

**Titre:** Algorithmes de routage dynamique dans les réseaux optiques  
Title:

**Auteur:** Sedogbo Rolland W. Mewanou  
Author:

**Date:** 2003

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

**Référence:** Mewanou, S. R. W. (2003). Algorithmes de routage dynamique dans les réseaux optiques [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.  
Citation: <https://publications.polymtl.ca/6986/>

## Document en libre accès dans PolyPublie

Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/6986/>  
PolyPublie URL:

**Directeurs de recherche:** Samuel Pierre  
Advisors:

**Programme:** Génie électrique  
Program:

UNIVERSITÉ DE MONTRÉAL

**ALGORITHMES DE ROUTAGE DYNAMIQUE  
DANS LES RÉSEAUX OPTIQUES**

SEDOGBO ROLLAND W. MEWANOU

DÉPARTEMENT DE GÉNIE INFORMATIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

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

JANVIER 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

Our file Notre référence

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.

0-612-81554-4

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

**ALGORITHMES DE ROUTAGE DYNAMIQUE  
DANS LES RÉSEAUX OPTIQUES**

présenté par : MEWANOU Sèdogbo Rolland W.

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

A été dûment accepté par le jury d'examen composé de :

M. QUINTERO Alejandro, Ph.D., président

M. PIERRE Samuel, Ph.D., membre et directeur de recherche

M. GALINIER Philippe, Ph.D., membre

*A mes parents et à Chantal,  
Avec affection, amour et gratitude...*

## REMERCIEMENTS

Je désire remercier tout d'abord mon directeur de recherche, le professeur Samuel Pierre, pour avoir cru en moi et m'avoir donné tout au long de ma recherche les moyens, les conseils et la patience nécessaire à l'accomplissement de ce mémoire.

Je voudrais ensuite remercier tous mes collègues du Laboratoire de Recherche en Réseautique et Informatique Mobile (LARIM), pour leurs critiques constructives très appréciées lors de la réalisation de ce travail. Ma gratitude va spécialement à l'égard de Fabien Houéto, Sabine Kébreau et Edgar Olougouna qui m'ont apporté leurs conseils et soutiens permanents.

Je ne saurais continuer sans adresser toute ma reconnaissance et mes remerciements les plus chaleureux à mes parents et mes frères et sœurs qui ont su m'accompagner de leurs prières et assistance tout au long de ce parcours.

Merci enfin à toutes les personnes qui m'ont accordé amour, amitié, soutien et réconfort lors de mes études. Chacune d'entre elles saura se reconnaître et me pardonner de ne pas les citer pour ne pas dépasser les limites raisonnables de cette page.

## RÉSUMÉ

Au cours des dernières années, de nombreuses applications multimédia ont nécessité un taux de transfert des données ultra-rapide et une haute fiabilité des canaux de transmission. Dans ce contexte, les réseaux optiques basés sur la technologie WDM représentent par leur grande largeur de bande une alternative intéressante aux réseaux de communication actuels. La gestion de ces réseaux a entraîné plusieurs problèmes parmi lesquels celui du routage dynamique qui permet d'acheminer les données dans le réseau en tenant compte de l'état des nœuds et des liaisons. À cause de l'absence de mémoire tampon suffisante au niveau des nœuds pour stocker les données qui transitent dans le réseau, on affecte un chemin spécifique aux connexions entre une source et une destination. Dans les réseaux optiques dépourvus de convertisseurs de longueur d'onde, ce problème devient plus complexe puisqu'il faut transférer les données entre un nœud source et un nœud destination sur une seule et même longueur d'onde. Pour un ensemble de requêtes de connexion générées dynamiquement pour des paires source-destination (s-d), il faut donc déterminer le meilleur chemin parmi  $k$  chemins possibles, en respectant les contraintes de coût et de continuité de longueur d'onde. L'optimisation de ce processus permettra alors de minimiser la congestion du réseau et par conséquent le taux de blocage des connexions entrantes.

Ce mémoire a pour but essentiel la mise au point de nouveaux algorithmes de routage dynamique pour améliorer les performances obtenues par les algorithmes existants. Pour atteindre nos objectifs, nous avons commencé par une formulation mathématique du problème ci-dessus exposé. Nous avons ensuite proposé une nouvelle approche pour estimer la congestion sur un chemin ainsi qu'un critère additionnel pour analyser les liaisons d'un chemin. Nous en avons alors déduit deux nouveaux algorithmes de routage dynamique. Nous avons donc défini une architecture globale des réseaux optiques, ce qui nous a permis d'implémenter les algorithmes proposés. L'évaluation de performance de nos deux algorithmes nous a permis de les comparer aux algorithmes antérieurs. Nous en concluons alors qu'une bonne stratégie d'affectation

des longueurs d'onde aux requêtes entrantes permet d'améliorer les performances de nos deux algorithmes. De plus, les résultats obtenus par un des deux algorithmes proposés montrent que l'utilisation du nombre de longueurs d'onde libres sur les liaisons ne suffit pas à estimer la congestion sur un chemin. Par ailleurs, on remarque que le temps d'exécution de nos deux algorithmes peut être amélioré en utilisant des algorithmes auxiliaires plus adaptés. Enfin, nous constatons que le deuxième algorithme que nous proposons tient mieux la comparaison avec les algorithmes existants, et qu'il faut améliorer la fonction d'estimation de la congestion au niveau de notre premier algorithme.

## ABSTRACT

During last years, many applications multi-media required a high-speed rate of data transfer and a high reliability of the transmission channels. In this situation, the optical networks based on WDM technology represent by their great bandwidth an interesting alternative to the current communication networks. The management of these networks involved several problems among which the dynamic routing, which allows to convey data in the network by considering node and link state. Due to limited memory on the optical switches and constraints of high switching frequencies, the data transmitted in these networks must be routed without waiting queues along a path from source to destination. In optical networks deprived of wavelength converters, this problem becomes more complex since it's necessary to transfer the data between source and destination by using only one wavelength. For a whole of requests of connection generated dynamically for pairs source-destination ( $s-d$ ), it is thus necessary to determine the best path among  $k$  paths possible, by taking in account the constraints of cost and wavelength continuity. The optimization of this process will then make it possible to minimize the congestion in the network and consequently the rate of blocking of entering connections.

The purpose of this thesis is primarily the development of new dynamic routing algorithms to improve the performances obtained by former algorithms in this field. To achieve our goals, we started with a mathematical formulation of the problem above exposed. Then we proposed a new approach to estimate the congestion on a path as well as an additional criterion to analyze the links on a path. We then deduced two new dynamic routing algorithms. We thus defined a total architecture of the optical networks, which allowed us to implement the algorithms proposed. The performance analysis of our two algorithms enabled us to compare them with existing algorithms. We then conclude that a good wavelength assignment strategy for the entering requests allows us to improve our algorithms' performances. Moreover, the results obtained by our first algorithm show that the utilization of the number of free wavelengths on links is not

enough to estimate the congestion on a path. In addition, we notice that the execution time of our two algorithms can be improved by using more adapted auxiliary algorithms. Lastly, we see that the second algorithm that we propose holds better the comparison with the existing algorithms and that it's necessary to improve the estimation function of the congestion on our first algorithm's level.

## 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>viii</b>
<b>TABLE DES MATIÈRES.....</b>	<b>x</b>
<b>LISTE DES FIGURES.....</b>	<b>xiii</b>
<b>LISTE DES SIGLES ET ABRÉVIATIONS .....</b>	<b>xv</b>
<b>CHAPITRE 1 : INTRODUCTION.....</b>	<b>1</b>
1.1    Définitions et concepts de base .....	1
1.2    Éléments de la problématique .....	3
1.3    Objectifs de recherche.....	5
1.4    Approche méthodologique .....	5
1.5    Plan du mémoire .....	6
<b>CHAPITRE 2 : TRANSMISSION DE DONNÉES DANS LES RÉSEAUX OPTIQUES .....</b>	<b>7</b>
2.1    Les réseaux optiques .....	7
2.1.1    Architecture générale .....	7
2.1.2    Transmission de données dans les réseaux optiques.....	9
2.2    Routage et affectation de longueurs d'onde.....	11
2.2.1    Routage optique .....	12
2.2.1.1    L'algorithme du plus court chemin.....	14
2.2.1.2    L'algorithme du chemin le moins congestionné: « Fixed-path Least Congested » .....	15
2.2.1.3    L'algorithme du chemin le moins chargé : « Least Loaded Routing »	18
2.2.1.4    Algorithme heuristique.....	20
2.2.2    Affectation de longueurs d'onde .....	22
2.3    Gestion des fautes et restauration de trafic .....	23
2.4    Modélisation du taux de blocage.....	27
<b>CHAPITRE 3 : ALGORITHMES DE ROUTAGE OPTIQUE .....</b>	<b>30</b>

3.1	Architecture de réseau optique considérée.....	30
3.2	Formulation du problème de routage dynamique optique .....	37
3.3	Algorithmes de routage dynamique proposés .....	39
3.3.1	Estimation de la congestion : ECR.....	39
3.3.2	Méthode hybride : HFPLC- $k$ .....	42
3.4	Analyse de complexité.....	44
3.5	Restauration en cas de pannes.....	46
<b>CHAPITRE 4 : IMPLÉMENTATION ET ÉVALUATION DE PERFORMANCE .....</b>		<b>48</b>
4.1	Implémentation du système.....	48
4.1.1	Analyse globale de l'implémentation.....	48
4.1.2	Modèle de classes.....	52
4.1.3	Scénario d'implémentation des algorithmes de routage .....	57
4.2	Plan d'expériences .....	58
4.2.1	Facteurs primaires .....	59
4.2.2	Indices de performance .....	60
4.3	Résultats .....	60
4.3.1	Réseau en anneau .....	60
4.3.1.1	Évaluation du taux de blocage .....	61
4.3.1.2	Évaluation du temps de réponse.....	68
4.3.1.3	Évaluation du coût total .....	69
4.3.2	Réseau maillé .....	72
4.3.2.1	Évaluation du taux de blocage .....	72
4.3.3	Réseau ARPANET.....	75
4.3.3.1	Évaluation du taux de blocage .....	75
4.3.3.2	Évaluation du temps de réponse.....	78
4.3.3.3	Évaluation du coût total .....	79
<b>CHAPITRE 5 : CONCLUSION .....</b>		<b>80</b>
5.1	Synthèse des travaux .....	80
5.2	Limitations des travaux .....	82

5.3	Indications de recherches futures .....	83
<b>BIBLIOGRAPHIE .....</b>		<b>85</b>
<b>ANNEXES .....</b>		<b>88</b>

## LISTE DES FIGURES

Figure 1.1 Exemple de réseau optique avec 6 routeurs.....	3
Figure 2.1 Topologie d'un réseau optique .....	9
Figure 2.2 Chemins optiques dans un réseau optique.....	11
Figure 2.3 Exemple illustratif de réseau optique .....	16
Figure 2.4 Réseau ARPA-2 des États-Unis.....	22
Figure 2.5 Principe de la protection 1+1 .....	25
Figure 2.6 Restauration de liaison pour une panne d'élément.....	26
Figure 3.1 Architecture du système de résolution.....	32
Figure 3.2 Algorithme de Yen modifié pour les K plus courts chemins.....	33
Figure 3.3 Forme simple de l'algorithme de Dijkstra.....	34
Figure 3.4 Algorithme du ECR .....	40
Figure 3.5 Réseau optique.....	41
Figure 3.6 Algorithme du HFPLC-k .....	43
Figure 3.7 Traitement en cas de bris de liaison.....	47
Figure 4.1 Fichier de représentation topologique d'un réseau en anneau.....	49
Figure 4.2 Exemple de fichier de génération de trafic .....	51
Figure 4.3 Diagramme de classes du système.....	53
Figure 4.4 Topologie du réseau simulé pour ARPANET .....	54
Figure 4.5 Contenu des tables de routage pour le plus court chemin (K=0).....	55
Figure 4.6 Diagramme de séquence du traitement d'une requête .....	59
Figure 4.7 Topologie du réseau optique en anneau simulé.....	61
Figure 4.8 Évaluation du taux de blocage en fonction du trafic pour W=1 .....	63
Figure 4.9 Évaluation du taux de blocage en fonction du trafic pour W=3 et WA=First Fit.....	63
Figure 4.10 Évaluation du taux de blocage en fonction du trafic pour W=3 et WA=Most Used .....	64
Figure 4.11 Évaluation du taux de blocage en fonction du trafic	

pour $W=3$ et $WA=Least\ Used$ .....	64
Figure 4.12 Évaluation du taux de blocage en fonction du trafic	
pour $W=3$ et $WA=Random$ .....	65
Figure 4.13 Évaluation du taux de blocage en fonction du trafic	
pour $W=6$ et $WA=First\ Fit$ .....	66
Figure 4.14 Évaluation du taux de blocage en fonction du trafic	
pour $W=6$ et $WA=Most\ Used$ .....	67
Figure 4.15 Évaluation du taux de blocage en fonction du trafic	
pour $W=6$ et $WA=Least\ Used$ .....	67
Figure 4.16 Évaluation du taux de blocage en fonction du trafic	
pour $W=6$ et $WA=Random$ .....	68
Figure 4.17 Évaluation du temps de réponse en fonction du trafic pour $W=1$ .....	69
Figure 4.18 Évaluation du coût total des requêtes en fonction du trafic pour $W=1$ .....	70
Figure 4.19 Évaluation du coût total des requêtes en fonction du trafic	
pour $W=3$ et $WA=First\ Fit$ .....	71
Figure 4.20 Évaluation du coût total des requêtes en fonction du trafic	
pour $W=3$ et $WA=Most\ Used$ .....	71
Figure 4.21 Topologie du réseau optique maillé.....	72
Figure 4.22 Évaluation du taux de blocage en fonction du trafic pour $W=1$ .....	73
Figure 4.23 Évaluation du taux de blocage en fonction du trafic	
pour $W=3$ et $WA=First\ Fit$ .....	74
Figure 4.24 Évaluation du taux de blocage en fonction du trafic	
pour $W=6$ et $WA=First\ Fit$ .....	74
Figure 4.25 Évaluation du taux de blocage en fonction du trafic pour $W=1$ .....	76
Figure 4.26 Évaluation du taux de blocage en fonction du trafic	
pour $W=3$ et $WA=First\ Fit$ .....	77
Figure 4.27 Évaluation du taux de blocage en fonction du trafic	
pour $W=3$ et $WA=Least\ Used$ .....	77
Figure 4.28 Évaluation du temps de réponse en fonction du trafic pour $W=1$ .....	78

