       |       |       |       |  |
|         |      |                                | dét                     | dét   | stoch | stochastique+, VERSION 3 |       |       |       |       |  |
|         |      |                                |                         |       |       | c=0.1                    | c=0.3 | c=0.5 | c=0.7 | c=0.9 |  |
| 10%     | 100  | 16.44                          | 35.76                   | 24.16 |       | <b>6.44</b>              | 9.20  | 19.24 | 22.48 | 23.28 |  |
| 30%     | 350  | 13.76                          | 18.72                   | 2.88  |       | <b>0.00</b>              | 0.44  | 0.72  | 1.24  | 2.08  |  |
| 60%     | 700  | 18.84                          | 21.80                   | 13.20 |       | <b>9.72</b>              | 11.00 | 11.48 | 12.28 | 12.64 |  |
| 90%     | 1100 | 22.72                          | 19.88                   | 9.56  |       | <b>4.60</b>              | 6.52  | 7.96  | 9.24  | 9.60  |  |

### Taux de blocage de sessions, ver. 1

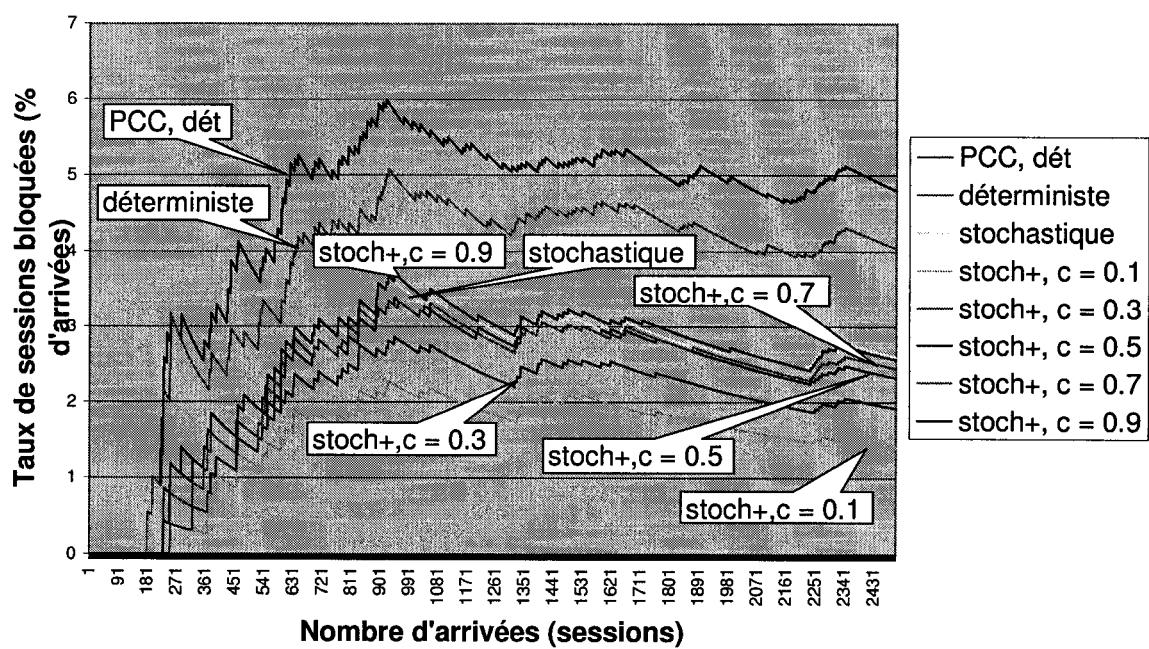


FIG. 5.9 Taux de blocage pour un réseau de 150 noeuds, 10000 arcs

5.10, 5.11 et 5.12, et pour les réseaux de 150 noeuds dans les tableaux 5.13, 5.14 et 5.15.

TAB. 5.10 *Taux de blocage en % après 2500 sessions, réseau de 100 noeuds, VER. 1*

| Densité | Arcs | Types d'algorithmes de routage |                         |       |              |                          |       |       |       |       |
|---------|------|--------------------------------|-------------------------|-------|--------------|--------------------------|-------|-------|-------|-------|
|         |      | PCC                            | Énumération des chemins |       |              |                          |       |       |       |       |
|         |      |                                | dét                     | dét   | stoch        | stochastique+, VERSION 1 |       |       |       |       |
|         |      |                                |                         |       |              | c=0.1                    | c=0.3 | c=0.5 | c=0.7 | c=0.9 |
| 10%     | 500  | 22.24                          | 22.87                   | 14.52 | 10.28        | <b>10.24</b>             | 13.64 | 13.88 | 14.32 |       |
| 30%     | 1500 | 6.72                           | 6.60                    | 1.16  | <b>0.04</b>  | 0.12                     | 0.64  | 0.80  | 1.24  |       |
| 60%     | 3000 | 7.72                           | 5.08                    | 0.72  | <b>0.00</b>  | <b>0.00</b>              | 0.12  | 0.36  | 0.68  |       |
| 90%     | 4500 | 36.04                          | 34.52                   | 20.55 | <b>11.30</b> | 13.52                    | 16.52 | 18.24 | 19.04 |       |

TAB. 5.11 *Taux de blocage en % après 2500 sessions, réseau de 100 noeuds, VER. 2*

| Densité | Arcs | Types d'algorithmes de routage |                         |       |             |                          |             |             |       |       |
|---------|------|--------------------------------|-------------------------|-------|-------------|--------------------------|-------------|-------------|-------|-------|
|         |      | PCC                            | Énumération des chemins |       |             |                          |             |             |       |       |
|         |      |                                | dét                     | dét   | stoch       | stochastique+, VERSION 2 |             |             |       |       |
|         |      |                                |                         |       |             | c=0.1                    | c=0.3       | c=0.5       | c=0.7 | c=0.9 |
| 10%     | 500  | 25.04                          | 20.71                   | 14.60 | <b>8.76</b> | 9.28                     | 14.56       | 14.60       | 14.56 |       |
| 30%     | 1500 | 6.56                           | 6.48                    | 6.08  | <b>2.20</b> | 6.08                     | 6.08        | 6.08        | 6.08  |       |
| 60%     | 3000 | 7.68                           | 5.24                    | 0.80  | 4.24        | 1.96                     | <b>0.68</b> | <b>0.68</b> | 0.72  |       |
| 90%     | 4500 | 40.75                          | 36.56                   | 32.36 | 21.44       | <b>21.04</b>             | 21.36       | 25.04       | 32.32 |       |

Pour tous les réseaux testés, les algorithmes d'énumération des chemins basés sur l'état moyen du réseau offrent de bonnes performances. Pour les réseaux de densité élevée (60% et plus du nombre de liens d'un réseau complètement connexe), les algorithmes d'énumération des chemins offrent de très bonnes performances par rapport à l'algorithme des plus courts chemins. Parmi les algorithmes d'énumération des chemins, les meilleurs résultats sont obtenus avec les algorithmes qui tiennent compte de l'état moyen du réseau dans une proportion de plus de 50%

TAB. 5.12 *Taux de blocage en % après 2500 sessions, réseau de 100 noeuds, VER. 3*

| Densité | Arcs | Types d'algorithmes de routage |       |                         |                          |             |       |       |       |
|---------|------|--------------------------------|-------|-------------------------|--------------------------|-------------|-------|-------|-------|
|         |      | PCC                            |       | Énumération des chemins |                          |             |       |       |       |
|         |      | dét                            | dét   | stoch                   | stochastique+, VERSION 3 |             |       |       |       |
|         |      |                                |       |                         | c=0.1                    | c=0.3       | c=0.5 | c=0.7 | c=0.9 |
| 10%     | 500  | 21.87                          | 22.44 | 14.56                   | 12.76                    | <b>8.64</b> | 13.84 | 13.92 | 14.48 |
| 30%     | 1500 | 6.60                           | 6.56  | 1.60                    | 2.04                     | <b>1.04</b> | 1.00  | 1.68  | 1.44  |
| 60%     | 3000 | 13.36                          | 6.64  | 4.20                    | <b>0.04</b>              | 0.28        | 0.44  | 0.56  | 2.60  |
| 90%     | 4500 | 36.56                          | 34.27 | 21.36                   | <b>12.72</b>             | 15.16       | 17.08 | 18.40 | 19.28 |

(voir les tableaux de résultats pour les réseaux de 100 noeuds et de 150 noeuds, par exemple le tableau 5.15 pour les algorithmes "stoch+", c=0.1", "stoch+", c=0.3" et "stoch+, c=0.5"). Ce résultat peut être attribué au fait que dans les réseaux de densité élevée, on obtient un plus grand nombre de chemins disjoints, ce qui augmente les chances de trouver un chemin ayant de bons paramètres de qualité de service.

Pour des réseaux peu denses (de 10% à 30% du nombre maximum de liens), l'algorithme "PCC, déterministe" offre de bonnes performances, quelquefois meilleures que celles des algorithmes basés sur l'énumération des chemins disjoints (voir le réseau de 100 noeuds et 500 arcs dans les tableaux 5.10 et 5.12 et le réseau de 150 noeuds et 1000 arcs dans les tableaux 5.13, 5.15 et 5.15). Ce résultat peut sembler contre-intuitif. En effet, lorsqu'on a le choix entre plusieurs chemins, les performances devraient être meilleures que lorsqu'on dispose d'un seul chemin (le plus court). Cependant, l'énumération de chemins disjoints ne garantit pas le choix des chemins les plus favorables du point de vue de la qualité de service. L'objectif de

cet algorithme est de générer *rapidement* un ensemble de chemins possibles, même pour des réseaux de grande taille. Les chemins trouvés sont ensuite évalués pour choisir le meilleur en termes de qualité de service. Il peut arriver que les chemins choisis soient moins performants que le plus court chemin. Cependant, le fait de disposer de plusieurs chemins disjoints devrait augmenter *en général* les chances de trouver un chemin qui respecte les contraintes de qualité de service, ce qui se traduit alors par un taux de blocage plus faible. Cet avantage est plus marqué pour les réseaux plus denses (voir les tableaux de résultats pour les réseaux avec des densités de 60% et 90%). Dans les réseaux épars, la faible densité de liens ne permet pas d'avoir assez de chemins disjoints pour générer une solution intéressante, ce qui neutralise l'avantage des algorithmes d'énumération des chemins.