## LISTE DES SIGLES ET ABRÉVIATIONS

- CDM** : Multiplexage par division de code (Code Division Multiplexing)
- ECR** : Routage par estimation de la congestion (Estimated Congestion Routing)
- FPLC** : Chemin le moins congestionné (Fixed-Path Least Congested)
- HFPLC** : Chemin hybride le moins congestionné (Hybrid Fixed-Path Least Congested)
- LLR** : Chemin le moins chargé (Least Loaded Routing)
- PCC** : Plus Court Chemin
- TDM** : Multiplexage par division temporelle (Time Division Multiplexing)
- TWDM** : Multiplexage mixte par division en temps et en longueur d'onde (Time Wavelength Division Multiplexing)
- WAN** : Réseau de grande étendue (Wide Area Network)
- WDM** : Multiplexage par division de longueurs d'onde (Wavelength Division Multiplexing)
- WRS** : Commutateur optique de longueurs d'onde (Wavelength Routing Switch)
- WSXC** : Commutateur optique à sélection de longueurs d'onde (Wavelength-Selective Cross-Connect)
- WIXC** : Commutateur optique à longueurs d'onde interchangeables (Wavelength-Interchange Cross-Connect)

## CHAPITRE 1

### INTRODUCTION

Au cours des dernières décennies, les réseaux de communication ont vu leur charge de transport augmenter de manière drastique. Les applications comme l'Internet, l'imagerie médicale et la vidéoconférence exigent de grandes largeurs de bande passante et obligent les entreprises à trouver de nouveaux supports de transmission plus rapides et mieux adaptés aux nouvelles technologies. Dans ce contexte, la fibre optique est venue révolutionner les réseaux dès son utilisation dans les années 1980. L'émergence de la technologie optique dans les réseaux de transmission de données entraîne d'importantes possibilités en largeurs de bande et une réduction des coûts pour les grandes applications. Mais malgré l'atout essentiel que représente la vitesse des lasers optiques, plusieurs problèmes qui ne se posaient pas dans les réseaux de communication habituels apparaissent. La planification et la conception des réseaux optiques soulèvent donc beaucoup d'interrogations. Parmi ces problèmes, figure celui du routage dynamique dans les réseaux optiques, objet du présent mémoire. Dans ce chapitre d'introduction, nous présenterons d'abord quelques définitions et concepts de base suivis des éléments de la problématique. Ensuite, nous préciserons nos objectifs de recherche et notre approche méthodologique pour les réaliser, puis nous finirons par une esquisse des grandes lignes du mémoire.

#### 1.1 Définitions et concepts de base

Un *réseau optique* est un réseau de télécommunications à grande capacité basé sur des équipements optiques qui utilisent les ondes électromagnétiques pour transmettre les signaux sous forme de longueur d'onde lumineuse. Le canal de transmission utilisé n'est plus le câble de cuivre traditionnel, mais plutôt la *fibre optique* qui possède de meilleures caractéristiques de transmission. Dans un tel réseau, l'ensemble des opérations effectuées pour diriger les trafics entrants d'une source vers une destination

en respectant les caractéristiques du trafic et les restrictions liées au réseau est appelé *routage*. Le choix du routage lors de la planification des réseaux à large bande constitue une étape très importante car elle permet de satisfaire les échanges de données entre usagers en tenant compte de la nature du réseau.

Différents algorithmes de routage ont été développés pour divers types de réseaux, mais on retrouve à la base certaines fonctionnalités communes à toute méthode de routage. L'information sur la topologie du réseau et ses restrictions est connue, ainsi que l'état du trafic usager entrant (localisation géographique des usagers, services requis par usager, etc.). Par ailleurs, toutes les techniques de routage génèrent un ensemble de chemins possibles entre une source et une destination pour choisir éventuellement le meilleur chemin dans cet ensemble, en considérant parfois l'état du réseau et les exigences de ses usagers. Les routages qui tiennent compte de l'état courant du réseau pour déterminer les chemins sont dits *dynamiques*, et ceux qui établissent à l'avance les chemins sont qualifiés de *statiques*.

Dans un réseau optique, les chemins choisis peuvent servir à transmettre les données sur le même canal représenté ici par une longueur d'onde ou en changeant de longueur d'onde selon leur disponibilité aux nœuds intermédiaires. Le changement de longueurs d'onde se fait grâce aux convertisseurs de longueurs d'onde qui sont très dispendieux. Les chemins pour lesquels on n'observe aucune conversion optique sont appelés *chemins optiques* et présentent la particularité de n'utiliser qu'une seule longueur d'onde.

Dans les réseaux optiques comme dans la plupart des réseaux de communication, les usagers sont exposés à différentes pannes du système qui peuvent entraîner des pertes financières extrêmement préjudiciables pour eux. Pour remédier à ce problème, il importe donc, dans l'administration des réseaux, de se prémunir d'un dispositif rapide et efficace pour répondre dans les plus brefs délais aux bris qui peuvent survenir. Les réseaux optiques qui sont munis d'un tel dispositif sont capables de survivre à une défaillance quelconque grâce à certaines méthodes de restauration des communications.

La protection et la restauration des pannes constituent des facteurs importants à considérer lors de la planification des réseaux optiques.

## 1.2 Éléments de la problématique

Lors de l'étape de routage des données, les plus courts chemins entre chaque paire de nœuds sont déterminés et on cherche à transmettre le flot de données qui arrivent avec un certain débit. Prenons par exemple le réseau présenté à la Figure 1.1. Chaque liaison est bidirectionnelle (simultanée) et dispose de deux longueurs d'onde  $\lambda_1$  et  $\lambda_2$ . Le coût de chaque liaison est déterminé par l'opérateur ou les gestionnaires du réseau et indiqué sur la Figure 1.1 par les chiffres placés au-dessus des liaisons.

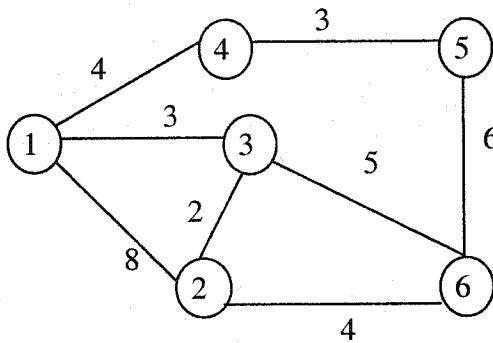


Figure 1.1 Exemple de réseau optique avec 6 routeurs

Supposons maintenant qu'à un temps  $t_0$ , il arrive simultanément au nœud 1 trois paquets différents pour les nœuds 2, 3 et 6. Si pour respecter uniquement le critère du plus court chemin, le nœud 1 décidait d'envoyer tous les paquets par le nœud 3, il y aurait une congestion au niveau de la liaison (1,3). Le dernier paquet ne pourra être envoyé sur le réseau avant que l'une des deux longueurs d'onde de la liaison (1,3) ne soit libérée. Il y aura donc blocage de ce paquet puisque, contrairement à ce qui se fait dans les réseaux de communications traditionnels, les routeurs optiques n'ont pas la propriété de stocker temporairement les données en attendant leur transmission. Pourtant, les chemins {1,2,6} ou {1,4,5,6} auraient bien pu utiliser une longueur d'onde disponible pour satisfaire la troisième requête. Pour n'importe quelle demande de trafic dans le réseau, il faut trouver la meilleure stratégie de routage pour transmettre les

données et satisfaire le maximum de requêtes possibles. La vitesse de traitement et la détermination des chemins d'envoi des données sont primordiales pour éviter la congestion du réseau et minimiser le taux de blocage. Considérons maintenant le cas d'un réseau réel, beaucoup plus grand que celui de la Figure 1.1, avec des milliers de requêtes qui arrivent à la seconde. Contrairement aux petits réseaux pour lesquels le problème peut se résoudre par l'exploration des solutions possibles, les calculs pour arriver à des solutions optimales deviennent plus complexes et plus coûteux en ressources et en temps. Le problème consiste alors à trouver la meilleure stratégie d'affectation des liaisons et de leurs longueurs d'onde pour la transmission des données qui arrivent aux nœuds, en réduisant leur temps de propagation et en minimisant les blocages de paquets. Il s'agit du problème conjoint de routage et d'affectation de longueurs d'onde dans les réseaux optiques. Aucune méthode de résolution mathématique n'a pu trouver la solution optimale à ce problème NP-complet.

Mokhtar et Azizoglu (1996, 1998) ont proposé une heuristique avec quatre méthodes d'affectation de longueurs d'onde différentes. Les performances globales sont satisfaisantes mais faute d'avoir comparé leur algorithme à d'autres recherches dans le domaine, il demeure difficile de se faire une idée nette de leurs travaux. D'autres chercheurs ont pour leur part réduit la complexité du problème en séparant le routage et l'affectation des longueurs d'onde. Ainsi, dans le cadre du routage optique, plusieurs heuristiques et autres méthodes classiques ont été étudiées. Li et Soman (1999) ont développé une heuristique appelée FPLC qui se base sur le contrôle de la congestion des liaisons entre nœuds voisins pour minimiser le taux de blocage des requêtes. Cette méthode se base sur la contrainte de continuité de longueur d'onde pour chaque chemin établi, ce qui la rend complexe et moins rapide à trouver des solutions proches de l'optimum. Par ailleurs, selon le nombre de chemins potentiels à analyser avant d'effectuer le meilleur choix, l'algorithme voit ses performances se dégrader. Une autre méthode tout aussi rapide mais moins complexe, le LLR, a été étudiée et développée par Zhou et Mouftah (2002). Le LLR vise essentiellement la minimisation du taux de blocage en trouvant des chemins alternatifs aux données chaque fois qu'une requête est

bloquée par la non-disponibilité du chemin initialement choisi. Une version dynamique a par ailleurs été proposée par les mêmes chercheurs et la comparaison avec le FPLC a montré que les deux méthodes avaient pratiquement les mêmes performances avec moins de complexité de calcul en faveur du LLR. Toutefois, aucune de ces deux méthodes ne donne la solution optimale au problème du routage.

### 1.3 Objectifs de recherche

Ce mémoire a pour objectif principal l'élaboration de nouveaux algorithmes de recherche de chemins dynamiques pour résoudre le problème du routage optique tel que décrit précédemment. Le taux de blocage obtenu lors de l'exécution de ces algorithmes doit être très faible et leur niveau de complexité au moins comparable à celui des méthodes traditionnellement proposées. Plus spécifiquement, ce mémoire vise :

- l'élaboration d'un modèle de réseau optique standard pour la simulation des différentes techniques de routage et d'affectation de longueurs d'onde ;
- la conception et l'implémentation de deux algorithmes de routage dynamique dans un réseau optique en fonction de la charge du réseau ;
- l'évaluation de performance de ces algorithmes ainsi que leur comparaison avec les meilleurs algorithmes existant dans la littérature.

Les méthodes proposées devront convenablement soutenir la confrontation avec les méthodes actuelles en terme de rapidité de calcul et de satisfaction de trafic.

### 1.4 Approche méthodologique

Pour atteindre nos objectifs, nous commencerons par développer un modèle mathématique du problème avec la fonction objectif à minimiser, les contraintes à satisfaire et les différents paramètres qui interagissent avec cette fonction. Une architecture de réseaux optiques basée sur la modularité des classes sera implémentée pour établir les bases de l'implémentation des deux méthodes que nous présenterons. Ensuite, nous étudierons en détail les techniques proposées actuellement, pour identifier leurs forces et faiblesses. Puis, nous identifierons des facteurs d'amélioration pour

lesdites techniques et nous proposerons deux nouvelles méthodes capables de remédier aux faiblesses de ces dernières. Et enfin, nous améliorerons cette approche pour satisfaire différents critères de performance : la complexité et le temps d'exécution des algorithmes, le taux de blocage des requêtes et l'aptitude du réseau à gérer efficacement les défaillances.

### **1.5 Plan du mémoire**

Ce mémoire est rédigé en cinq chapitres. Après ce chapitre d'introduction, le chapitre 2 présente une vue générale des réseaux optiques et un sommaire des méthodes existantes pour la résolution du problème de routage et d'affectation de longueurs d'onde. Le chapitre 3 décrit en détail les deux méthodes que nous proposons avec les algorithmes qui s'y rattachent. Ensuite, le chapitre 4 présente l'implémentation de notre système ainsi que l'évaluation des performances de nos algorithmes de routage dynamique, selon certains critères spécifiques. Enfin, le chapitre 5 conclut ce mémoire avec une synthèse des travaux et des résultats obtenus, les limites de notre travail et les améliorations futures à y apporter.

## CHAPITRE 2

### TRANSMISSION DE DONNÉES DANS LES RÉSEAUX OPTIQUES

Les *réseaux optiques* acheminent les communications sous la forme d'ondes lumineuses qui circulent à travers des *fibres optiques*. De par leur architecture, ces réseaux doivent observer certaines règles et contraintes pour réaliser les transferts de données. Certaines opérations comme le choix d'un chemin ou d'une longueur d'onde à affecter à une transmission obéissent à un processus de prise de décision. Tous ces problèmes ont et continuent de faire l'objet de plusieurs travaux de recherche. Nous traitons ici celui du routage dynamique pour lequel nous ferons une revue des travaux déjà réalisés. Nous commencerons par une description générale de l'architecture des réseaux optiques et nous poursuivrons par une présentation des types de routage utilisés et des différents algorithmes d'affectation de longueurs d'onde. Ensuite, nous ferons une synthèse des différentes techniques utilisées pour assurer la sauvegarde des données d'usager lors de l'occurrence de pannes dans le réseau. Et enfin, nous ferons un bref rappel sur la modélisation du taux de blocage dans les réseaux optiques.

#### 2.1 Les réseaux optiques

Nous illustrons dans cette partie l'ensemble des caractéristiques des réseaux optiques ainsi que leurs mécanismes de transmission de données.

##### 2.1.1 Architecture générale

L'objectif principal d'un réseau de communications est de fournir une bonne connectivité entre les paires de nœuds qui échangent des messages. Les réseaux optiques sont classés dans la catégorie des réseaux à routage. Dans ceux-ci, le signal provenant d'un nœud est transmis sélectivement à travers une partie du réseau vers un nœud destination, contrairement aux réseaux à diffusion où le message est diffusé à tous les nœuds. Les réseaux étendus (WAN : *Wide Area Network*) font également partie des

réseaux nécessitant des algorithmes de routage car les nœuds sont interconnectés par des liaisons point à point ou des fibres optiques. Les modèles d'interconnexion utilisés pour les fibres optiques se répartissent en trois types :

- étoile passive ;
- routeur passif ;
- commutateur actif.

L'*étoile passive* est basée sur le principe du "*multicast*". Un signal entrant verra sa puissance divisée également entre toutes les sorties. Toutes les machines du réseau reçoivent donc les messages lancés par n'importe quelle machine du réseau. Ainsi, l'étoile passive est employée pour établir les réseaux locaux ou les réseaux de diffusion.

Un *routeur passif* permet d'acheminer un message d'un nœud source vers un nœud destination en utilisant une matrice de routage propre au routeur. Cette matrice de routage est statique et ne peut être modifiée au cours du temps ou en fonction de l'évolution du réseau.

Un *commutateur actif* peut se comparer au routeur passif sauf que sa matrice de routage peut être modifiée à travers des commandes électroniques en fonction des besoins du réseau. Cependant, cela induit un contrôle additionnel, contrairement à l'étoile passive et au routeur passif. Un commutateur actif est également appelé commutateur/routeur de longueur d'onde (*WRS* « *wavelength routing switch* ») ou interconnecteur (*cross-connect*). Les commutateurs actifs sont essentiellement utilisés dans les réseaux métropolitains et dans les réseaux étendus.

Les réseaux que nous considérons dans ce mémoire sont des réseaux de transport étendus et leur topologie, illustrée à la Figure 2.1, se compose donc :

- des routeurs optiques ;
- des fibres optiques ;
- et des ondes lumineuses désignées par  $\lambda$  qui représentent les canaux de communication sur les fibres optiques.

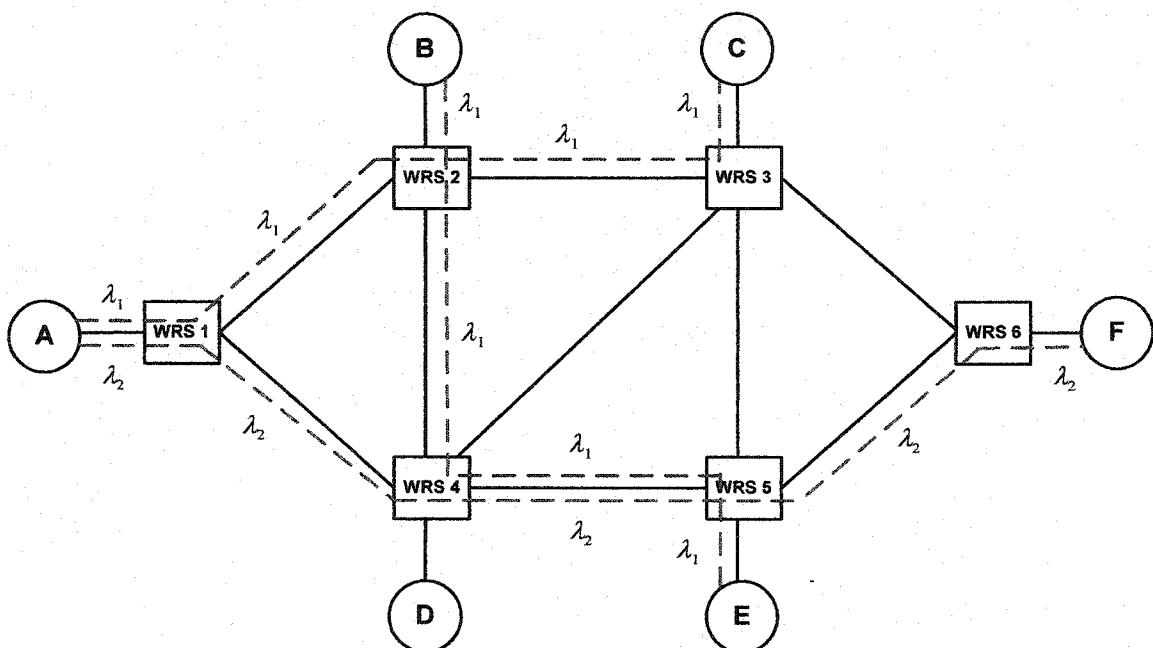


Figure 2.1 Topologie d'un réseau optique

### 2.1.2 Transmission de données dans les réseaux optiques

Quelques techniques sont proposées pour effectuer la transmission de données dans les réseaux optiques :

- le *multiplexage par division temporelle* (TDM « *Time Division Multiplexing* ») ;
- le *multiplexage par division de longueur d'onde* (WDM « *Wavelength Division Multiplexing* ») ;
- le *multiplexage mixte par division en temps et en longueur d'onde* (TWDM « *Time Wavelength Division Multiplexing* ») ;
- et le *multiplexage par division de code* (CDM « *Code Division Multiplexing* »).

Le multiplexage par division temporelle est une technique qui consiste à entrelacer des unités d'informations, ici des bits, prélevées de plusieurs sources et à transmettre la séquence formée sur une liaison de fibre optique.

Le multiplexage par division de longueur d'onde utilise quant à lui une technique dans laquelle la lumière est modulée selon plusieurs longueurs d'onde différentes, chacune évoluant à une vitesse propre aux composants électroniques. Puisque la technique la plus courante et la mieux implantée est celle de la WDM, nous ne nous

intéresserons dans la suite de ce mémoire qu'aux réseaux optiques basés sur cette technologie. Les autres techniques citées plus haut, à savoir le CDM et le TWDM, sont futuristes et présentent encore quelques inconvénients qui compromettent leur essor auprès des entreprises.

Les réseaux optiques construits selon la technique WDM peuvent être de très grande taille et de différentes topologies. Les connecteurs employés au niveau des nœuds du réseau sont des commutateurs actifs : des inter-connecteurs sélectifs de longueur d'onde (WSXC : *wavelength-selective cross-connect*) ou des inter-connecteurs de longueurs d'onde interchangeables (WIXC : *wavelength-interchange cross-connect*). Les routeurs associés aux premiers sont capables de faire le routage d'un signal optique d'une fibre à une autre en conservant la longueur d'onde du signal, alors que les routeurs utilisant les WIXCs sont capables de changer la longueur d'onde d'un signal optique d'entrée en utilisant des convertisseurs de longueurs d'onde qui transmettent en sortie sur une fibre optique ce signal selon une nouvelle longueur d'onde. Les réseaux optiques utilisant les WSXCs sont plus répandus car le coût des convertisseurs de longueurs d'onde est exorbitant. De plus, on peut transmettre des signaux optiques d'un nœud à un autre nœud de ces réseaux en établissant des *chemins optiques (lightpaths)* caractérisés par la même longueur d'onde des signaux et ne nécessitant aucune conversion optoélectronique ou aucune mémorisation au niveau des nœuds intermédiaires. Un exemple de réseau optique avec des chemins optiques de différentes longueurs d'onde  $\lambda_i$  est donné à la Figure 2.2. Les nœuds ronds représentent des routeurs alors que les carrés sont des connecteurs optiques. À cause des contraintes liées à la technologie des fibres optiques, le nombre de signaux lumineux de longueurs d'onde différentes que peut supporter chaque fibre est limité. Pour compenser ces limites et répondre à de gros trafics, certains réseaux optiques mixtes utilisent essentiellement des WSXCs avec des WIXCs à certains nœuds stratégiques. De plus, la majorité des liaisons utilisées dans ces réseaux comportent chacune plusieurs fibres, ce qui augmente la possibilité pour ces réseaux de supporter plus de trafic.

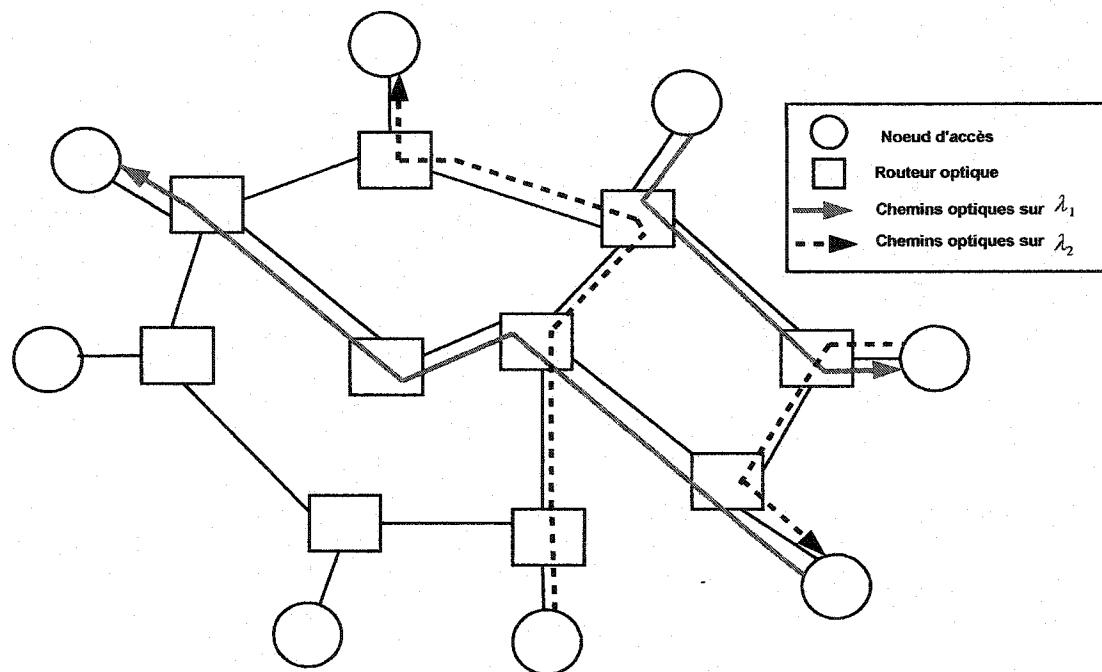


Figure 2.2 Chemins optiques dans un réseau optique

## 2.2 Routage et affectation de longueurs d'onde

Dans un réseau optique WDM comportant  $N$  noeuds, il y aurait un canal de communication dédié entre chaque paire de noeuds si chaque noeud était équipé de  $(N-1)$  émetteurs et  $(N-1)$  récepteurs. Le contrôle requis sur le réseau serait donc minimal, mais il faut convenir que ce type de réseau ne conviendrait pas à des réseaux de très grande taille et que les ajouts de nouveaux noeuds au réseau seraient fastidieux car il faudrait  $\frac{N * (N - 1)}{2}$  liaisons. De plus, les émetteurs et récepteurs étant coûteux, une telle

structure de réseau s'avère totalement inadéquate. Ainsi, il faut penser à un autre dispositif capable d'acheminer l'information entre les différentes paires de noeuds tout en minimisant l'utilisation quantitative des ressources du réseau. Ce dispositif requiert les caractéristiques suivantes :

- il faut avoir un ensemble de chemins optiques qui peuvent être établis sur le réseau ;

- on doit considérer les contraintes sur le nombre de longueurs d'onde disponibles (éventuellement celle sur la *continuité des longueurs d'onde* dans un réseau sans convertisseur optoélectronique) pour déterminer les chemins optiques à installer ;
- il faut identifier les meilleures longueurs d'onde à affecter aux chemins optiques.

Un tel problème d'optimisation des réseaux optiques est qualifié dans la littérature scientifique de problème *de routage et d'affectation de longueurs d'onde* (RWA : *Routing and Wavelength Assignment*). Sa résolution est complexe et varie selon le type et la charge du réseau étudié. Cependant, plusieurs méthodes de routage fixe ou semi-adaptatif ainsi que différentes manières d'affecter les longueurs d'onde ont été développées et évaluées expérimentalement. Pris séparément, le problème de routage et celui de l'affectation des longueurs d'onde sont moins complexes à résoudre mais on perd en optimisation par rapport à une résolution conjointe des deux problèmes. Dans notre cas, nous considérons ces problèmes de manière disjointe.

### 2.2.1 Routage optique

De manière conventionnelle, le routage se définit comme le choix de la meilleure voie à utiliser pour acheminer de l'information d'une source à une destination. Dans les réseaux optiques, il représente l'ensemble des opérations par lesquelles un ensemble de nœuds et liaisons est choisi pour transmettre des données d'un nœud source vers un nœud destination. Ces opérations peuvent se faire de manière *statique* ou *dynamique* selon les paramètres courants du réseau. Il existe deux formes de routage selon la finalité d'une requête de transmission :

- le routage *fixe* (*fixed routing*) qui essaie de trouver une route prédéterminée pour une paire source – destination et, si la route n'est pas disponible, la requête est bloquée ;

- le routage *alterné* (*fixed alternate routing*) tente de choisir parmi un ensemble de routes prédéterminées pour chaque paire source – destination une route disponible. La requête a donc moins de probabilité d'être bloquée, quel que soit l'algorithme de routage associé.

Une grande différence existe entre les deux formes de routage en terme de trafic rejeté par le réseau qui les utilise. Jue et Gaoxi (2000) prouvent dans leur article que le taux de blocage engendré par l'algorithme du plus court chemin (routage statique) augmente de manière presque exponentielle en fonction de la charge du réseau. Par contre, le taux de blocage du même réseau lorsque ce dernier utilise l'algorithme du chemin le moins congestionné (routage alterné) reste bas à mesure que le trafic augmente.