Par exemple, la figure 5.10 présente un cas où l'algorithme "PCC, dét" offre de meilleures performances que plusieurs algorithmes basés sur l'énumération des chemins disjoints. Ce résultat est causé par la faible densité du réseau (seulement 100 arcs pour 50 noeuds).

Pour un réseau de 50 noeuds et 1100 arcs (fig. 5.11) les algorithmes de type "stoch+" offrent des performances bien supérieures à celle de l'algorithme "PCC, dét", permettant de réduire de moitié le taux de blocage des sessions.

L'analyse de tous les résultats permet de conclure que la connaissance de l'état moyen du réseau est bénéfique pour l'efficacité du réseau. Si on connaît la charge moyenne sur les noeuds, on peut acheminer les sessions de façon à éviter certains

### Taux de blocage de sessions, ver. 1

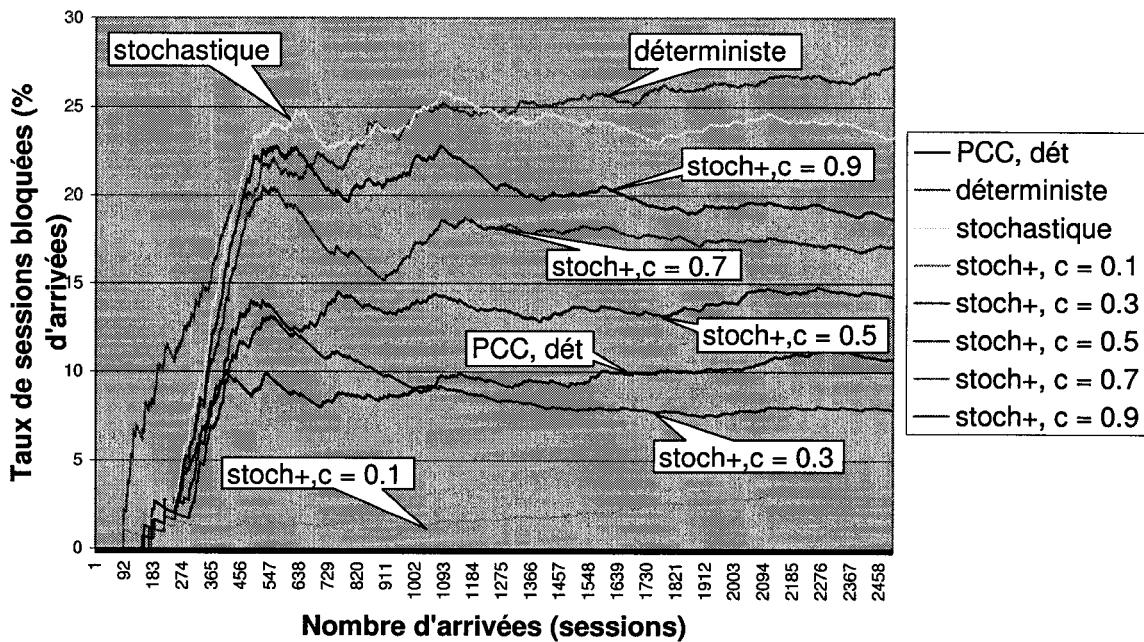


FIG. 5.10 Taux de blocage pour un réseau de 50 noeuds, 100 arcs

TAB. 5.13 Taux de blocage en % après 2500 sessions, réseau de 150 noeuds, VER. 1

| Densité | Arcs  | Types d'algorithmes de routage |                         |       |       |                          |       |      |      |  |
|---------|-------|--------------------------------|-------------------------|-------|-------|--------------------------|-------|------|------|--|
|         |       | PCC                            | Énumération des chemins |       |       |                          |       |      |      |  |
|         |       |                                | dét                     | dét   | stoch | stochastique+, VERSION 1 |       |      |      |  |
|         |       |                                | c=0.1                   | c=0.3 | c=0.5 | c=0.7                    | c=0.9 |      |      |  |
| 10%     | 1000  | 1.32                           | 6.04                    | 0.60  | 0.00  | 0.00                     | 0.00  | 0.36 | 0.52 |  |
| 30%     | 3000  | 23.20                          | 15.16                   | 5.56  | 4.60  | 4.80                     | 5.00  | 5.04 | 5.16 |  |
| 60%     | 6500  | 7.52                           | 3.52                    | 1.00  | 0.00  | 0.00                     | 0.28  | 0.56 | 0.80 |  |
| 90%     | 10000 | 4.80                           | 4.04                    | 2.60  | 1.40  | 1.92                     | 2.32  | 2.44 | 2.56 |  |

### Taux de blocage de sessions, ver. 1

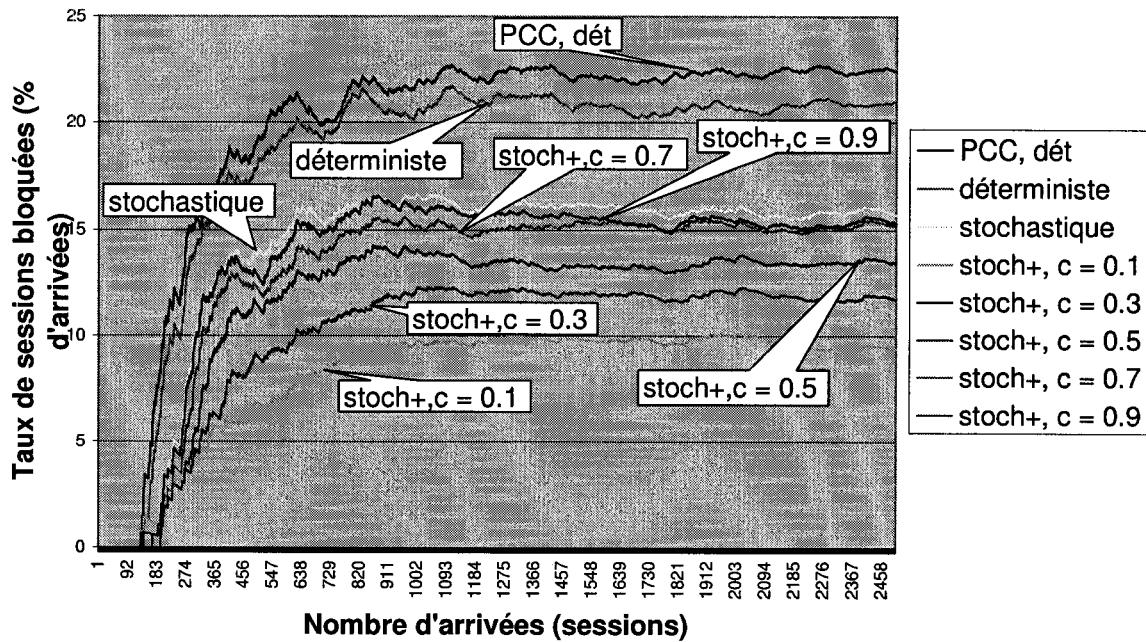


FIG. 5.11 *Taux de blocage pour un réseau de 50 noeuds, 1100 arcs*

TAB. 5.14 *Taux de blocage en % après 2500 sessions, réseau de 150 noeuds, VER. 2*

| Densité | Arcs  | Types d'algorithmes de routage |                         |             |             |                          |             |             |             |       |  |
|---------|-------|--------------------------------|-------------------------|-------------|-------------|--------------------------|-------------|-------------|-------------|-------|--|
|         |       | PCC                            | Énumération des chemins |             |             |                          |             |             |             |       |  |
|         |       |                                | dét                     | dét         | stoch       | stochastique+, VERSION 2 |             |             |             |       |  |
|         |       |                                |                         |             |             | c=0.1                    | c=0.3       | c=0.5       | c=0.7       | c=0.9 |  |
| 10%     | 1000  | 1.36                           | 6.40                    | 0.84        | <b>0.56</b> | 0.60                     | 0.64        | 0.76        | 0.76        |       |  |
| 30%     | 3000  | 22.88                          | 14.92                   | 5.32        | 5.36        | 5.44                     | <b>5.32</b> | <b>5.52</b> | <b>5.32</b> |       |  |
| 60%     | 6500  | 7.84                           | 3.48                    | 0.80        | <b>0.72</b> | 0.76                     | 0.88        | 0.80        | 0.92        |       |  |
| 90%     | 10000 | 5.04                           | 3.84                    | <b>2.44</b> | 2.56        | 2.52                     | 2.56        | 2.52        | 2.48        |       |  |

TAB. 5.15 *Taux de blocage en % après 2500 sessions, réseau de 150 noeuds, VER. 3*

| Densité | Arcs  | Types d'algorithmes de routage |       |                         |                          |             |             |       |       |
|---------|-------|--------------------------------|-------|-------------------------|--------------------------|-------------|-------------|-------|-------|
|         |       | PCC                            |       | Énumération des chemins |                          |             |             |       |       |
|         |       | dét                            | dét   | stoch                   | stochastique+, VERSION 3 |             |             |       |       |
|         |       |                                |       |                         | c=0.1                    | c=0.3       | c=0.5       | c=0.7 | c=0.9 |
| 10%     | 1000  | 2.08                           | 7.00  | 1.12                    | <b>0.00</b>              | <b>0.00</b> | <b>0.00</b> | 0.84  | 0.96  |
| 30%     | 3000  | 23.21                          | 11.12 | 5.40                    | <b>4.64</b>              | 4.88        | 4.84        | 4.88  | 5.20  |
| 60%     | 6500  | 7.56                           | 5.32  | 1.12                    | <b>0.00</b>              | 0.24        | 0.48        | 0.64  | 0.88  |
| 90%     | 10000 | 4.88                           | 3.84  | 2.68                    | <b>1.96</b>              | 2.20        | 2.48        | 2.64  | 2.60  |

endroits qui sont plus chargés, *avant* même que la congestion ne se produise. L'algorithme basé sur l'état moyen du réseau permet de "prévoir" les zones qui seront plus chargées et de répartir la charge de façon différente, pour réduire de façon significative le taux de blocage. Les résultats montrent que ces algorithmes permettent d'accepter plus de clients dans le réseau, tout en respectant leurs exigences de qualité de service.