Dans les routages *statiques*, la topologie du réseau est connue et chaque nœud détient une table de routage fixe sur les meilleurs chemins à emprunter pour envoyer de l'information vers chacun des autres nœuds du réseau. Les tables de routage sont périodiquement mises à jour pour assurer la fiabilité du réseau et sa survie en cas de panne ponctuelle d'un des composants du réseau. Les stratégies de routage statique proposées sont peu complexes mais elles ont un fort taux de blocage des requêtes dans les réseaux optiques selon l'état de congestion de certaines liaisons du réseau. Les ressources du réseau ne sont pas utilisées de manière optimale et il peut y avoir des nœuds qui supportent beaucoup plus de trafic que d'autres.

Les algorithmes de routage *dynamique* (*adaptive routing*) présentent moins d'inconvénients que les stratégies de routage statique. La recherche des chemins s'effectue en fonction de plusieurs critères dont la charge du réseau, le contrôle de congestion au niveau de chaque nœud et la disponibilité des liaisons du réseau. L'information de contrôle utilisée peut être globale pour le réseau ou locale, c'est-à-dire entre nœuds voisins. Dans le contrôle global, chaque nœud connaît l'ensemble des liaisons actives ainsi que les longueurs d'onde utilisées sur ces liaisons. En fonction de ces informations, le nœud source peut alors affecter, suivant une métrique donnée, un poids à chaque liaison et calculer le meilleur chemin entre lui et le nœud destination.

Cependant, une telle méthode peut surcharger le réseau à cause de toutes les informations de contrôle qui doivent y circuler. Avec le contrôle local, chaque nœud ne maintient des informations que sur ses liaisons incidentes. Ainsi, la recherche d'un chemin optique se fait de manière répartie. Le nœud source cherche des liaisons disponibles chez ces voisins et ces derniers font de même par propagation jusqu'à la destination. Il y a donc moins de trafic additionnel lié au contrôle dans le réseau.

Nous allons présenter quelques algorithmes pour illustrer ces différentes méthodes de routage.

### 2.2.1.1 L'algorithme du plus court chemin

Fondé sur la théorie des graphes et des réseaux, cet algorithme utilise les travaux de Dijkstra et de Bellmann-Ford pour trouver le chemin le plus court en terme de sauts entre une source et une destination dans le réseau. La charge du réseau et la congestion au niveau des nœuds évoluant en temps réel, cet algorithme est appliqué en établissant à chaque requête de transmission un chemin qui minimise le nombre de sauts entre la source et la destination. Si plusieurs chemins de longueur identique sont retrouvés, d'autres critères de sélection leur sont appliqués ou on en choisit un aléatoirement.

Il existe une version améliorée de l'algorithme du plus court chemin : l'algorithme du plus court chemin avec contraintes (*CSPF : Constrained Shortest Path First Algorithm*), étudié par Poppe et al. (2001), qui prend en compte certaines contraintes comme l'état des liaisons ou le délai maximal de propagation, etc. En général, cet algorithme se décompose en deux parties :

- étape 1 : éliminer de la topologie logique du réseau toutes les liaisons qui ne respectent pas les contraintes imposées ;
- étape 2 : déterminer un plus court chemin entre la source et la destination dans le réseau ainsi obtenu en fonction des métriques choisies.

Cet algorithme présente l'avantage de la simplicité et de la rapidité de traitement des calculs.

### 2.2.1.2 L'algorithme du chemin le moins congestionné: « Fixed-path Least Congested »

Cet algorithme utilise la contrainte de continuité des longueurs d'onde aux nœuds ainsi que le nombre de chemins disponibles pour évaluer la congestion du réseau. Par définition, plus une liaison possède des longueurs d'onde libres, moins elle est congestionnée. Le degré de congestion d'une liaison exprime le nombre de canaux libres qu'elle possède. Périodiquement, un ensemble de chemins possibles entre chaque paire de nœuds source-destination est calculé et stocké au niveau de chaque nœud source. Dès qu'une requête de transmission est reçue, le nœud source cherche les chemins praticables en envoyant vers la destination des paquets de petite taille dans le réseau. Le chemin ayant le plus de longueurs d'ondes disponibles sur les liaisons entre la source et la destination est alors choisi. Si aucun chemin n'est trouvé ou si la contrainte de continuité des longueurs d'onde n'est pas respectée, la requête est bloquée. Cette méthode a été implémentée et les résultats des simulations montrent qu'elle a de meilleures performances que l'algorithme du plus court chemin. Cependant, le temps de calcul initial des chemins entre toutes les paires de nœuds et les informations de contrôle nécessaires à l'algorithme diminuent l'efficacité et les performances de ce dernier.

#### Algorithme :

Soient les variables suivantes :

$S_p$  : l'ensemble des chemins utilisant le nœud  $p$  ;

$x_k$  : le nombre de longueurs d'onde libres sur la fibre optique  $k$  ;

$\min_L(x_k)$  : le nombre de longueurs d'onde libres sur la liaison  $k$  la plus congestionnée, c'est-à-dire le nombre minimum de longueurs d'onde libres sur toutes les liaisons du chemin  $L$ , avec  $k \in L$ .

Pour tous les chemins admissibles entre la source et la destination, il existe deux approches possibles :

*Approche 1 : maximiser  $S_p [\min_L(x_k)]$*

- Choisir le chemin ayant le nombre maximum de longueurs d'onde libres communes pour toutes ses liaisons.
- En cas d'égalité entre plusieurs chemins, choisir celui qui a le moins de sauts entre la source et la destination.

### Approche 2 : chemin le plus court

- Choisir le chemin le plus court ayant le plus grand  $\min_L(x_k)$  sur sa liaison la plus congestionnée.
- Si ce critère ne suffit pas, utiliser le plus grand  $\min_L(x_k)$  sur la deuxième liaison la plus congestionnée de ces chemins pour faire le choix final.

Le réseau présenté à la Figure 2.3 nous permettra d'illustrer les deux approches :

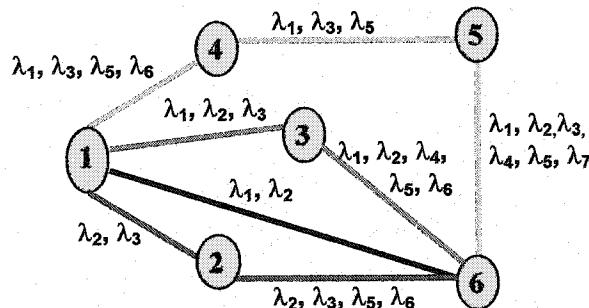


Figure 2.3 Exemple illustratif de réseau optique

Les lettres se trouvant sur les liaisons du graphe représentent les longueurs d'onde disponibles sur ces liaisons au moment de l'arrivée d'une requête. On désire effectuer une transmission entre la paire de nœuds 1 et 6. Les chemins admissibles dans ce cas sont au nombre de quatre :

- *Approche 1 :*

Chemin A {1, 6} :  $\{\lambda_1, \lambda_2\} \Rightarrow \min_A(x_k) = 2$

Chemin B {1, 3, 6} :  $\{\lambda_1, \lambda_2, \lambda_3\} \cap \{\lambda_1, \lambda_2, \lambda_4, \lambda_5, \lambda_6\} = \{\lambda_1, \lambda_2\} \Rightarrow \min_B(x_k) = 2$

Chemin C {1, 2, 6} :  $\{\lambda_2, \lambda_3\} \cap \{\lambda_2, \lambda_3, \lambda_5, \lambda_6\} = \{\lambda_2, \lambda_3\} \Rightarrow \min_C(x_k) = 2$

Chemin D {1, 4, 5, 6} :  $\{\lambda_1, \lambda_3, \lambda_5, \lambda_6\} \cap \{\lambda_1, \lambda_3, \lambda_5\} \cap \{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_7\} =$

$$\{\lambda_1, \lambda_3, \lambda_5\} \Rightarrow \min_D(x_k) = 3 \text{ avec } (\lambda_1, \lambda_3 \text{ et } \lambda_5)$$

Résultat : Le chemin D est choisi car il a le plus grand  $\min_L(x_k)$  avec trois sauts.

- *Approche 2 :*

Chemin A {1, 6} : 1 hop

Chemin B {1, 3, 6} : 2 hops

Chemin C {1, 2, 6} : 2 hops

Chemin D {1, 4, 5, 6} : 3 hops

Résultat : Le chemin A est choisi car il est le chemin le plus court ayant le nombre maximum de longueurs d'onde sur sa seule liaison.

Les deux approches ont été évaluées dans un réseau WDM symétrique représentant un graphe complet de  $N$  noeuds et il apparaît que la deuxième approche produit les taux de blocage de requêtes les plus faibles. La première approche peut établir des connexions sur des chemins alternatifs alors que des chemins plus directs avec moins de longueurs d'onde disponibles peuvent mieux convenir. Ceci entraîne une augmentation du trafic sur certaines liaisons principales et peut donc devenir un goulot d'étranglement pour les blocages de requêtes.

Li et Somani (1999) ont développé un modèle analytique pour évaluer la probabilité de blocage d'une requête par la première approche du FPLC, puis ils ont proposé un nouvel algorithme utilisant des données entre liaisons voisines pour améliorer les performances du FPLC original. Cette méthode qu'ils ont appelée FPLC- $N(k)$ <sup>1</sup> utilise aussi un ensemble de chemins initialement calculé et stocké à chaque noeud. Mais, lors de l'arrivée d'une requête, au lieu d'inspecter toutes les liaisons pour chaque chemin candidat, seules les longueurs d'onde disponibles sur les  $k$  premières liaisons de chaque chemin sont analysées. Si plusieurs longueurs d'onde sont disponibles sur le chemin choisi, la règle d'affectation<sup>2</sup> du premier trouvé ou *first fit* est appliquée et les auteurs ont prouvé qu'elle donne de meilleurs résultats que celle de l'aléatoire ou

<sup>(1)</sup> Par souci de simplification, nous utiliserons par la suite dans ce mémoire l'acronyme FPLC-k

<sup>(2)</sup> Les notions d'affectation de longueurs d'onde sont expliquées après en détail dans ce chapitre

*random*. Il est important de noter que l'efficacité du FPLC-N( $k$ ) repose sur le nombre  $k$  de liaisons considérées et ce nombre est étroitement lié au diamètre et à la topologie du réseau étudié. Li et Somani (1999) évaluent alors les performances de leur algorithme en considérant le taux de blocage des requêtes pour diverses valeurs de  $k$  attribuées à l'algorithme. Ils constatent qu'à partir de  $k=2$ , le taux de blocage diminue considérablement et que la différence avec  $k=3$  est peu sensible. Les tests menés à ce sujet ont prouvé qu'une valeur de  $k=2$  suffisait à garantir une bonne performance de l'algorithme dans un réseau bipartite 4x4 ou dans le réseau dorsal des États-Unis.

### 2.2.1.3 L'algorithme du chemin le moins chargé : « Least Loaded Routing »

Dans cet algorithme, le choix du routage est fait en fonction de la capacité disponible sur les liaisons les moins chargées en terme d'utilisation de leurs longueurs d'onde dans le réseau. Tous les chemins possibles entre la source et la destination sont examinés et le chemin le moins chargé, c'est-à-dire qui emprunte les liaisons les moins utilisées, est choisi pour équilibrer la charge du réseau. Cela limite aussi les congestions et sert à diminuer le nombre de blocages de requêtes de transmission dans le réseau. Cependant, le temps d'initialisation et les données de contrôle additionnelles rendent cet algorithme moins efficace. Désignons par  $H$  la longueur maximale d'un chemin faisant partie de l'ensemble des  $K$  plus courts chemins entre la source et la destination, et  $W$  le nombre de longueurs d'onde d'une fibre pour un réseau de  $N$  nœuds. La complexité d'exécution du *Least Loaded Routing* (LLR), selon Zhou et Mouftah (2002), est égale à  $O(KHW)$  dans le pire cas alors que celle du FPLC est de  $O(N^2HW^5)$ .

Si nous appliquons cette méthode au réseau de la Figure 2.3, nous avons :

Soit  $x_k$  le nombre minimal total de longueurs d'onde disponibles sur toutes les liaisons d'un chemin  $k$ .

Chemin A {1, 6} :  $x_k = 2$

Chemin B {1, 3, 6} :  $x_k = 3$

Chemin C {1, 2, 6} :  $x_k = 2$

Chemin D {1, 4, 5, 6} :  $x_k = 3$

Résultat : Le chemin B est choisi car il a le plus grand nombre de liaisons disponibles et il est le chemin avec le moins de sauts.

Une variante de cet algorithme est le LLR- $k$  dans lequel on examine  $k$  chemins dans l'ensemble des chemins admissibles au lieu de vérifier tous les chemins appartenant à cet ensemble. Zhou et Mouftah (2002) en ont proposé une version dynamique améliorée : le ALLR- $k$  ou « *Adaptive Least Load Routing-k* ». Dans leurs travaux, ils ont modifié les caractéristiques suivantes :

- au lieu de bloquer une requête quand le chemin le moins chargé choisi est pris,  $k$  chemins les moins chargés sont analysés pour satisfaire les requêtes. Si  $k=1$ , on revient au cas classique du LLR ;
- l'ensemble des chemins possibles est créé de manière semi-dynamique en fonction de l'historique du trafic dans le réseau alors que, pour le LLR classique, l'ensemble des chemins admissibles est fixe ;
- un protocole distribué de réservation de longueur d'onde est proposé également pour accélérer la procédure d'établissement des chemins optiques.

Les auteurs proposent qu'un ensemble initial de six chemins admissibles soit calculé pour chaque paire de nœuds source-destination, de manière périodique ou en cas d'ajout ou de panne d'un des composants du réseau. Ces six chemins sont répartis comme suit : le chemin le plus court, une paire de chemins les plus courts et disjoints, le chemin le plus court avec la plus grande largeur de bande et une paire de chemins les plus courts et disjoints avec la plus grande largeur de bande. Lorsque ces chemins sont obtenus par les algorithmes de Dijkstra et Suurballe (1984), les requêtes de transmission arrivant à chaque nœud sont traitées de manière à trouver au moins un chemin faisable. Contrairement à l'algorithme initial du chemin le moins chargé qui bloque la requête si le seul chemin trouvé par l'algorithme est occupé par une autre longueur d'onde, l'algorithme modifié de Zhou et Mouftah (2002) permet de sélectionner un chemin faisable parmi  $k$  chemins possibles. Ainsi, on a moins de blocage de requête au niveau des nœuds du réseau et le fait d'explorer plusieurs chemins possibles augmente la

chance de trouver le meilleur chemin. Les résultats de la simulation de cette méthode ainsi que de celles du LLR- $k$  et du FPLC sur des réseaux de 10 et 23 nœuds sont reportés dans Zhou et Mouftah (2002). Au cours de ces expériences, le FPLC et le LLR- $k$  utilisaient un ensemble de chemins admissibles préalablement calculés. Le FPLC offre de meilleures performances que le LLR- $k$  pour de faibles valeurs de  $k$  ( $k \in \{1,2\}$ ). Par contre, pour  $k=3$ , la différence est moins grande; mais dès qu'on dépasse ce nombre, la performance du LLR- $k$  se dégrade. De même, le ALLR-3 se comporte mieux que tous les LLR- $k$  car il utilise un ensemble de chemins admissibles semi-dynamique qui tient compte des changements de charge du réseau. De plus, le ALLR-3 est capable d'obtenir d'aussi bons résultats que le FPLC tout en utilisant moins de ressources du réseau et en demandant moins de complexité de calcul ( $O(kKHW)$  dans le pire cas). Les auteurs ont constaté qu'il était néanmoins difficile d'obtenir de meilleurs résultats lorsqu'on augmentait le nombre  $k$  de chemins examinés au-delà de 3.

#### 2.2.1.4 Algorithme heuristique

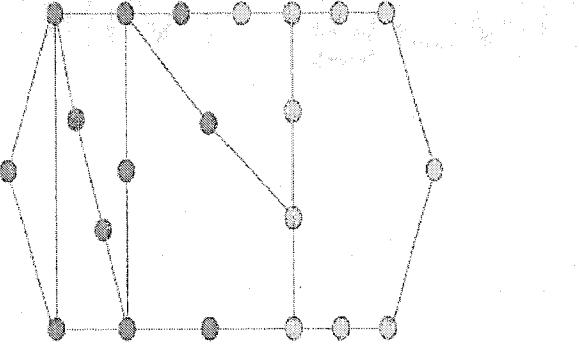
Tous les algorithmes présentés précédemment utilisaient un ensemble de chemins calculés d'avance selon les plus courts chemins pour effectuer une sélection lors de l'arrivée d'une requête. Mokhtar et Azizoglu (1996) ont décidé de modéliser et d'implémenter un algorithme fondé sur une recherche dynamique de chemin et de longueur d'onde sans ensemble initial. Ils ont alors proposé et évalué 4 méthodes conjointes de routage et d'affectation de longueur d'onde qui se définissent comme suit :

- la méthode « pack » qui essaie d'allouer la session à la longueur d'onde la plus utilisée parmi un ensemble décroissant des longueurs d'onde utilisées, dans le but de maximiser l'utilisation des longueurs d'onde disponibles dans le réseau ;
- la méthode « spread » qui essaie d'allouer la session à la longueur d'onde la moins utilisée parmi un ensemble croissant des longueurs d'onde utilisées dans le but de distribuer uniformément l'utilisation des longueurs d'onde disponibles dans le réseau ;

- la méthode « random » qui est une méthode aléatoire consistant à allouer la session à une longueur d'onde parmi un ensemble de longueurs d'onde choisies au hasard dans le réseau ;
- la méthode « exhaustive » qui considère l'ensemble des longueurs d'onde existant dans le réseau et trouve la longueur d'onde minimisant le nombre de sauts (hops) entre la source et la destination.

Les résultats de simulation de ces quatre méthodes sur le réseau ARPA-2 des États-Unis représenté à la Figure 2.4 sont donnés dans Mokhtar et al. (1996). On considère dans ce cas-ci que chaque liaison du réseau ARPA-2 comporte quatre longueurs d'onde. Les quatre méthodes ont été comparées à celle du routage fixe. On remarque que la méthode « pack » a le plus faible taux de blocage, suivie par les méthodes « random » et « spread ». La méthode « pack » maximise l'utilisation des longueurs d'onde disponibles au niveau de chaque liaison, alors que la méthode « spread » améliore la probabilité de trouver un chemin à chaque requête en répartissant la charge du réseau sur l'ensemble des longueurs d'onde. La méthode « random » arrive à équilibrer l'utilisation des longueurs d'onde puisque leur nombre est faible et donne ainsi de bonnes performances. La méthode « exhaustive » a naturellement été la moins performante car l'espace des solutions possibles augmente proportionnellement au nombre de longueurs d'onde existant sur chaque liaison du réseau. On note également que, lorsque ce nombre est assez grand, la méthode « pack » a de meilleurs résultats que toutes les autres méthodes.

Bien que n'ayant pas présenté des résultats de comparaison entre leurs méthodes et les algorithmes décrits auparavant, les auteurs ont affirmé que leurs méthodes avaient d'aussi bonnes performances que les algorithmes de routage dynamique présentés plus haut.



**Figure 2.4 Réseau ARPA-2 des États-Unis**

### 2.2.2 Affectation de longueurs d'onde

Dans un réseau tout optique, affecter une longueur d'onde à un chemin optique revient à trouver une longueur d'onde disponible sur chacune des fibres optiques constituant ce chemin. On peut viser au cours de cette affectation deux objectifs différents : soit maximiser la réutilisation (*maximum reuse*) en utilisant la plupart du temps les mêmes canaux, soit minimiser la réutilisation (*minimum reuse*) en essayant de distribuer équitablement la charge du réseau entre tous les canaux. Selon les simulations effectuées pour ces deux approches, la réutilisation maximale donne de meilleurs résultats dans les réseaux WDM (Stern, 2000). Il existe plusieurs méthodes pour affecter des longueurs d'onde aux chemins trouvés selon les différents objectifs poursuivis : l'aléatoire, la première trouvée, la moins utilisée, la plus utilisée, le *max-sum* et la capacité relative de perte.

La méthode aléatoire consiste à choisir au hasard une longueur d'onde parmi un ensemble de canaux disponibles pour un chemin.

La méthode de la première trouvée (*first fit*) permet au processus de choisir la première longueur d'onde disponible sur l'ensemble des liaisons du chemin. Cela peut occasionner des conflits lorsque la même longueur d'onde est affectée à deux chemins différents obtenus simultanément pour deux requêtes.

La méthode d'affectation de la longueur d'onde la moins utilisée (*least used*) a pour objectif de minimiser la réutilisation des canaux. Ainsi, lorsqu'un chemin est

trouvé pour une paire de nœuds source – destination, la longueur d'onde la moins utilisée parmi un ensemble de longueurs d'onde disponible est affectée à celui-ci.

La méthode de la longueur d'onde la plus utilisée (*most used*) vise à réutiliser les mêmes longueurs d'onde souvent sollicitées par souci de garantir une certaine fiabilité de la transmission. Intuitivement, si on utilise des longueurs d'onde qui ont déjà effectué quelques transmissions, il est probable qu'elles fonctionnent comme avant et elles présentent moins de risques de tomber en panne que celles qui n'ont pas été utilisées.

La méthode du *max-sum* essaie de maximiser le nombre restant de longueurs d'onde sur le chemin optique après l'affectation d'une longueur d'onde à ce dernier.

La méthode de la capacité relative de perte (*RCL* : *Relative Capacity Loss*) est une amélioration de la méthode max-sum. On maximise dans ce cas le nombre restant de longueurs d'onde disponibles pour de futures connexions sur le chemin optique établi.

Dès que le problème de routage et d'affectation de longueurs d'onde est résolu, le flot de données à envoyer entre la source et la destination peut être acheminé selon un certain débit. Mais la défaillance d'un élément du réseau (liaison, commutateur, etc.) le long du chemin choisi peut entraîner de grandes pertes de données. La prochaine section aborde la gestion des fautes dans les réseaux et les principes mis en œuvre pour assurer leur survivabilité.

### 2.3 Gestion des fautes et restauration de trafic

La protection contre la restauration du trafic affecté prend de plus en plus d'importance dans les réseaux. Beaucoup plus de trafics sont acheminés sur les mêmes supports de communication et il importe de garantir la fiabilité des réseaux pour éviter de nombreuses pertes de données aux multiples usagers de ces réseaux. Des études menées sur deux ans, entre avril 1992 et mars 1994 (Stern, 2000), ont montré que les erreurs humaines, les catastrophes naturelles et les surcharges de réseaux sont les principales causes des pannes. Le nombre d'usagers affectés par ces pannes s'élevait environ à 216 690. Dans les réseaux optiques, la rupture des fibres optiques est la source principale des pannes. En 1987, un câble optique de 12 fibres faisant partie d'un

important réseau dorsal a cassé, entraînant la mise hors circuit de 125 000 communications et la perte de 100 000 connexions deux secondes après la rupture. La réparation a pris 2 heures et coûté des millions de dollars. À juste titre donc, plusieurs études ont été menées au sujet des défaillances dans la littérature et la plupart d'entre elles portent sur des réseaux en anneaux.

Dans les réseaux optiques, les connexions qui sont établies de manière à détecter et à corriger les pannes sont appelées des connexions sécuritaires (*dependable connections*). Le modèle de défaillance le plus étudié pour ces connexions est celui de la rupture éventuelle d'une seule liaison (*single link failure*). Les modèles considérant la rupture simultanée de plusieurs liaisons sont beaucoup plus compliqués et n'aboutissent pas à une optimisation de l'utilisation des ressources du réseau. Par contre, un autre type de défaillance plus réaliste - la panne d'un composant (*element failure*) - caractérisant la panne d'un des éléments du réseau (nœud ou liaison) est aussi étudié par Kodialam et al. (2001). La protection contre cette dernière défaillance demande beaucoup plus de ressources de la part du réseau car un nœud défaillant entraîne la rupture de toutes les liaisons qui lui sont incidentes.

Les deux méthodes préconisées (Mohan et al., 2001) pour gérer ces défaillances matérielles ou logicielles sont les suivantes :

- la *méthode proactive* dans laquelle des ressources de sauvegarde sont dédiées et réservées à l'avance pour chaque connexion établie ;
- la *méthode réactive* ou restauration dynamique qui consiste à trouver des solutions alternatives selon les ressources disponibles lorsqu'une panne survient.

Les approches de protection qui illustrent le mieux la méthode proactive sont la *protection 1+1* et la *protection 1 à N*.

Selon l'approche 1+1, on prévoit une route de protection et une longueur d'onde en réserve chaque fois qu'on établit un chemin. Le second chemin est disjoint en nœuds et en liaisons du premier. Ainsi, en cas de panne sur le chemin initial, les nœuds extrêmes (source et destination) utilisent l'itinéraire et la longueur d'onde de

sauvegarde. Le principe de la protection 1+1 est illustré à la Figure 2.5. Lorsque le chemin initial établi entre la source  $s$  et la destination  $t$  est affecté par une liaison rompue, l'information est transmise à  $s$  et ce dernier retransmet alors les données sur le chemin de sauvegarde. Cette méthode a pour avantage de fournir un temps de réaction aux pannes très court puisque des ressources résiduelles du réseau étaient réservées d'avance.

La protection 1 à  $N$  est une variante de la protection 1+1 où un chemin ou une liaison de secours peut être établie pour protéger plusieurs chemins actifs dans le réseau. La plupart du temps, les chemins protégés par agrégation sont établis entre les mêmes nœuds source et destination. La valeur de  $N$  varie entre 1 et 14 selon les normes ANSI-1988 (*American National Standards Institute*). Quand deux pannes surviennent simultanément sur deux chemins différents, il y a concertation et le chemin de sauvegarde est affecté à celui qui a la plus haute priorité ou la plus grande qualité de service.

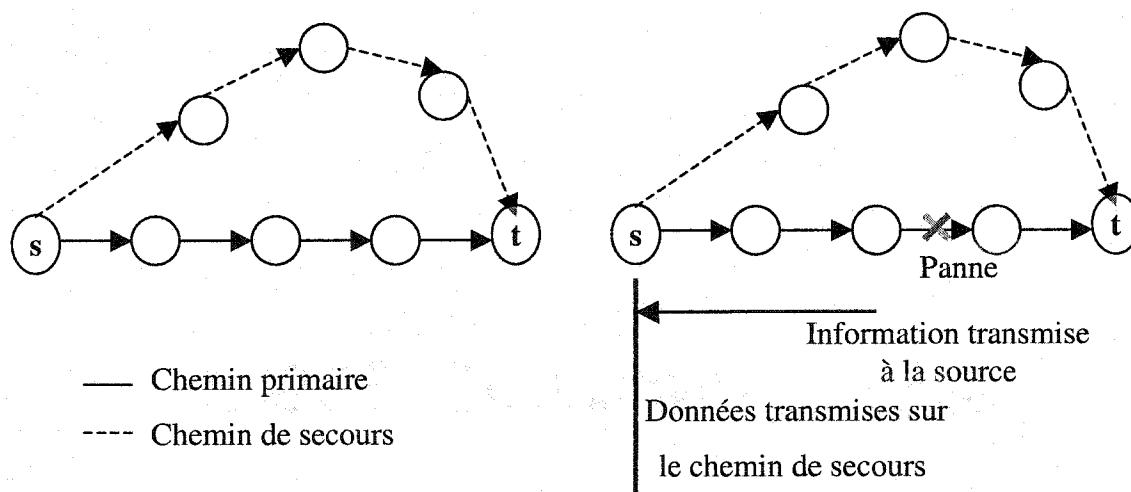


Figure 2.5 Principe de la protection 1+1

Pour la méthode réactive, il existe un avantage majeur : on n'a pas besoin d'entête de contrôle dans le flot de données acheminées, ce qui diminue la charge du réseau. Par contre, en cas de panne, on n'a pas la garantie à 100% de la méthode

proactive de retrouver un autre chemin. Deux techniques majeures sont utilisées pour la méthode réactive : la *restauration de liaison* et la *restauration de chemin*.