Les performances des algorithmes qui tiennent compte de l'état moyen du réseau dans une proportion allant de 10% à 70% sont quelquefois assez rapprochées (par exemple, voir le tableau 5.14 pour le réseau de 3000 arcs, algorithmes "stoch+", "stoch+, c=0.3", "stoch+, c=0.5", "stoch+, c=0.7" et "stoch+, c=0.9"). Les résultats sont similaires en ce qui concerne le classement des algorithmes par rapport au taux de blocage obtenu ainsi que par rapport aux valeurs numériques elles-mêmes du taux de blocage. Les valeurs les plus faibles du taux de blocage sont obtenues avec la version 1 de l'algorithme de routage, où la probabilité de délai est basée sur une combinaison de l'état courant et de l'état moyen de chaque noeud (voir par

exemple les tableaux 5.13, 5.14 et 5.15). Cependant, la différence n'est pas très marquée avec les versions 2 et 3 de l'algorithme. Rappelons que dans la version 2 de l'algorithme, le contrôle d'admission est basé uniquement sur l'état courant du réseau, ce qui garantit que les exigences de qualité de service sont respectées au moment où la session est admise dans le réseau. Pour cette raison, la version 2 de l'algorithme est préférable aux deux autres, dans lesquelles le contrôle d'admission est fonction de l'état courant et de l'état moyen du réseau.

Les graphiques complets pour toutes les topologies de réseaux de 50 noeuds, ainsi que certains graphiques pour d'autres tailles de réseaux (100 et 150 noeuds) sont fournies en annexe.

## CONCLUSION

Dans ce mémoire, nous avons présenté une étude comparative d'algorithmes de routage dans un réseau sous la discipline de service GPS qui est largement utilisée pour mettre en oeuvre la réservation de ressources. Le critère de comparaison est le taux de blocage de sessions, que nous appelons "efficacité" du réseau. Les sessions sont bloquées lorsqu'on ne peut trouver un chemin satisfaisant leurs exigences de qualité de service.

Nous avons présenté une comparaison des modèles déterministe et stochastique de l'algorithme de service GPS. Le modèle déterministe offre des garanties strictes sur le délai, au prix d'une plus faible utilisation des ressources. Le modèle déterministe nécessite beaucoup de ressources car il doit garantir le délai même dans le pire cas, lorsque toutes les sources transmettent avec leur débit maximal. Dans le modèle stochastique, on accepte une certaine probabilité de dépasser les limites de délai, mais on augmente l'efficacité du réseau.

Nous avons proposé aussi un algorithme de routage qui évalue des chemins disjoints pour une paire de noeuds et choisit le chemin qui offre les meilleures caractéristiques de qualité de service pour la nouvelle session. Nous avons montré que l'utilisation d'un tel algorithme permet d'augmenter l'efficacité du réseau. Le modèle stochastique, combiné avec l'algorithme des chemins disjoints, permet

d'obtenir une efficacité supérieure à celle obtenue avec le modèle déterministe de GPS.

Nous avons déterminé que l'avantage de l'algorithme des chemins disjoints est plus important dans les réseaux denses. Lorsque la densité des liens est élevée, il y plus de chemins disjoints pour une paire de noeuds, ce qui augmente les chances de trouver un chemin acceptable et diminue le taux de blocage. Inversement, dans les réseaux épars, la faible densité de liens ne permet pas d'avoir beaucoup de chemins disjoints, ce qui diminue les chances de trouver un chemin respectant les exigences de qualité de service. Dans les réseaux épars, il arrive donc que l'algorithme des plus courts chemins soit plus efficace que l'algorithme d'énumération des chemins disjoints.

Finalement, nous avons proposé un nouvel algorithme de routage, dans lequel les calculs sont basés en partie sur un état moyen du réseau. Cet état moyen peut être le résultat de l'observation du réseau à long terme, dans le but de représenter les zones où la congestion a tendance à se produire. Le poids de l'état moyen dans les calculs est un paramètre de la simulation.

Les simulations montrent que la connaissance de l'état moyen du réseau au moment du routage est bénéfique du point de vue de l'efficacité du réseau. En connaissant la charge moyenne sur les noeuds, on peut acheminer les sessions de façon à éviter certains endroits qui sont plus chargés, *avant* même que la congestion ne se produise. Ces algorithmes permettent de prévoir les zones qui seront plus

chargées et de répartir la charge d'une façon différente, pour éviter de bloquer les sessions faute de ressources suffisantes. Les algorithmes basés sur l'état moyen du réseau sont les plus performants, pour des réseaux de tailles et de densités variées, surtout lorsque le poids de l'état moyen dans les calculs dépasse 50%. Il est à noter que notre simulation a été conduite avec un taux d'arrivée de sessions qui est constant. Les résultats en régime non-stationnaire pourraient être différents de ceux présentés ici.

Nous avons comparé trois versions de l'algorithme prédictif qui diffèrent par la façon de combiner l'information sur l'état courant et l'état moyen du réseau. Les performances de ces trois versions sont assez proches, cependant, l'algorithme stochastique+, version 2 est préférable, car le contrôle d'admission y est basé seulement sur l'état courant du réseau, ce qui garantit que les exigences de qualité de service sont respectées au moment où la session est admise dans le réseau. Le routage dans cet algorithme est basé sur une fonction linéaire de l'état courant et de l'état moyen du réseau.

Cet algorithme de routage est réalisable en pratique. Le noeud qui prend la décision de routage doit connaître seulement la charge sur les autres noeuds du réseau. L'algorithme doit maintenir un ensemble de chemins disjoints pour toutes les paires de noeuds du réseau. Ces chemins peuvent aussi être calculés au besoin, lorsque les sessions arrivent dans le réseau.

Dans les recherches futures, il serait intéressant de faire des simulations avec des

topologies et des modèles de trafic plus réalistes. On pourrait utiliser une topologie de réseau qui approxime mieux la structure hiérarchique de l'Internet. En ce qui concerne le trafic, nous avons utilisé des processus de type ON-OFF. En pratique, le trafic dans les réseaux de communication par paquets est représenté par des processus plus complexes. Il serait intéressant de trouver des caractérisations EBB pour les processus stochastiques qui approximent le trafic observé dans l'Internet. De plus, on pourrait utiliser des taux d'arrivées des sessions qui varient dans le temps, de façon à simuler les pointes de trafic et évaluer le comportement des algorithmes en régime non-stationnaire.

Un autre élément de la simulation qui peut être modifié est la façon de générer l'ensemble de chemins candidats pour une nouvelle session. Dans notre étude, l'algorithme des chemins disjoints est utilisé pour générer cet ensemble de chemins. D'autres algorithmes peuvent être envisagés, par exemple l'algorithme des  $k$ -plus courts chemins. De plus, cet algorithme peut être modifié pour enlever les liens qui ne sont pas favorables du point de vue de la qualité de service.

## RÉFÉRENCES

- [1] Demers A, Keshav S, et Shenker S. « Analysis and simulation of a fair queueing algorithm ». *Internetworking, Research and Experience*, 1:3–26, septembre 1990.
- [2] Girard A et Rosenberg C. « Optimal Rate Allocation for Maximum-Delay GPS Networks ». Dans *Proceedings of the 8th International Conference on Advances in Communications and Control: Telecommunications/Signal Processing, COMCON 8*, juin 2001.
- [3] Greenberg A et Madras N. « How fair is fair queueing ». *Journal of the Association for Computing Machinery*, 39(3):568–598, juillet 1992.
- [4] Orda A et Sprintson A. « QoS Routing: The Precomputation Perspective ». Dans *Proceedings of IEEE INFOCOM 2000*, pages 128–136, mars 2000.
- [5] Parekh A et Gallager R. « A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - The Single Node Case ». *IEEE/ACM Transactions on Networking*, 1(3):344–357, juin 1993.
- [6] Parekh A et Gallager R. « A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks - The Multiple Node Case ». *IEEE/ACM Transactions on Networking*, 2(2):137–150, avril 1994.

- [7] Li B, Hamdi M, Cao X, et Thomas Hou Y. « QoS-enabled voice support in the next-generation Internet: Issues, existing approaches and challenges ». *IEEE Communications Magazine*, 38(4):54–61, avril 2000.
- [8] CISCO. « Cisco Tech Notes. OSPF Design Guide, section 1 », 2002.
- [9] Clark D, Shenker S, et Zhang L. « Supporting real-time applications in an integrated services packet network: Architecture and mechanism ». *ACM Computer Communication Review*, 22(4):14–26, juillet 1992.
- [10] Stiliadis D et Varma A. « Design and analysis of Frame-based Fair Queueing: A new traffic scheduling algorithm for packet switched networks ». Dans *Proceedings of ACM SIGMETRICS'96*, pages 104–115, mai 1996.
- [11] Mills D.L. « DCN Local-Network Protocols – RFC no 891 », décembre 1983.
- [12] Crawley E, Nair R, Rajagopalan B, et Sandick H. « A framework for QoS-based routing in the internet – RFC no 2386 », août 1998.
- [13] Gaines G et Festa M. « A Survey of RSVP/QoS Implementations (Update 2), RSVP Working Group », juillet 1998.
- [14] Schulzrinne H, Casner S, Frederick R, et Jacobson V. « RTP: A Transport Protocol for Real-Time Applications – RFC no 1889 », janvier 1996.

- [15] Cidon I, Rom R, et Shavitt Y. « Multi-Path Routing combined with Resource Reservation ». Dans *Proceedings of IEEE INFOCOM'97*, pages 92–100, avril 1997.
- [16] Heinanen J, Baker F, Weiss W, et Wroclawski J. « Assured Forwarding PHB Group – RFC no 2597 », juin 1999.
- [17] Moy J. « OSPF Version 2 – RFC no 2328 », avril 1998.
- [18] Turner J. « New directions in communications (or Which way to the Information Age?) ». *IEEE Communications Magazine*, 24(10):8–15, octobre 1986.
- [19] Wroclawski J. « The Use of RSVP with IETF Integrated Services – RFC no 2210 », septembre 1997.
- [20] Bennet J.C.R et Zhang H. «  $WF^2Q$ -Worst-case fair weighted fair queueing ». Dans *Proceedings of IEEE INFOCOM'96*, pages 120–128, mars 1996.
- [21] Nichols K, Blake S, Baker F, et Black D. « Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers – RFC no 2474 », décembre 1998.
- [22] Georgiadis L, Guerin R, et Parekh A. « Optimal multiplexing on a single link: Delay and buffer requirements ». *IEEE Transactions on Information Theory*, 43(5):1518–1535, septembre 1997.

- [23] Migliore M, Martorana V, et Sciortino F. « An Algorithm to Find All Paths between Two Nodes in a Graph ». *Journal of Computational Physics*, 87(1):231–236, 1990.
- [24] Shreedhar M et Varghese G. « Efficient fair queuing using deficit round-robin ». *IEEE/ACM Transactions on Networking*, 4(3):375–385, juin 1996.
- [25] Rao N et Batsell S. « QoS routing via multiple paths using bandwidth reservation ». Dans *Proceedings of IEEE INFOCOM'98*, volume 1, pages 11–18, mars 1998.
- [26] Taft-Plotkin N, Bellur B, et Ogier R. « Quality-of-Service Routing Using Maximally Disjoint Paths ». Dans *Proceedings of the IEEE International Workshop on Quality of Service (IWQoS)*, pages 119–128, juin 1999.
- [27] Yaron O et Sidi M. « Performance and Stability of Communication Networks Via Robust Exponential Bounds ». *IEEE/ACM Transactions on Networking*, 1(3):372–385, juin 1993.
- [28] Yaron O et Sidi M. « Generalized Processor Sharing Networks with Exponentially Bounded Burstiness Arrivals ». Dans *Proceedings of IEEE INFOCOM'94*, pages 628–634, juin 1994.

- [29] Goyal P et Vin H. « Generalized guaranteed rate scheduling algorithms: a framework ». *IEEE/ACM Transactions on Networking*, 5(4):561–572, août 1997.
- [30] Goyal P, Lam S, et Vin H. « Determining end-to-end delay bounds in heterogeneous networks ». *ACM/Springer Verlag Multimedia Systems Journal*, 5(3):157–163, mai 1997.
- [31] Ma Q et Steenkiste P. « On Path Selection for Traffic with Bandwidth Guarantees ». Dans *Proceedings of IEEE International Conference on Network Protocols*, pages 191–202, octobre 1997.
- [32] Ma Q et Steenkiste P. « Quality of Service Routing for Traffic with Performance Guarantees ». Dans *Proceedings of IFIP Fifth International Workshop on Quality of Service*, pages 115–126, mai 1997.
- [33] Ma Q, Steenkiste P, et Zhang H. « Routing High Bandwidth Traffic in Max-Min Fair Share Networks ». Dans *Proceedings of ACM SIGCOMM'96*, pages 206–217, août 1996.
- [34] Braden R, Clark D, et Shenker S. « Integrated Services in the Internet Architecture: an Overview – RFC no 1633 », juin 1994.
- [35] Guérin R et Peris V. « Quality-of-service in packet networks: Basic mechanisms and directions ». *Computer Networks*, 31(3):169–189, février 1999.

- [36] Blake S, Black D, Carlson M, Davies E, Wang Z, et Weiss W. « An Architecture for Differentiated Services – RFC no 2475 », décembre 1998.
- [37] Chen S et Nahrstedt K. « An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions ». *IEEE Network Magazine*, 12(6):64–79, novembre 1998.
- [38] Jacobson V, Nichols K, et Poduri K. « An Expedited Forwarding PHB – RFC no 2598 », juin 1999.
- [39] Sivaraman V et Chiussi F. « Providing End-to-End Statistical Delay Guarantees with Earliest Deadline First Scheduling and Per-Hop Traffic Shaping ». Dans *Proceedings of IEEE INFOCOM 2000*, pages 631–640, mars 2000.
- [40] Zhao W, Olshefski D, et Schulzrinne H. « Internet Quality of Service: an Overview ». Rapport Technique CUCS-003-00, Department of Computer Science, Columbia University, février 2000.
- [41] Zhao W et Tripathi S. « Routing guaranteed quality of service connections in integrated services packet networks ». Dans *Proceedings of IEEE International Conference on Network Protocols*, pages 175–182, octobre 1997.
- [42] Yuan X, Ding H, Zhong Y, et Zhang J. « Resource Reservation Mechanisms for Distributed Multi-path Quality of Service Routing ». Rapport Technique

TR-000701, Department of Computer Science, Florida State University, juillet 2000.

- [43] Bernet Y. « A Framework for use of RSVP with Diff-serv Networks – Internet draft », juin 1998.
- [44] Bernet Y. « The Complementary Roles of RSVP and Differentiated Services in the Full-Service QoS Network ». *IEEE Communications Magazine*, pages 154–162, février 2000.
- [45] Wang Z et Crowcroft J. « Quality-of-Service Routing for Supporting Multimedia Applications ». *Journal on Selected Areas in Communications*, 14(7):1228–1234, septembre 1996.
- [46] Zhang Z, Towsley D, et Kurose J. « Statistical Analysis of the Generalized Processor Sharing Scheduling Discipline ». *Journal on Selected Areas in Communications*, 13(6):1071–1080, août 1995.