La restauration de liaison rétablit la communication seulement autour de la liaison endommagée. Les nœuds source et destination ne sont pas conscients de la panne; les nœuds aux extrémités de la liaison défaillante découvrent dynamiquement un itinéraire autour de ce dernier et y déroutent toutes les longueurs d'onde qui étaient utilisées. Un exemple d'illustration de la restauration de liaison pour une panne de liaison ou de nœud est donné à la Figure 2.6. Si la liaison intermédiaire  $m$  tombe en panne par exemple, le nœud  $a$  est averti et il s'occupe de rechercher une liaison de secours  $x$  qui le relie au nœud précédent le nœud  $t$ .

Par contre, la restauration de chemin informe le nœud source par propagation lorsqu'une panne survient et ce dernier découvre indépendamment un chemin de secours sur une base bout à bout. Lorsqu'une nouvelle route est découverte, les modifications adéquates sont faites et la source redirige la connexion sur ce nouveau chemin; sinon la connexion est bloquée. La restauration de chemin utilise mieux les ressources du réseau que la restauration par liaison, mais elle nécessite plus de signalisation et donc plus de temps pour remédier à une panne.

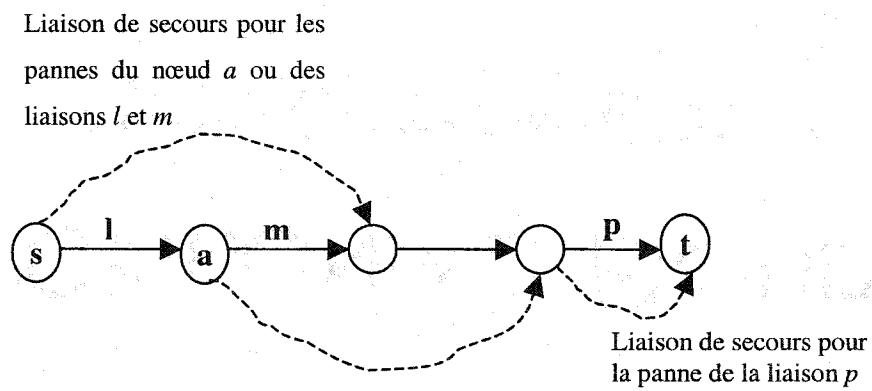


Figure 2.6 Restauration de liaison pour une panne d'élément

## 2.4 Modélisation du taux de blocage

La modélisation du taux de blocage dans les réseaux optiques a fait l'objet de plusieurs travaux recensés par Sridharan et al. (2000). Dans la plupart des cas, les modèles étaient trop complexes ou ne tenaient pas compte de l'aspect dynamique du trafic ou de la contrainte de continuité de longueur d'onde. Subramaniam et Azizoglu (1996) ont proposé un modèle basé sur les chaînes de Markov que nous exposons brièvement ici. Le modèle tient compte du type de routage proposé et de la stratégie d'affectation aléatoire des longueurs d'onde après le choix du chemin. L'étude des autres stratégies d'affectation aurait été fort complexe au niveau du calcul de la probabilité de choisir une longueur d'onde ou une autre. Dans le cas de la stratégie aléatoire, la probabilité d'être choisie est égale pour chaque longueur d'onde.

Nous utiliserons les probabilités suivantes dont la plupart ont été définies par Subramaniam et Azizoglu :

- $Q(w_f) = \Pr \{ w_f \text{ longueurs d'onde soient libres sur une liaison} \}$
- $Q_P(w_f) = \Pr \{ w_f \text{ longueurs d'onde soient libres sur un chemin } P \text{ de longueur } l \}$
- $S(w_f | x_{pf}) = \Pr \{ y_f \text{ longueurs d'onde soient libres sur une liaison d'un chemin } p \text{ sachant que } x_{pf} \text{ longueurs d'onde sont libres sur la liaison précédente de } p \}$
- $U(z_c | y_f, x_{pf}) = \Pr \{ z_c \text{ requêtes continuent sur la liaison courante à partir de la liaison précédent d'un chemin } p \text{ sachant que } y_f \text{ longueurs d'onde sont libres sur la liaison courante de } p \text{ et que } x_{pf} \text{ longueurs d'onde sont libres sur la liaison précédente de } p \}$
- $R(n_f | x_{ff}, y_f, z_c) = \Pr \{ n_f \text{ longueurs d'onde soient libres sur un chemin à deux sauts sachant que } x_{ff} \text{ longueurs d'onde sont libres sur la première liaison}$

de ce chemin,  $y_f$  longueurs d'onde sont libres sur la seconde liaison et  $z_c$  requêtes continuent sur la seconde liaison à partir de la première liaison }

- $T^{(l)}(n_f, y_f) = \Pr \{ n_f \text{ longueurs d'onde soient libres sur un chemin de longueur } l \text{ et } y_f \text{ longueurs d'onde soient libres sur la } l^{\text{ème}} \text{ liaison de ce chemin} \}$
- $p_l = \Pr \{ \text{un chemin de longueur } l \text{ soit choisi pour le routage} \}$

La probabilité de blocage se calcule alors par :

$$P_b = \sum_{l=1}^{N-1} \sum_{y_f=0}^W T^{(l)}(0, y_f) p_l \quad (2.1)$$

où :

$$T^{(l)}(n_f, y_f) = \sum_{x_{pf}=0}^W \sum_{x_g=0}^W \sum_{z_c=0}^{\min(W-x_{pf}, W-y_f)} R(n_f | x_{ff}, z_c, y_f) U(z_c | y_f, x_{pf}) S(y_f | x_{pf}) T^{(l-1)}(x_{ff}, x_{pf}) \quad (2.2)$$

Les auteurs donnent de plus amples détails de calcul pour les autres probabilités.

On remarque que la probabilité  $p_l$ , qui tient compte du choix du chemin par l'algorithme de routage, apparaît dans l'expression finale; ce qui démontre l'importance du choix des chemins dans la réduction du taux de blocage.

Pour le FPLC, Li et Somani (1999) ont calculé des probabilités  $\Pr(P_\alpha^1)$  et  $\Pr(P_\alpha^2)$  qu'une requête arrivant pour une paire source-destination  $\alpha$  soit acheminée sur le premier ou le second chemin candidat  $P_\alpha^1$  et  $P_\alpha^2$ . D'après la technique du FPLC, le premier chemin est choisi s'il a plus de longueurs d'onde libres que le deuxième chemin. Sinon, le deuxième est choisi, d'où :

$$\Pr(P_\alpha^1) = \sum_{i=1}^W Q_{P_\alpha^1}(i) \left( \sum_{n=0}^i Q_{P_\alpha^1}(n) \right) \quad \text{et} \quad \Pr(P_\alpha^2) = \sum_{i=1}^W Q_{P_\alpha^2}(i) \left( \sum_{n=0}^{i-1} Q_{P_\alpha^2}(n) \right) \quad (2.3)$$

avec :

$$Q_P(w_f) = \sum_{y_f=0}^W T_P^{(l(P))}(w_f, y_f)$$

La seule chose qui change pour le FPLC- $k$  est qu'au lieu d'étudier toutes les liaisons de chaque chemin, on en étudie  $k$ ; donc les probabilités deviennent :

$$\hat{\Pr}(P_\alpha^1) = \sum_{i=1}^W \hat{Q}_{P_\alpha^1}^k(i) \left( \sum_{n=0}^i \hat{Q}_{P_\alpha^2}^k(n) \right) \left( 1 - V_{P_\alpha^1}^{(l-k)}(0, i) \right) \quad (2.4)$$

$$\hat{\Pr}(P_\alpha^2) = \sum_{i=1}^W \hat{Q}_{P_\alpha^2}^k(i) \left( \sum_{n=0}^i \hat{Q}_{P_\alpha^1}^k(n) \right) \left( 1 - V_{P_\alpha^2}^{(l-k)}(0, i) \right) \quad (2.5)$$

où :

$$\hat{Q}_P^k(w_f) = \sum_{y_f=0}^W T_{P_j}^{(k)}(w_f, y_f) \quad (2.6) \quad \text{et}$$

$$V_P^l(n_f, x_{P_f}) = \sum_{y_f=0}^W \sum_{z_c=0}^{\min(W-x_{P_f}, W-y_f)} \sum_{x_{ff}=0}^y R(n_f | x_{P_f}, z_c, x_{ff}) U(z_c | y_f, x_{P_f}) S(y_f | x_{P_f}) V_P^{(l-1)}(x_{ff}, y_f) \quad (2.7)$$

## CHAPITRE 3

### ALGORITHMES DE ROUTAGE OPTIQUE

Les algorithmes de routage optique sont généralement limités par l'absence de mémoire tampon au niveau des routeurs du réseau. Il est donc important de trouver une stratégie simple et efficace pour trouver les chemins et acheminer les données dans des délais relativement courts. Dans ce chapitre, nous exposons les méthodes que nous avons proposées pour la résolution du problème de routage dynamique dans les réseaux optiques. Nos algorithmes essaieront de minimiser, pour un réseau tout-optique recevant des requêtes de connexion à ses différents nœuds, le nombre de demandes bloquées. Suite à un survol de l'architecture des réseaux optiques considérés, nous discuterons de la formulation mathématique du problème ainsi que de la fonction objective à minimiser. Ensuite, nous exposerons les deux approches que nous préconisons pour résoudre le problème puis nous donnerons les caractéristiques de nos algorithmes. Nous achèverons ensuite par quelques illustrations de ces derniers et leur analyse de complexité.

#### 3.1 Architecture de réseau optique considérée

Le réseau optique que nous étudierons est constitué de  $N$  nœuds interconnectés par  $M$  liaisons optiques unidirectionnelles. Dans ce graphe  $(N, M)$ , chaque nœud peut être un commutateur optique ou un émetteur/récepteur. Chaque liaison est constituée de  $f$  fibres ayant chacune  $w$  longueurs d'onde  $(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_w)$ . Chaque fibre est également caractérisée par des attributs physiques comme sa largeur de bande, sa latence et un estimateur qui représente son coût. Nous considérons qu'il n'y a pas de convertisseur de longueur d'onde au niveau des routeurs, donc toute demande acceptée doit être acheminée le long de son chemin optique suivant une seule et même longueur d'onde. On suppose également que les routeurs optiques disposent de suffisamment de mémoire

pour conserver des données comme l'état courant du réseau et qu'il existe une unité centrale capable de contrôler l'ensemble des opérations qui s'effectuent dans le réseau.

Le modèle utilisé comprend un nœud central qui traite les informations générales du réseau, à savoir la topologie, les bris éventuels de liaisons, ainsi que tous les évènements qui se déroulent dans le réseau. Le nœud principal que nous qualifierons de serveur central ne fait pas partie de la topologie du réseau qu'il charge à partir d'un fichier de données. Il conserve en mémoire non volatile les algorithmes de routage et d'affectation de longueurs d'onde. Les autres nœuds représentent les routeurs optiques et ont une interface qui leur permet d'identifier leurs voisins immédiats et toutes les liaisons incidentes. Ainsi, tout nœud dont une des liaisons sortantes ou entrantes brise s'en aperçoit aussitôt et le serveur central se charge des mesures requises. Les liaisons contrôlent quant à elles la gestion de leurs longueurs d'onde. À tout moment, une liaison peut savoir combien de longueurs d'onde sont disponibles et lesquelles acheminent de l'information.

Les requêtes de connexion arrivent à chaque nœud suivant un processus de Poisson avec un taux moyen  $\lambda$ . La durée moyenne d'une connexion acceptée suit également une loi exponentielle de moyenne  $1/\mu$ . Le trafic supporté par chaque nœud se résume alors à :  $\rho = \lambda/\mu$  évalué en Erlangs.

Chaque arrivée de paquets est traitée dans le réseau comme un évènement discret. Les arrivées de paquets aux nœuds sont générées au départ par un créateur d'évènements qui affecte à chaque nœud du réseau un ensemble de transmissions à réaliser au cours de la simulation. Des évènements de bris de liaison peuvent également survenir dans le réseau suivant un taux spécifié. Les différentes entités du réseau sont illustrées à la Figure 3.1.

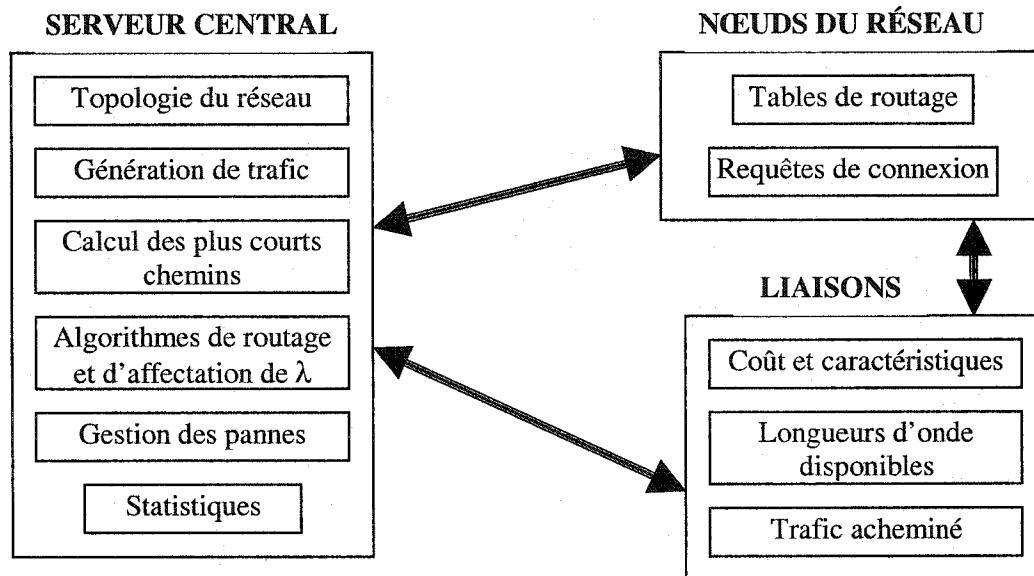


Figure 3.1 Architecture du système de résolution

Au début du problème, le serveur central initialise la topologie du réseau; chaque nœud et chaque liaison sont initialisés avec leurs caractéristiques respectives. Ensuite, le serveur lance la génération du trafic et des pannes éventuelles. L'algorithme de calcul des  $K$  plus courts chemins entre chaque paire de nœuds est ensuite exécuté. L'algorithme est défini à la Figure 3.2 et tire sa base de la méthode de Yen modifiée (Martins et al., 2000). Au départ, on calcule le plus court chemin trouvé entre chaque paire de nœuds par l'algorithme de Dijkstra. Ensuite, pour trouver les autres plus courts chemins pour chaque paire de nœuds, on enlève temporairement du réseau la liaison de plus faible coût sur le plus court chemin et on recherche à nouveau le plus court chemin. Cette approche s'inspire de la notion de déviation utilisée par Yen dans son algorithme. En effet, à chaque itération, il modifiait le réseau de manière à enlever une liaison dont la disparition entraînait la déviation du flot sur un autre chemin. Son approche était plus rigoureuse que celle de notre algorithme car elle permettait de trouver les  $K$  plus courts chemins. Par contre, après avoir analysé la complexité d'implémentation de son algorithme avec les structures de données que nous avions, nous avons décidé de simplifier le choix et la stratégie d'enlèvement des liaisons du réseau.