## Annexe I

### Résultats numériques

I.1 Réseau de 50 noeuds, version 1

I.2 Réseau de 50 noeuds, version 2

I.3 Réseau de 50 noeuds, version 3

I.4 Autres résultats pour des réseaux de 100 noeuds et 150 noeuds

### Taux de blocage de sessions, ver. 1

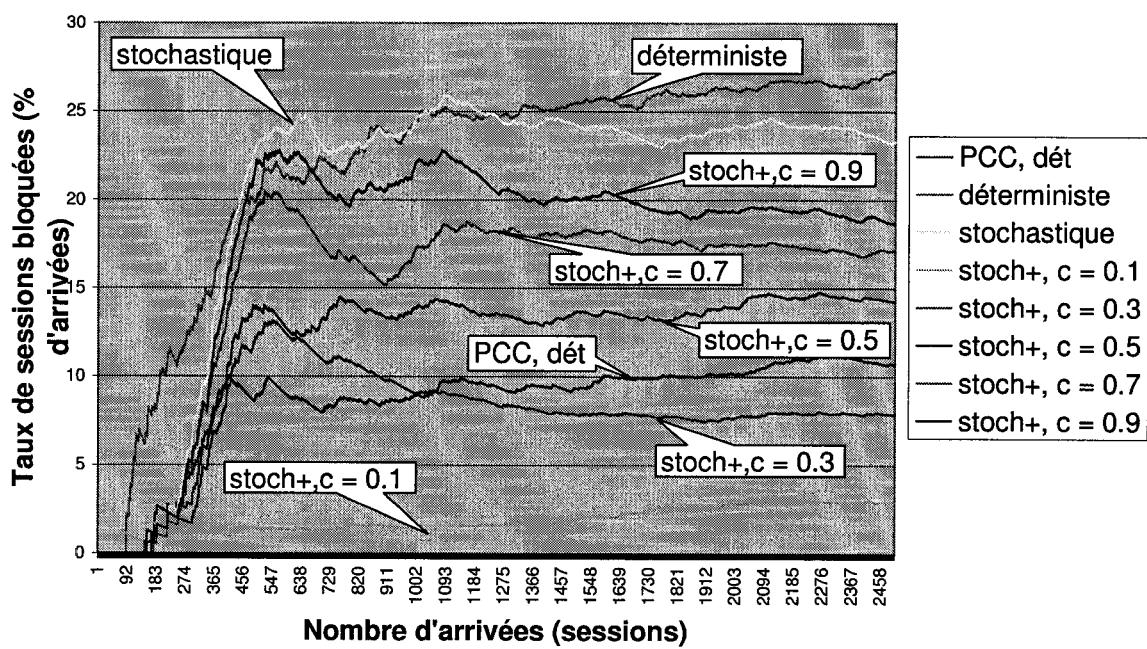


FIG. I.1 *Taux de blocage pour un réseau de 50 noeuds, 100 arcs*

### Taux de blocage de sessions, ver. 1

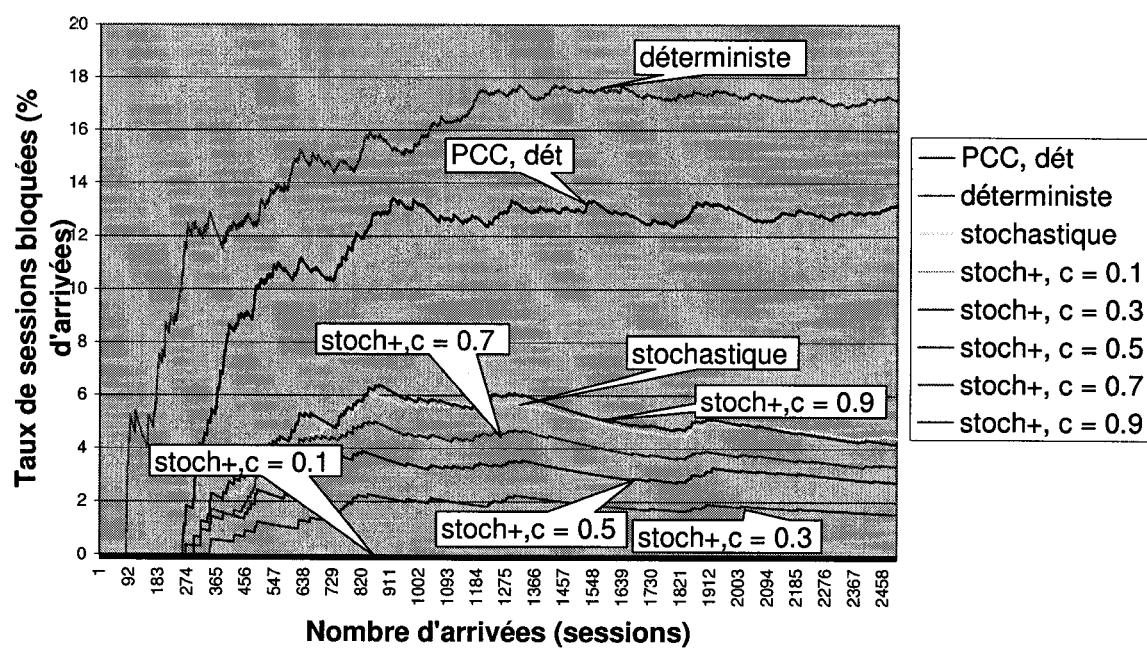


FIG. I.2 Taux de blocage pour un réseau de 50 noeuds, 350 arcs

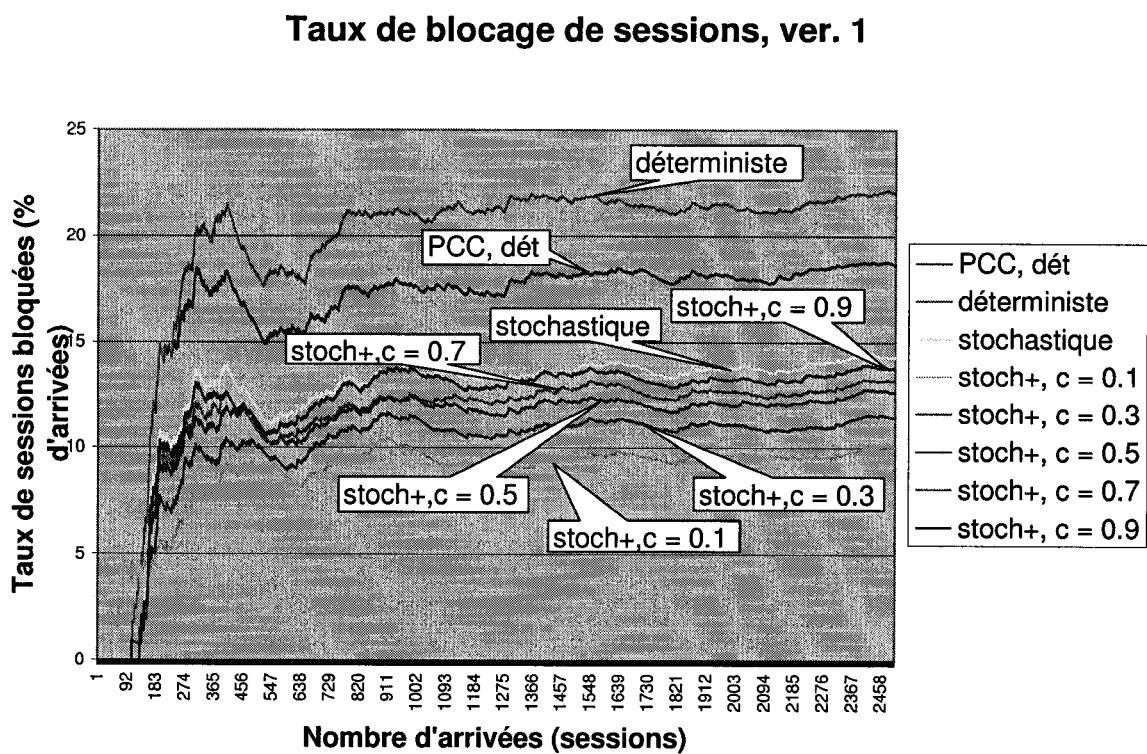


FIG. I.3 Taux de blocage pour un réseau de 50 noeuds, 700 arcs

### Taux de blocage de sessions, ver. 1

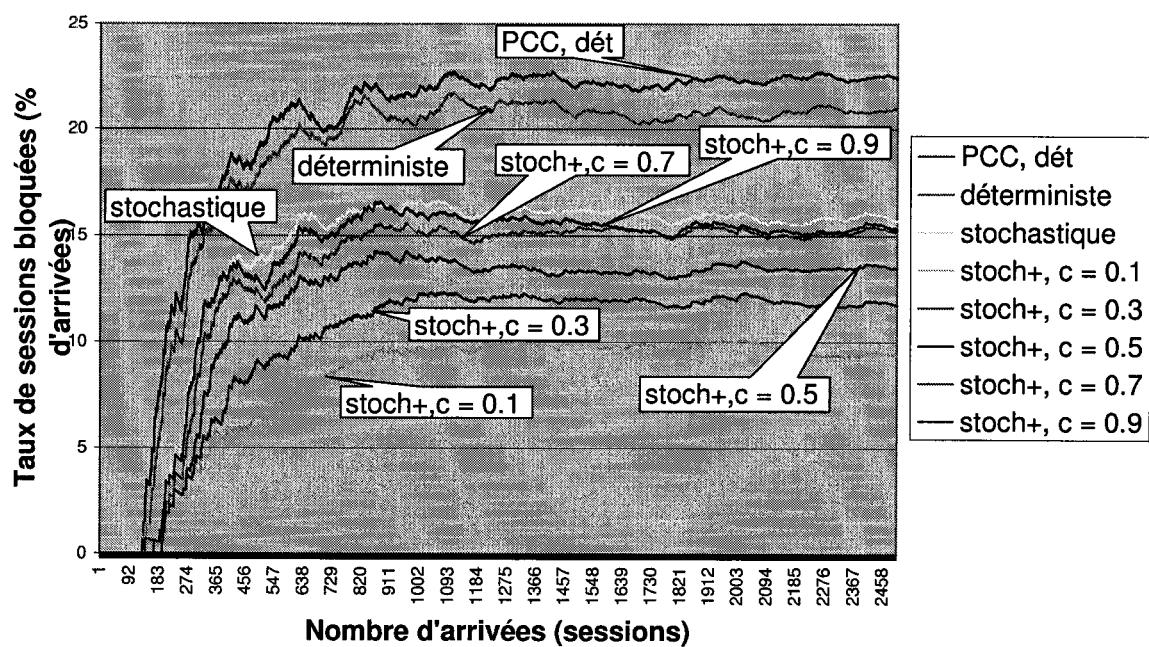


FIG. I.4 Taux de blocage pour un réseau de 50 noeuds, 1100 arcs

### Taux de blocage de sessions, ver. 2

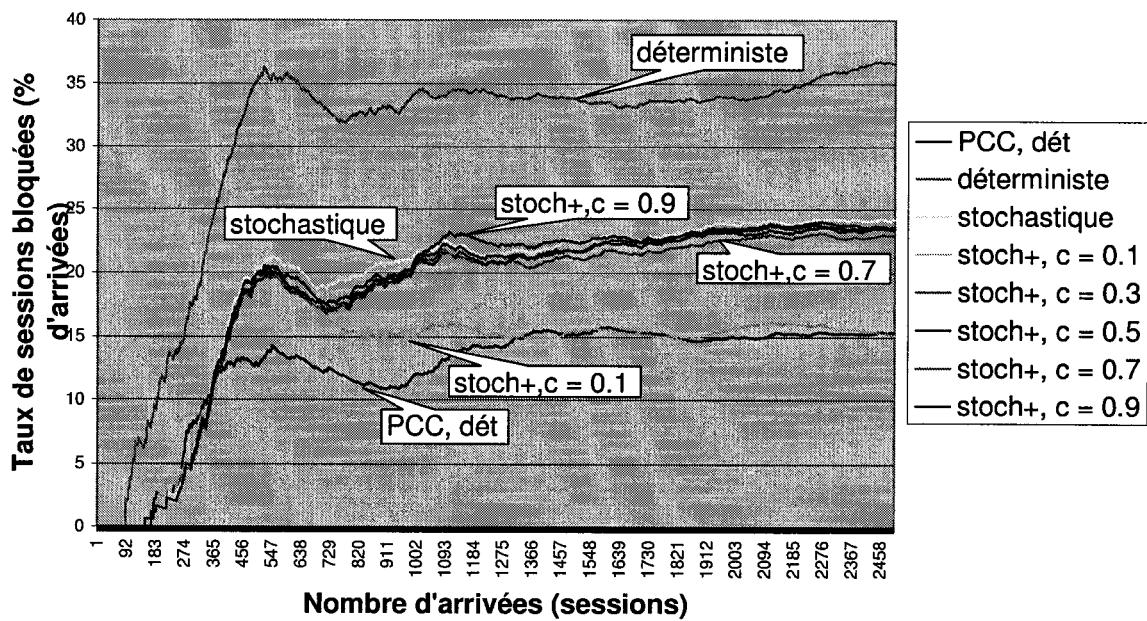


FIG. I.5 Taux de blocage pour un réseau de 50 noeuds, 100 arcs

### Taux de blocage de sessions, ver. 2

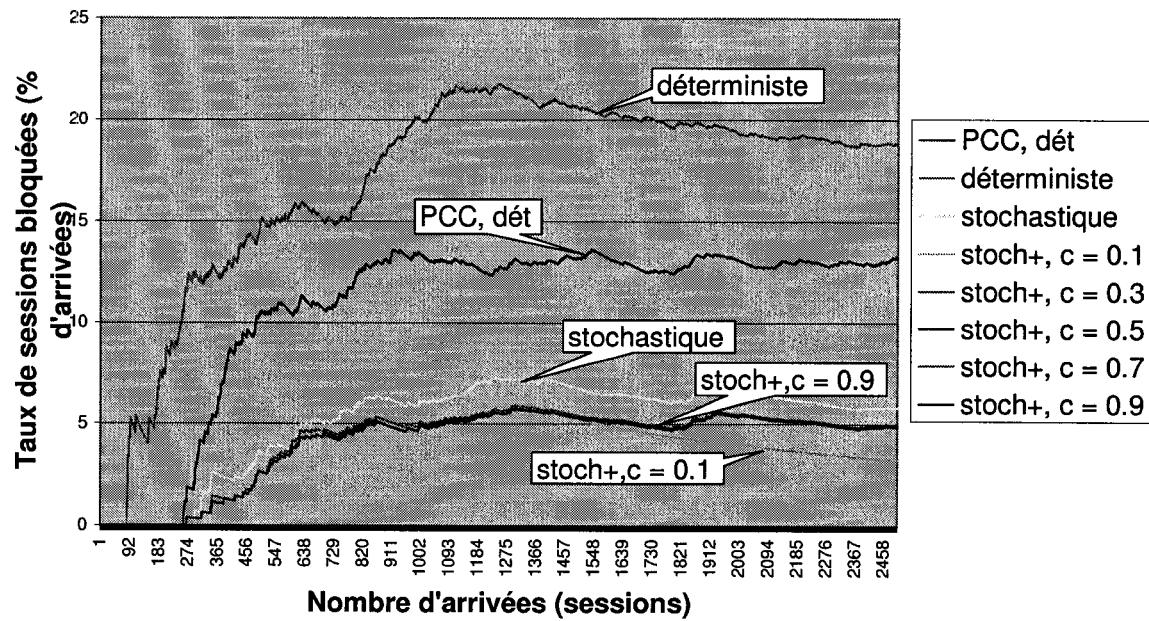


FIG. I.6 *Taux de blocage pour un réseau de 50 noeuds, 350 arcs*

### Taux de blocage de sessions, ver. 2

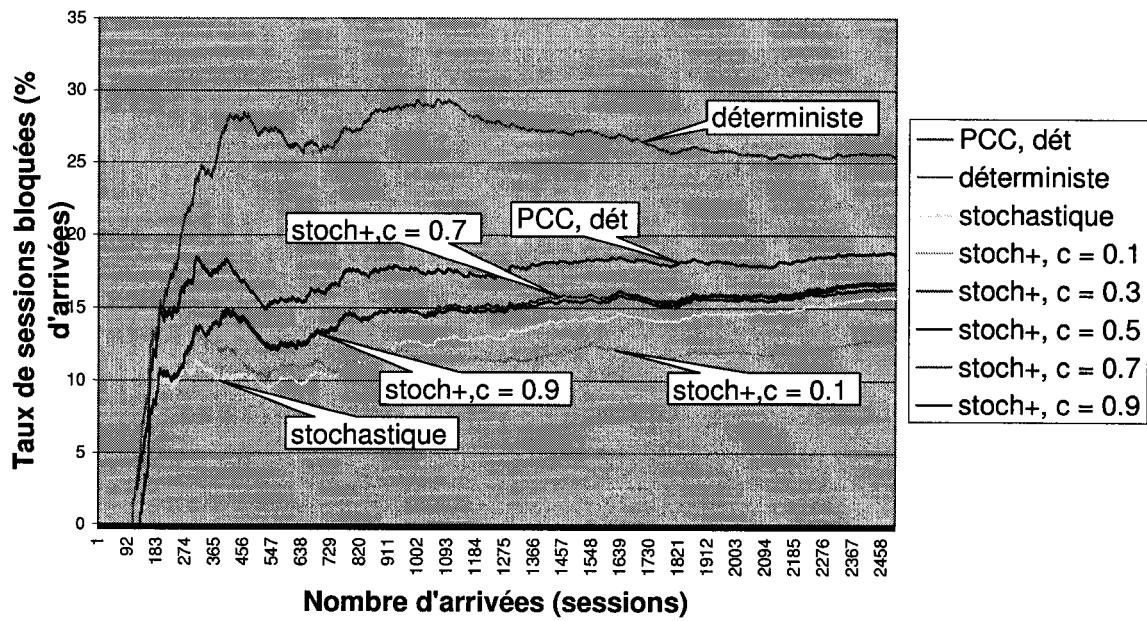


FIG. I.7 Taux de blocage pour un réseau de 50 noeuds, 700 arcs

### Taux de blocage de sessions, ver. 2

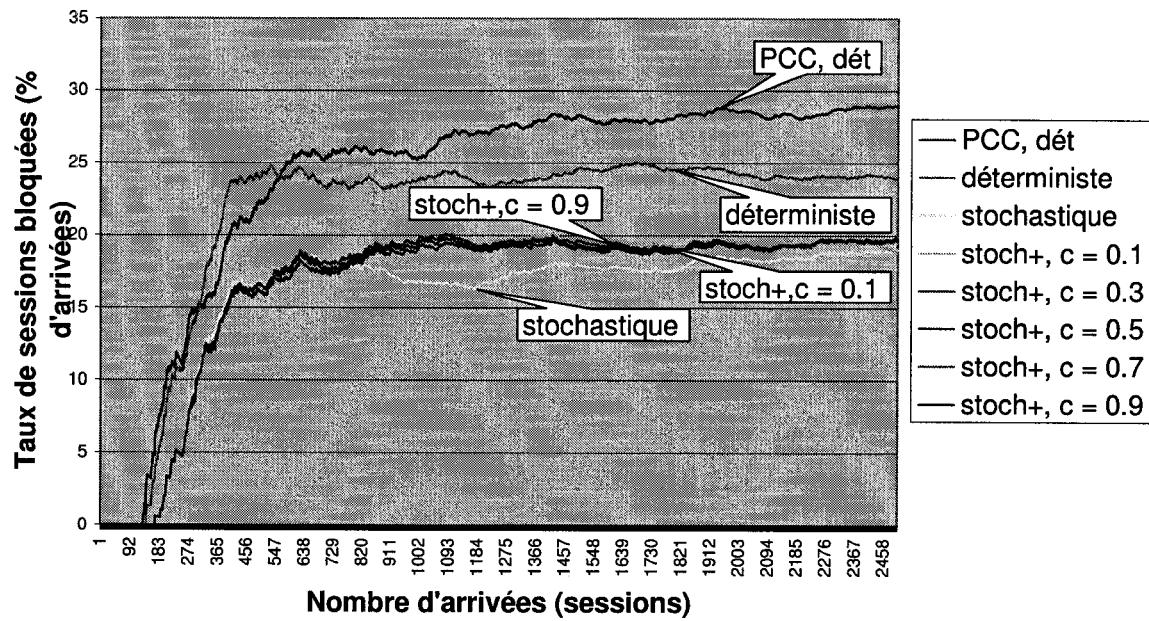


FIG. I.8 Taux de blocage pour un réseau de 50 noeuds, 1100 arcs

### Taux de blocage de sessions, ver. 3

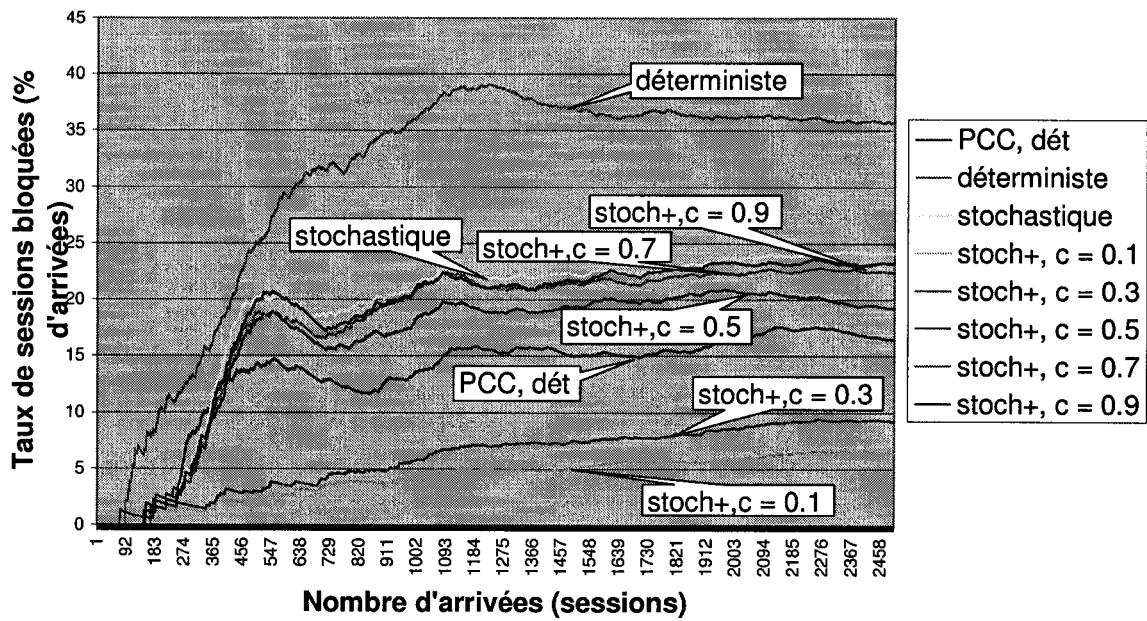