L'algorithme de Dijkstra, utilisé ici dans sa forme la plus simple pour calculer le plus court chemin entre un sommet  $s$  et les autres nœuds, est montré à la Figure 3.3. La complexité de l'algorithme de Dijkstra se résume dans le pire cas à  $O(N^2)$  puisqu'à chaque étape, on transfert un nœud de l'ensemble  $\bar{S}$  vers  $S$ , ce qui donne  $N$  itérations. Et durant chaque itération, on recherche le nœud de  $\bar{S}$  qui a la plus petite distance temporaire, ce qui se fait en  $O(N)$  dans le pire cas.

Soient  $Paths$  l'ensemble des  $n$  plus courts chemins trouvés et  $PathsVector$  l'ensemble des chemins possibles. Pour un chemin  $p$  donné, désignons par  $L(p)$  l'ensemble des liaisons qui constituent  $p$ , classées par ordre croissant de coût.

```

Paths = Φ
PathsVector = Φ
POUR chaque nœud s ∈ N
    POUR chaque nœud d ∈ N – {s}
        Trouver le plus court chemin PCC entre s et d avec Dijkstra
        PathsVector = PathsVector ∪ {PCC}
        n = 1
        TANT QUE (n < K) ET (PathsVector ≠ Φ)
            Prendre le premier chemin p de PathsVector
            PathsVector = PathsVector – {p}
            Paths = Paths ∪ {p}
            Trouver L(p)
            TANT QUE (L(p) ≠ Φ) ET (n < K)
                Choisir la liaison l de plus faible coût dans L(p)
                L(p) = L(p) – {l}
                Enlever l du réseau et rechercher le PCC' entre s et d (Dijkstra)
                SI le PCC' est trouvé
                    PathsVector = PathsVector ∪ {PCC'}
                    Paths = Paths ∪ {PCC'}
                    Incrémenter n
                FIN SI
                Réinsérer l dans le réseau
            FIN TANT QUE
        FIN TANT QUE
    FIN POUR
FIN POUR

```

Figure 3.2 Algorithme de Yen modifié pour les  $K$  plus courts chemins

La complexité de l'algorithme de Yen modifié dans le pire cas s'évalue à  $O(N^4H^2)$  si on désigne par  $H$  la longueur maximale en nombre de sauts (hops) d'un chemin.  $H$  est également appelé le *diamètre* du réseau. En effet, on a deux grandes itérations sur les  $N$  nœuds du réseau et ensuite l'algorithme de Dijkstra ( $O(N^2)$ ) qui est appliqué au cours de chaque itération. Ensuite, un tri est fait au niveau des liaisons de chaque chemin analysé pour les classer par ordre croissant de coût. Ce tri par sélection présente une complexité moyenne de  $O(H^2)$ .

```

 $S \leftarrow \emptyset$ 
 $\bar{S} \leftarrow X$ 
 $d(s) \leftarrow 0$ 
 $\text{pred}[s] \leftarrow 0$ 
 $d(i) \leftarrow +\infty \quad \forall i \in X \setminus \{s\}$ 
Tant que ( $\bar{S} \neq \emptyset$ )
   $i \leftarrow \arg \min_{j \in \bar{S}} (d(j))$ 
   $S \leftarrow S \cup \{i\}$ 
   $\bar{S} \leftarrow \bar{S} \setminus \{i\}$ 
   $\forall (i, j) \in \omega^+(i)$ 
    Si ( $d(j) > d(i) + c_{ij}$ )
      Alors
         $d(j) \leftarrow d(i) + c_{ij}$ 
         $\text{pred}[j] \leftarrow i$ 
    Fin Si
  Fin  $\forall$ 
Fin Tant que

```

Figure 3.3 Forme simple de l'algorithme de Dijkstra

L'algorithme de Yen modifié est donc utilisé par le serveur central pour calculer un ensemble de  $K$  plus courts chemins entre chaque paire de nœuds du réseau. La métrique utilisée pour le calcul des plus courts chemins est une valeur de coût positive fixée par l'administrateur du réseau optique avant sa mise en fonction. Toutefois, cet

algorithme resterait le même si les coûts des liaisons variaient dans le temps. L'algorithme est exécuté chaque fois que la configuration topologique du réseau change : bris de liaison, ajout d'un nouveau nœud, etc. Cela permet de mettre à jour les tables de routage des nœuds et permet de fournir à tous les algorithmes de routage implémentés une base commune d'analyse. La complexité des algorithmes de routage présentés plus loin dans ce chapitre ne tient donc pas compte de celle de l'algorithme de Yen modifié.

Une autre approche pour réaliser le routage consisterait à ajouter une métrique de coût tenant compte des longueurs d'onde et donc d'avoir une fonction de coût variable pour les liaisons :

$$C_{(i,j)} = f\left(c_{(i,j)}, g\left(\sum_{k=1}^W \lambda_k\right)\right)$$

Cela nous permettrait de ne pas développer un algorithme séparé de sélection des routes en fonction de l'état du réseau. Mais pour partir sur une même base de comparaison que les algorithmes de routage optique habituels, nous avons préféré écarter ce modèle d'algorithme. Dans le cas où la fonction aurait uniquement tenu compte du nombre de longueurs d'onde portées par chaque fibre, cela n'aurait rien changé dans notre modèle puisque toutes les liaisons sont mono-fibres et qu'elles ont le même nombre initial de longueurs d'onde. Par contre, pour une fonction de coût dynamique qui tiendrait compte du nombre de longueurs d'onde libres sur chaque liaison, il aurait fallu exécuter l'algorithme par intervalles de temps très courts ou à l'arrivée d'une requête. Dans ce cas, la complexité totale de l'algorithme serait égale à  $O(N^4H^2) + O(M)$  puisqu'au départ de l'algorithme, il faudra affecter à toutes les liaisons du réseau un coût tenant compte de leur état courant. Pour un ensemble de requêtes élevé, le temps de calcul s'accroît alors exponentiellement. L'inconvénient majeur de cette stratégie relève donc de l'augmentation considérable de la surcharge des ressources du réseau en terme d'information de contrôle.

Dès que l'ensemble des  $K$  plus courts chemins est connu, on commence alors à traiter les requêtes de connexion qui arrivent aux nœuds grâce aux algorithmes de

routage dynamique. Ces algorithmes font le choix du meilleur chemin parmi l'ensemble pré-calculé en tenant compte de l'état courant des liaisons du réseau. Ces opérations sont répétées jusqu'à ce que tout le trafic soit satisfait ou que les ressources du réseau soient congestionnées.

La table de routage de chaque nœud comprend l'énumération des  $K$  chemins possibles pour chaque destination avec les éléments suivants :

- le prochain nœud vers une destination donnée et ce pour toutes les destinations possibles dans le réseau ;
- le coût du chemin qui joint le nœud source au nœud destination d'un paquet.

Ainsi, chaque fois que l'algorithme de Yen modifié est exécuté ou qu'il survient une panne de liaison dans le réseau, les informations contenues dans cette table sont mises à jour. Au départ, les tables de routage sont initialisées avec l'adresse du nœud local comme prochain nœud et un coût nul.

Un changement survient dans l'état du réseau lorsqu'on satisfait une requête ou qu'une liaison est brisée. Toutes ces variations n'entraînent pas systématiquement la diffusion de l'information dans le réseau. Dans le cas d'une panne de liaison, un message est envoyé par les nœuds aux extrémités de la liaison défaillante, au serveur central qui se charge de réexécuter l'algorithme de Yen modifié pour mettre à jour toutes les tables de routage des nœuds du réseau. La méthode de proche en proche, selon laquelle chaque nœud contacterait ses voisins les plus proches pour les avertir de l'événement et permettre ainsi une diffusion par étapes de l'information, est donc inutile dans ce cas où tout est centralisé. Par contre, le fait d'allouer les ressources pour satisfaire une requête n'est pris en considération qu'au moment de l'arrivée d'une autre requête qui pourrait avoir besoin des mêmes ressources. L'état des liaisons composant les différents chemins possibles est alors analysé par les algorithmes de sélection de chemin pour effectuer leur choix.

Le réseau optique considéré dans cette section peut présenter quelques limitations comme :

- la sécurité du réseau : l'attaque ou la mise hors service du serveur principal du réseau implique l'arrêt du système ;
- les échanges de données entre le serveur central et les autres ressources du réseau peuvent considérablement accroître la charge du réseau. De plus, la mauvaise transmission de ces données peut entraîner des erreurs de routage et des pertes de paquets.

Pour remédier à tous ces problèmes, nous préconiserons une approche basée sur l'existence d'un serveur de sauvegarde en cas de panne du serveur principal. Il faudra également planter un algorithme de cohésion entre ces deux serveurs pour maintenir à tout moment une base de données cohérente entre ces deux machines. De plus, un médium de transport doit être exclusivement destiné aux échanges de données entre les nœuds du réseau et le serveur principal pour éviter que les informations de contrôle ne surchargent le réseau.

Dans la prochaine section, nous présenterons le problème du routage dynamique dans cet environnement ainsi que les divers algorithmes que nous proposons.

### 3.2 Formulation du problème de routage dynamique optique

Le problème du routage dynamique est un problème NP-complet pour lequel plusieurs techniques de résolution ont été proposées. Il se formule comme suit :

Étant donné :

- la configuration topologique du réseau ( $N$  nœuds et  $M$  liaisons géographiquement localisées) ;
- le nombre  $f$  de fibres par liaisons et le nombre  $W$  de longueurs d'onde par fibre ;

Soit :

- $F_{ij}^{sdw}$  l'appel qui s'établit entre la source  $s$  et la destination  $d$  en utilisant la longueur d'onde  $w$  sur la liaison  $(i,j)$  ;

- $\gamma_{sdw}$  le trafic (nombre d'appels acceptés) entre la source  $s$  et la destination  $d$  qui utilise la longueur d'onde  $w$  ;

On cherche à :

- minimiser le taux de blocage en maximisant l'utilisation de chaque liaison du réseau :

$$\max_{ij} \sum_{s,d,w} F_{ij}^{sdw}$$

Sujet aux contraintes :

- $F_{ij}^{sdw} \in \{0, 1\}$
- $F_{ij} = \sum_{s \in N} \sum_{d \in N} \sum_{w \in W} F_{ij}^{sdw} \leq W$
- $\sum_i F_{ij}^{sdw} - \sum_k F_{jk}^{sdw} = \begin{cases} -\gamma_{sdw} & \text{si } s = j \\ \gamma_{sdw} & \text{si } d = j \\ 0 & \text{autrement} \end{cases}$

Par choix des  $F_{ij}^{sdw}$  suivant l'état des liaisons dans le réseau.

Ce problème ne se résout pas facilement puisque chaque acceptation d'appel dépend des appels précédents acceptés sur le même chemin et du trafic supporté par chaque liaison dans le réseau. Et en augmentant le nombre d'appels acceptés, on diminue le nombre d'appels bloqués et, par conséquent, on limite la congestion dans le réseau.

Aucune méthode mathématique ne pouvant être appliquée pour résoudre le problème de manière dynamique, nous proposons par la suite deux approches heuristiques pour essayer de minimiser l'utilisation maximale des ressources du réseau.

### 3.3 Algorithmes de routage dynamique proposés

Face au problème du routage dynamique, nous avons élaboré deux stratégies susceptibles de limiter le blocage des requêtes de connexion. La première se base sur une estimation globale de la congestion dans le réseau en se fiant à l'analyse de certains paramètres au niveau des liaisons. Quant à la seconde, elle s'avère plus rigoureuse par l'analyse exhaustive des ressources qui sont disponibles dans le réseau, et l'évaluation du nombre de longueurs d'onde disponibles par liaison permet d'effectuer un choix final de chemin. Les deux algorithmes que nous exposons par la suite illustrent ces deux approches qui exploitent les avantages et inconvénients des algorithmes de routage développés auparavant.

#### 3.3.1 Estimation de la congestion : ECR

Avec la méthode appelée ECR pour *Estimated Congestion Routing*, nous avons essayé de trouver une fonction qui donne une bonne estimation de la congestion dans le réseau. Selon la plupart des définitions, la congestion survient quand la majorité des liaisons utilisent toute leur capacité pour satisfaire les requêtes qui entrent dans le réseau. Dans ce contexte, les critères les plus importants que nous avons identifiés pour estimer le degré de congestion d'une liaison optique sont les suivants :

- les caractéristiques de la liaison (largeur de bande, nombre de longueurs d'onde, etc.) ;
- le trafic courant acheminé sur la liaison (nombre de longueurs d'onde occupées).