FIG. I.9 Taux de blocage pour un réseau de 50 noeuds, 100 arcs

### Taux de blocage de sessions, ver. 3

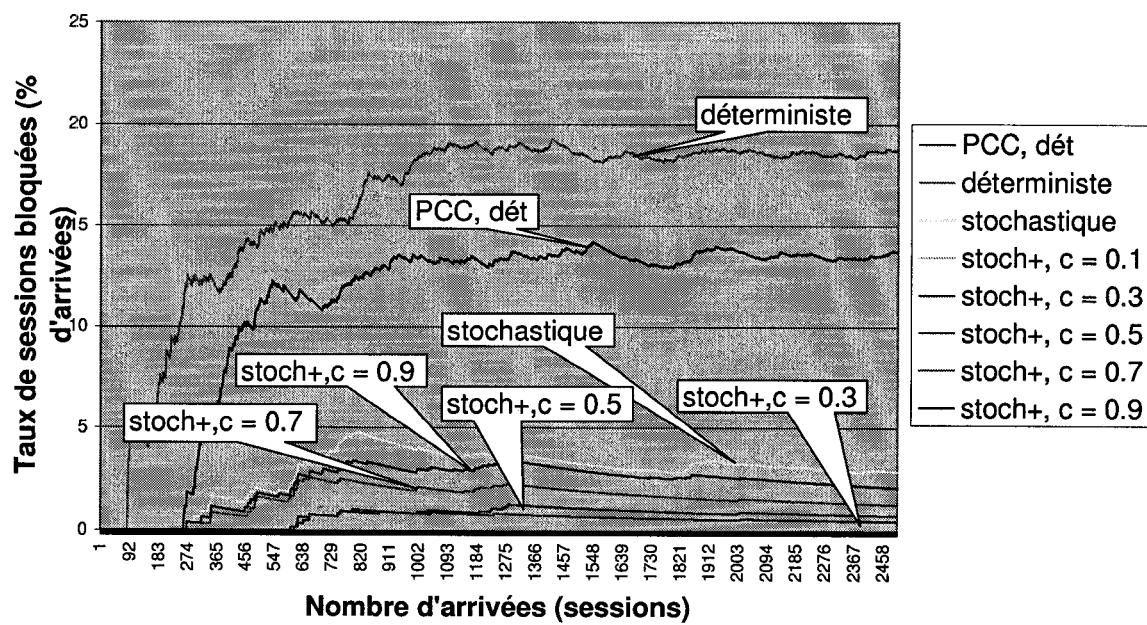


FIG. I.10 Taux de blocage pour un réseau de 50 noeuds, 350 arcs

### Taux de blocage de sessions, ver. 3

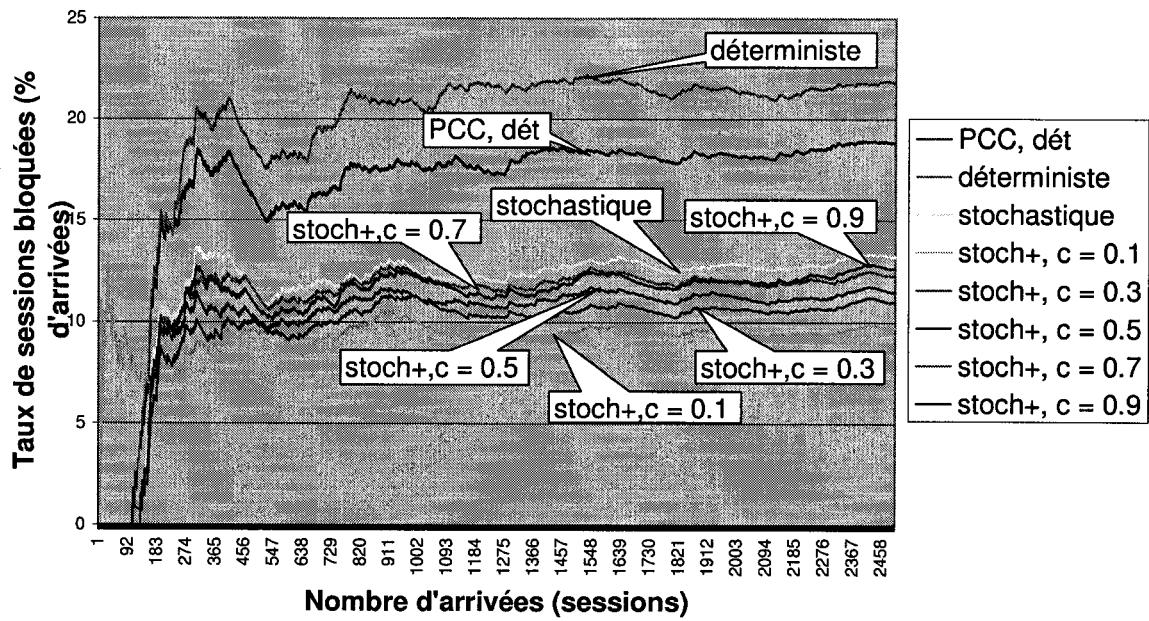


FIG. I.11 *Taux de blocage pour un réseau de 50 noeuds, 700 arcs*

### Taux de blocage de sessions, ver. 3

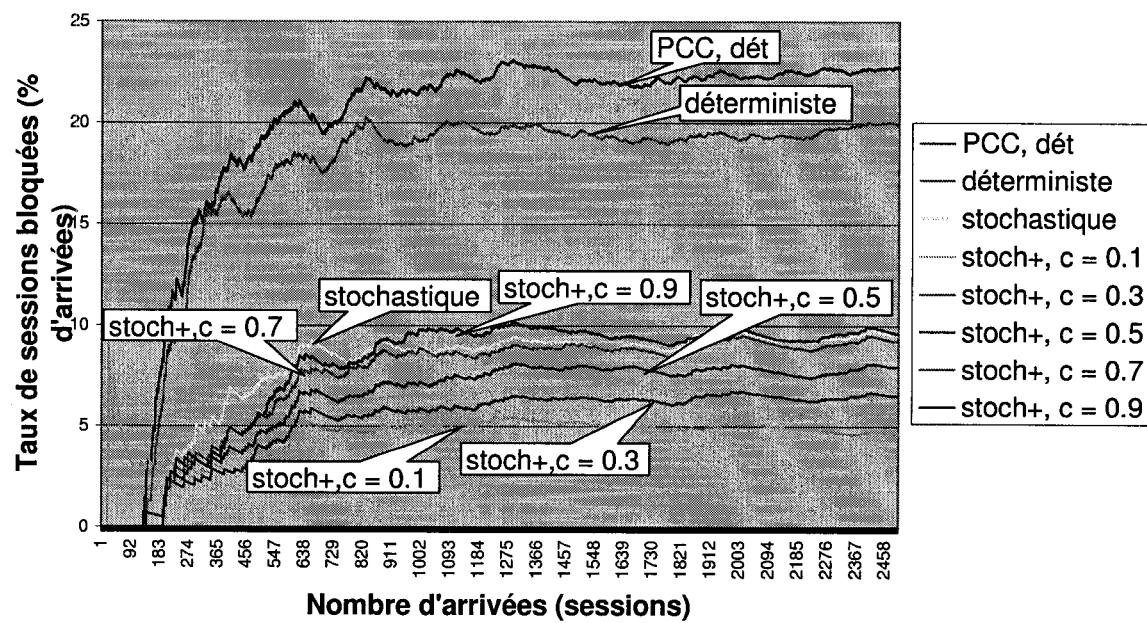


FIG. I.12 Taux de blocage pour un réseau de 50 noeuds, 1100 arcs

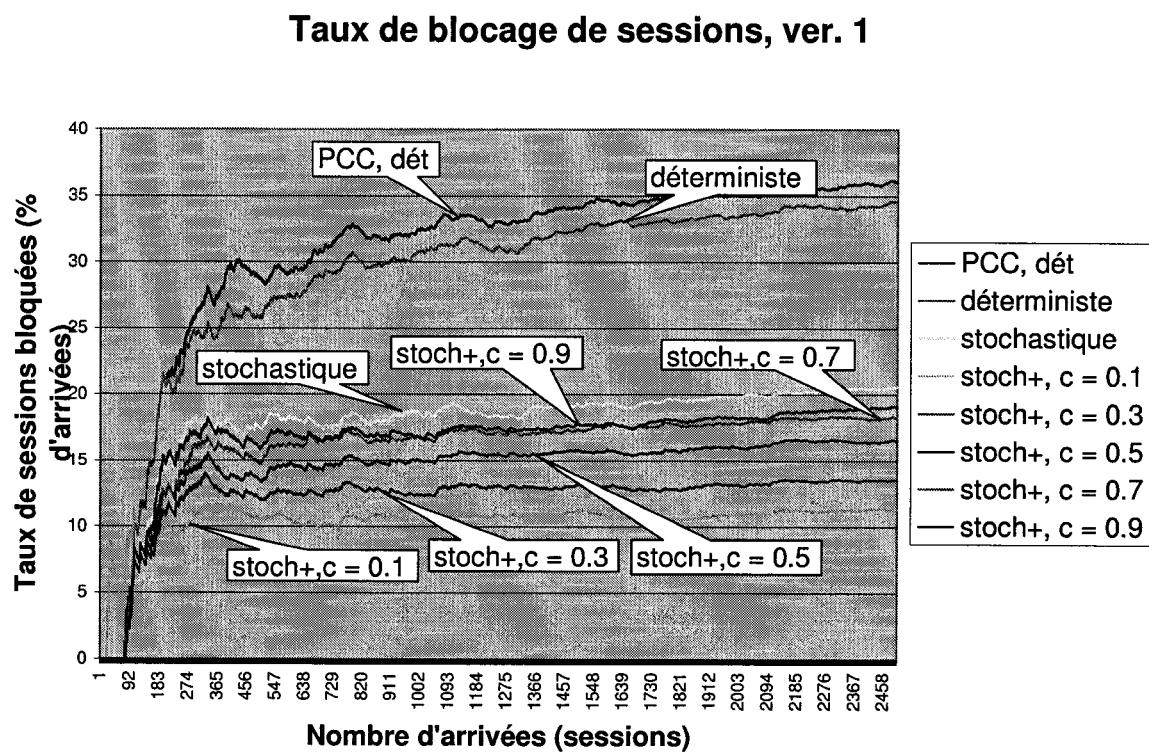


FIG. I.13 *Taux de blocage pour un réseau de 100 noeuds, 4500 arcs, version 1*

### Taux de blocage de sessions, ver. 2

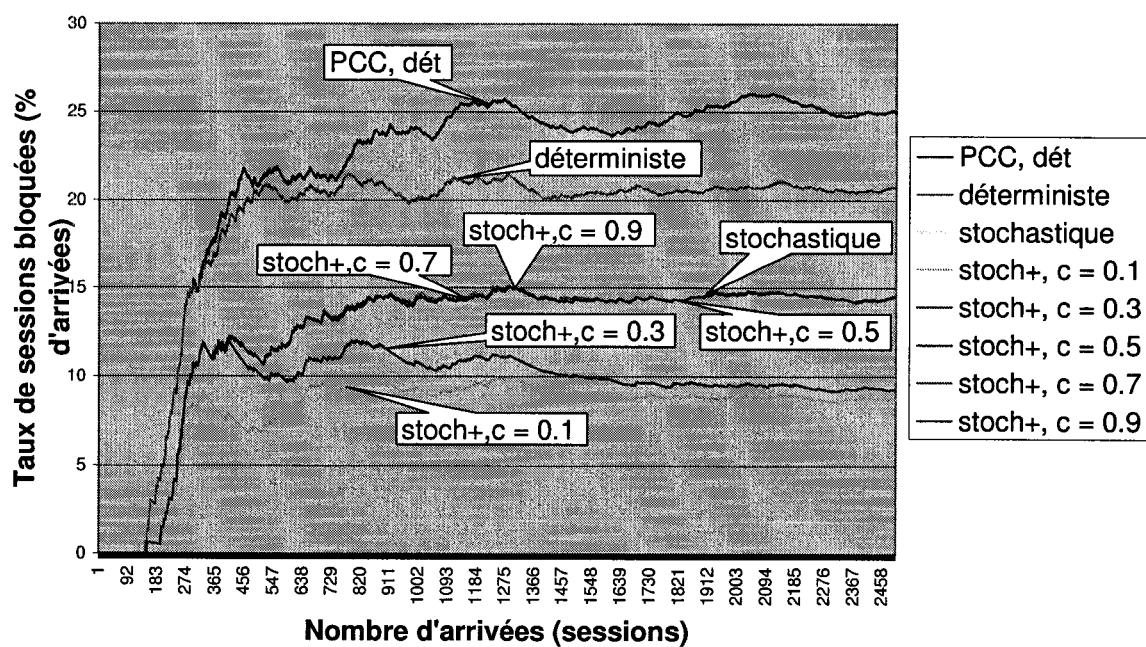


FIG. I.14 Taux de blocage pour un réseau de 100 noeuds, 500 arcs, version 2

### Taux de blocage de sessions, ver. 3

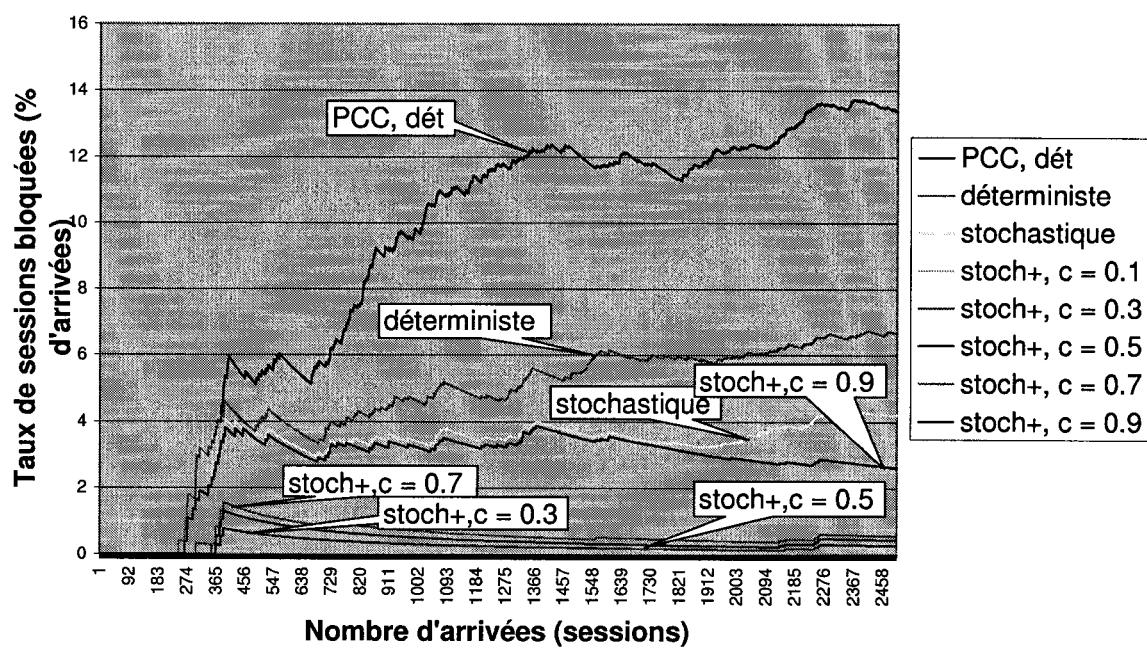


FIG. I.15 Taux de blocage pour un réseau de 100 noeuds, 3000 arcs, version 3

### Taux de blocage de sessions, ver. 1

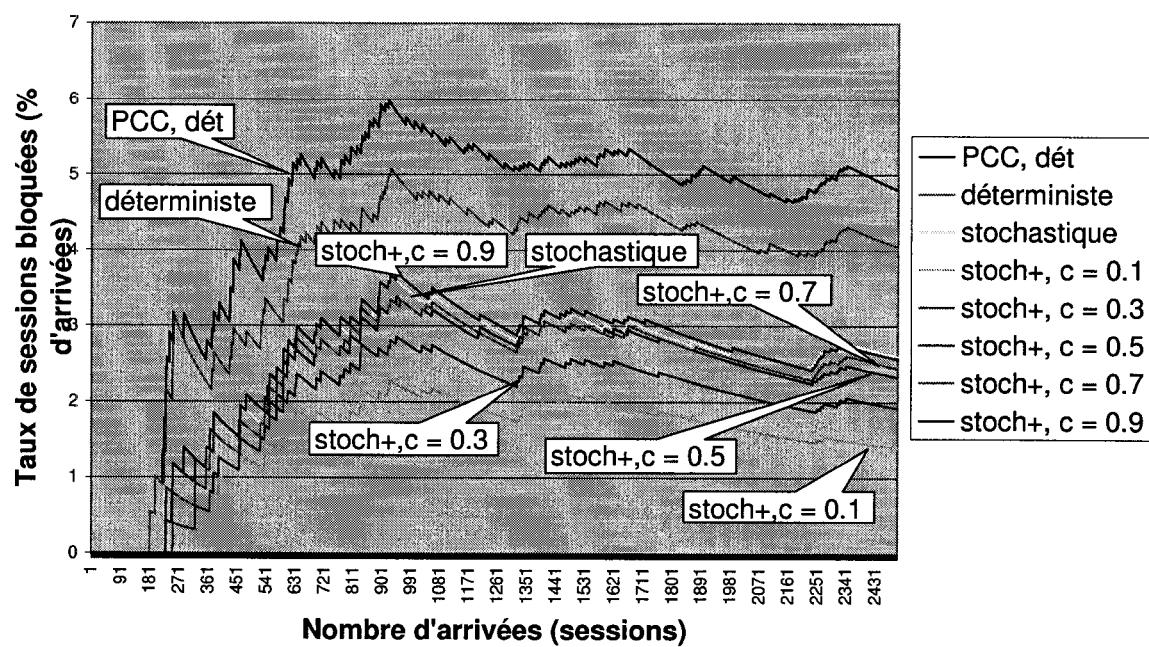


FIG. I.16 *Taux de blocage pour un réseau de 150 noeuds, 10000 arcs, version 1*

### Taux de blocage de sessions, ver. 2

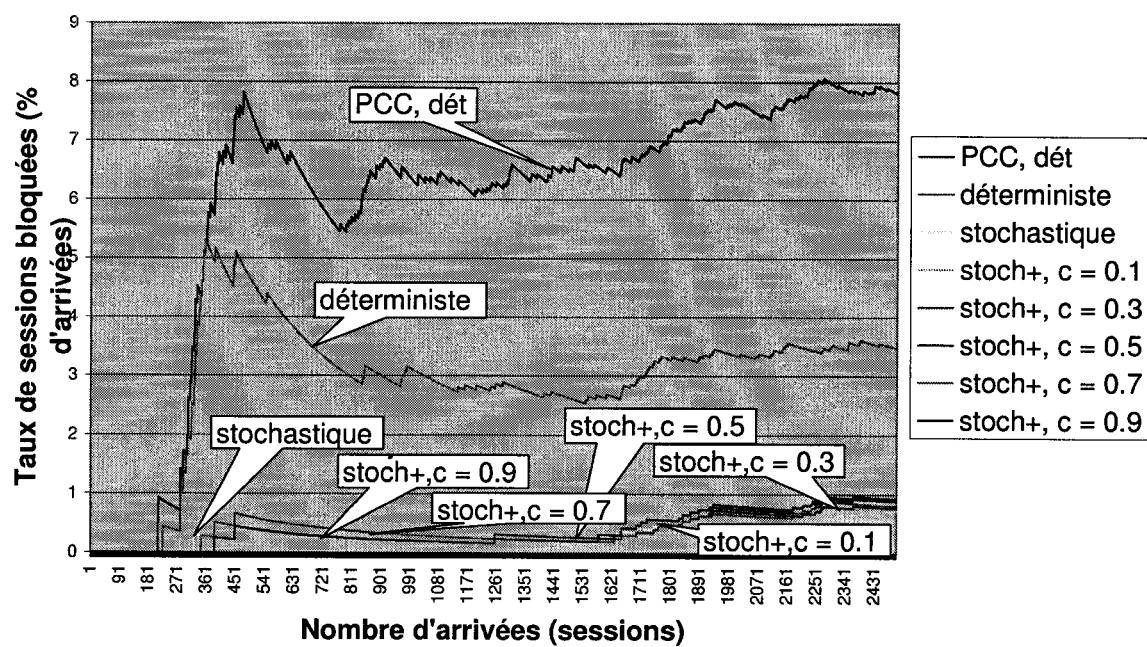


FIG. I.17 Taux de blocage pour un réseau de 150 noeuds, 6500 arcs, version 2

### Taux de blocage de sessions, ver. 3

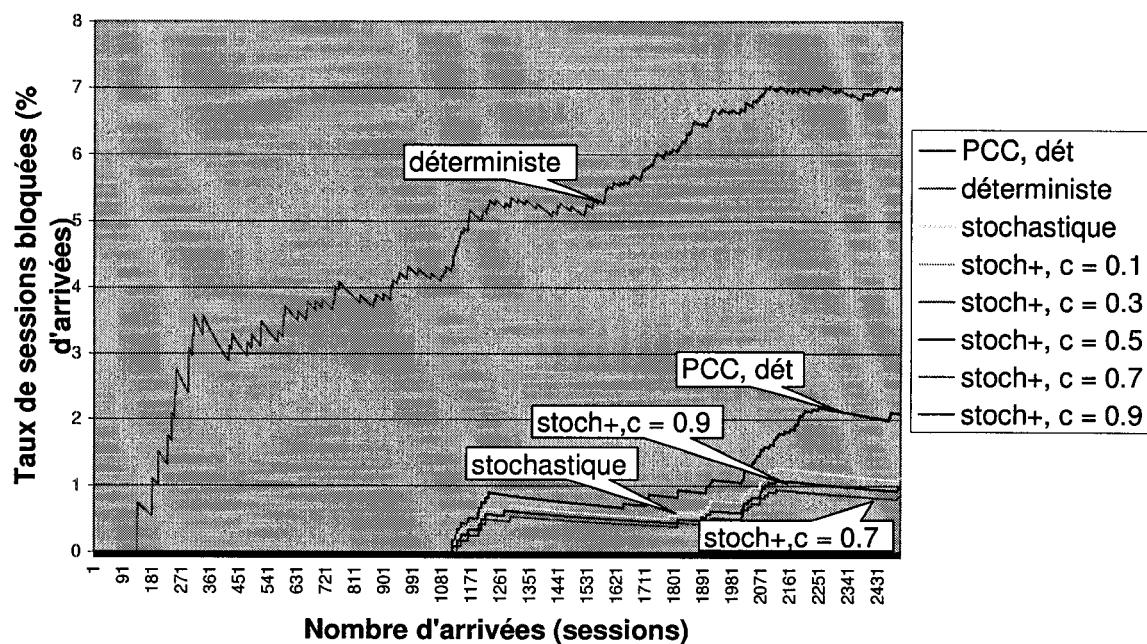


FIG. I.18 Taux de blocage pour un réseau de 150 noeuds, 1000 arcs, version 3