L'algorithme de routage doit donc obtenir la meilleure approximation possible de la congestion du réseau pour éviter les chemins avec des liaisons surchargées. Sachant que plus un chemin est congestionné, moins il y a de longueurs d'ondes libres, nous avons pris un estimateur basé sur l'inverse du nombre de longueurs d'onde libres. Ensuite, le coût d'une liaison sur un chemin étant une valeur indicative de sa fiabilité ainsi que de son importance dans le réseau, nous avons ajouté à notre estimateur un coefficient basé sur le rapport du coût de la liaison au coût du chemin. Nous en sommes

alors arrivés à proposer la fonction suivante pour évaluer la congestion d'une liaison  $(i,j)$  dans un chemin potentiel  $P$  :

$$C_{i,j} = \frac{1}{(\text{nombre de } \lambda \text{ libres})} * \frac{\text{Coût } (P)}{\text{Coût } ((i,j))} \quad (3.1)$$

avec  $\text{Coût } (P) = \sum_{(i,j) \in P} \text{Coût } ((i,j))$

La congestion d'un chemin potentiel  $P$  qui contient  $l$  hops revient alors à :

$$C_P = \frac{\sum_{(i,j) \in P} C_{(i,j)}}{l} \quad (3.2)$$

Le chemin ayant le taux de congestion le moins élevé est alors utilisé pour acheminer la requête. En effet, étant donné la distribution exponentielle uniforme de la durée des connexions, on essaie de répartir le plus uniformément possible la charge dans le réseau. L'algorithme du *ECR* est décrit à la Figure 3.4.

```

POUR chaque requête de connexion entre une paire de noeuds (s-d)
  POUR chaque chemin P calculé par l'algorithme de Yen modifié
    POUR chaque liaison de P
      SI (nombre de λ libres ≠ 0)
        Calculer Ci,j
      SINON
        Ci,j = 50
      FIN SI
    FIN POUR
    Calculer la moyenne de congestion CP pour chaque chemin P
  FIN POUR
  Choisir le chemin de congestion moyenne la moins élevée pour la requête
  SI deux chemins ont la même moyenne de congestion
    Prendre le chemin le plus court en nombre de hops
  FIN SI
FIN POUR

```

**Figure 3.4 Algorithme du ECR**

Prenons le réseau représenté à la Figure 3.5. Une requête de connexion entre les noeuds 1 et 6 arrive à un moment  $t$ . Les longueurs d'onde libres sur chaque liaison à cet

instant sont indiquées à leur côté. Ainsi, pour la liaison (1,4), on a les longueurs d'onde  $\lambda_1, \lambda_3, \lambda_5$  et  $\lambda_6$ . Par souci de simplification, supposons que les liaisons ont chacune un coût unitaire. L'algorithme de routage commence par identifier les quatre chemins possibles dans ce cas pour router la connexion si on suppose que  $K = 4$ . On a les chemins  $\{1,6\}$ ,  $\{1,2,6\}$ ,  $\{1,3,6\}$  et  $\{1,4,5,6\}$ . Ensuite chaque route est analysée.

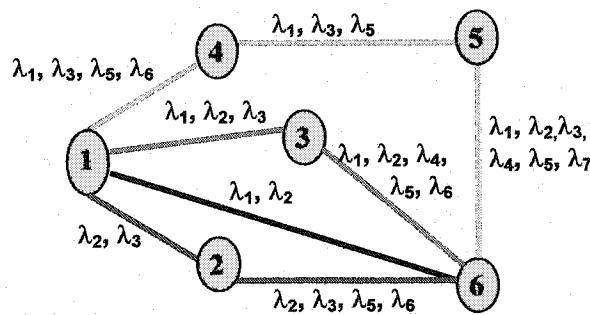


Figure 3.5 Réseau optique

Pour le chemin  $\{1,2,6\}$  par exemple, on a le degré de congestion suivant pour chaque liaison :

$$(1,2) : C = (2)^{-1} * 2/1 = 1$$

$$(2,6) : C = (4)^{-1} * 2/1 = 1/2$$

Le niveau de congestion de ce chemin est donc égal à la moyenne arithmétique de ces deux congestions :

$$C_{\{1,2,6\}} = (1 + 1/2)/2 = 3/4$$

De la même manière, on obtient la congestion pour les autres chemins :

$$\bullet \quad C_{\{1,6\}} = 1/2$$

$$\bullet \quad C_{\{1,3,6\}} = (2/3 + 2/5)/2 = 8/15$$

$$\bullet \quad C_{\{1,4,5,6\}} = (3/4 + 1 + 1/2)/3 = 9/12$$

Le chemin choisi est celui qui a le plus petit niveau de congestion, donc dans ce cas on trouve le chemin  $\{1,6\}$ .

D'autres modèles de fonctions de calcul de la congestion utilisant des outils statistiques pour les chemins candidats ont été étudiés. Des fonctions basées sur l'écart moyen entre les indices de congestion de chaque liaison, pour chaque chemin, n'ont pas été concluantes car on réalise très tôt que cela augmente le temps de calcul de l'algorithme sans apporter d'améliorations sensibles. Par ailleurs, l'estimation de la moyenne de congestion avec les indices ci-après cités s'est révélée insuffisante :

$$C_{i,j} = \frac{(\text{nombre de } \lambda \text{ occupées})}{(\text{nombre de } \lambda \text{ total})} * \frac{\text{Coût (P)}}{\text{Coût ((i, j))}} \quad (3.3)$$

$$C_{i,j} = \frac{\text{Nombre d'utilisations} * (\text{nombre de } \lambda \text{ occupées})}{(\text{nombre de } \lambda \text{ total})} * \frac{\text{Coût (P)}}{\text{Coût ((i, j))}} \quad (3.4)$$

$$C_{i,j} = \frac{\text{Nombre d'utilisations}}{(\text{nombre de } \lambda \text{ libres})} * \frac{\text{Coût (P)}}{\text{Coût ((i, j))}} \quad (3.5)$$

On constate ici l'apparition d'un autre critère qui est celui du nombre d'utilisation des liaisons. Il représente le nombre moyen de requêtes acheminées à un moment donné par chaque longueur d'onde. Ce paramètre peut nous servir à identifier l'historique de chaque liaison. Ainsi, plus les liaisons d'un chemin auraient été utilisées, moins grande serait la probabilité de choisir ce chemin car, au bout d'un certain moment, ce dernier risque d'être congestionné.

### 3.3.2 Méthode hybride : HFPLC-*k*

La méthode hybride que nous avons développée tente de fusionner la technique du FPLC et celle du LLR. Elle est appelée *Hybrid Fixed-Paths Least Congested-k* (HFPLC-*k*) car elle est une modification de la méthode FPLC-*k*. Ainsi pour *K* chemins candidats, on choisit le meilleur en identifiant sur les *k* liaisons de chaque chemin candidat le nombre de longueurs d'onde partagées par les *k* liaisons. Le chemin choisi sera celui dont les *k* liaisons les plus chargées ont un maximum de longueurs d'onde en commun. Ainsi, on utilise le critère de continuité de longueur d'onde très important pour

la méthode FPLC, mais on évite de le vérifier pour toutes les liaisons de chaque chemin. Par contre, on analyse  $k$  liaisons pour chaque chemin, comme le FPLC- $k$ , mais à la différence principale que ce sont les  $k$  liaisons les plus congestionnées pour chaque chemin. On introduit donc le critère de la liaison la plus congestionnée utilisé dans la méthode du LLR. Le résumé de l'algorithme du HFPLC- $k$  est décrit à la Figure 3.6.

L'application d'un critère additionnel comme celui des  $k$  liaisons les plus chargées apporte une précision supplémentaire susceptible de nous éviter de faire le mauvais choix de route. Il est évident qu'en portant notre attention sur les liaisons les plus chargées, on restreint l'éventail des possibilités pour le choix d'une route, par contre, on évite de surcharger les liaisons qui portent le plus de trafic dans le réseau. En indexant les liaisons les plus chargées pour comparer les degrés de congestion de chaque route, on prend en compte les goulets d'étranglement que constituent ces dernières. Cela augmente alors nos chances de ne pas choisir les routes les plus chargées et d'éviter une surutilisation des ressources. Notons que  $k \leq H$  et que le cas  $k = H$  correspond à l'algorithme du FPLC pour la méthode initiale du FPLC- $k$ .

```

POUR chaque requête de connexion entre une paire de nœuds (s-d)
    POUR chaque chemin P calculé par l'algorithme de Yen modifié
        Comparer et classer toutes les liaisons de P des plus chargées aux
        moins chargées (en fonction du nombre de longueurs d'onde libres)
        POUR les k premières liaisons de P
            Vérifier la contrainte de continuité de  $\lambda$  en calculant le
            nombre de longueurs d'onde libres communes à ces liaisons
        FIN POUR
    FIN POUR
    Choisir pour la requête le chemin ayant le plus grand nombre de longueurs
    d'onde libres communes à ses  $k$  liaisons les plus chargées
    SI deux chemins ont le même nombre
        Prendre le chemin le plus court en nombre de hops
    FIN SI
FIN POUR

```

Figure 3.6 Algorithme du HFPLC- $k$

Reprendons l'exemple de la Figure 3.5, avec  $k = 2$ . Si on désigne par  $d$  le niveau de disponibilité de chaque chemin candidat, on a :

- $\{1,6\}$  : la seule liaison donc la plus congestionnée est  $(1,6)$ ; d'où :

$$d_{\{1,6\}} = 2 \quad (\lambda_1 \text{ et } \lambda_2);$$

- $\{1,2,6\}$  : les 2 liaisons les plus chargées sont respectivement  $(1,2)$  et  $(1,6)$  ; d'où :

$$d_{\{1,2,6\}} = 2 \quad (\lambda_2 \text{ et } \lambda_3);$$

- $\{1,3,6\}$  : les 2 liaisons les plus chargées sont  $(1,3)$  et  $(1,6)$  ; d'où :

$$d_{\{1,3,6\}} = 2 \quad (\lambda_1 \text{ et } \lambda_2);$$

- $\{1,4,5,6\}$  : les 2 liaisons les plus chargées sont  $(4,5)$  et  $(1,4)$  ; d'où :

$$d_{\{1,4,5,6\}} = 3 \quad (\lambda_1, \lambda_3 \text{ et } \lambda_5).$$

À partir de ces calculs, on choisit donc le chemin ayant le plus de disponibilité, c'est-à-dire  $\{1,4,5,6\}$ . On remarque également que le chemin choisi est identique à celui du FPLC. L'algorithme du FPLC- $k$  aurait donné le même résultat, mais en analysant les liaisons  $(1,4)$  et  $(4,5)$  alors que notre algorithme s'est plutôt basé sur les liaisons importantes dans ce cas, c'est-à-dire  $(5,6)$  et  $(1,4)$ .

### 3.4 Analyse de complexité

Dans le pire cas, la complexité de l'algorithme d'estimation de la congestion (ECR) est d'ordre  $\mathbf{O}(NKH^2W)$ . En effet, on évalue les  $K$  chemins candidats pour une requête arrivant à chacun des  $N$  nœuds. Pour chaque chemin, on en calcule le coût par une opération de complexité  $\mathbf{O}(H)$  et ensuite on parcourt le diamètre  $H$  du réseau pour calculer le degré de disponibilité de chaque liaison avec l'analyse dans le pire cas, comme dans le meilleur cas, des  $W$  longueurs d'onde portées par chaque liaison.

La complexité du HFPLC- $k$  dans le pire cas est de  $\mathbf{O}(NkkH^2W)$ . Pour chaque nœud du réseau qui reçoit une requête, on analyse les  $K$  chemins possibles. Pour chaque chemin, on effectue un tri par sélection des liaisons les plus congestionnées, ce qui

donne une complexité moyenne de  $O(H^2)$ . Ensuite, pour les  $k$  premières liaisons de chaque chemin, on analyse les longueurs d'onde disponibles. Dans le pire cas, toutes les liaisons d'un chemin  $P$  n'acheminent aucune requête. La liaison la plus congestionnée dispose donc de  $W$  longueurs d'onde libres au moment de la requête. Il faut alors analyser ces  $W$  longueurs d'onde au niveau des  $k-1$  autres liaisons d'où la complexité de  $O(NKkH^2W)$ . Dans le meilleur cas, la liaison la plus congestionnée n'a aucune longueur d'onde disponible et la complexité de l'algorithme retombe à  $O(NKH^2)$ .

Puisque nous avons décidé d'établir une base commune pour la comparaison de tous les algorithmes, nous avons réévalué la complexité des autres algorithmes comme le PCC, le FPLC, le FPLC- $k$  et le LLR. Ainsi, la complexité du PCC est pratiquement nulle puisque l'algorithme de calcul des  $K$  plus courts chemins nous donne dès le départ le plus court chemin. Cet algorithme ne fait donc que choisir le premier chemin de l'ensemble des  $K$  plus courts chemins sans appliquer d'autres critères.

Par contre, pour le LLR, la complexité dans le pire cas s'évalue à  $O(NKH)$ . Ainsi, pour toutes les requêtes qui arrivent aux  $N$  nœuds, on étudie l'ensemble des  $K$  plus courts chemins. Pour chaque chemin, on compare ensuite toutes les liaisons, ce qui dans le pire cas, correspond à  $H$  opérations pour déterminer la liaison la plus congestionnée. Dans le meilleur des cas, le chemin peut être composé d'une seule liaison, ce qui réduirait la complexité du LLR à  $O(NK)$ .

Quant à la complexité du FPLC dans le pire cas, on l'estime à  $O(NKHW)$ . En effet, on analyse  $K$  chemins candidats pour chaque requête de connexion. Sur chaque chemin, on vérifie liaison par liaison, donc dans le pire cas  $H$  fois, la continuité de longueur d'onde sur les  $W$  longueurs d'onde de chaque fibre.

Le FPLC- $k$  présente une complexité similaire à celle du FPLC, sauf qu'il fait la vérification de la contrainte de continuité de longueur d'onde sur  $k$  liaisons au lieu de toutes les liaisons de chaque chemin. Donc, dans le pire cas, la complexité du FPLC- $k$  revient à  $O(NKkW)$  avec  $k \leq H$ .

On constate donc, à partir de tous ces calculs de complexité, que les algorithmes des moins complexes aux plus complexes sont le PCC, le LLR, le FPLC- $k$ , le FPLC, le

ECR et enfin le HFPLC- $k$ . Ceci est principalement dû au fait que, dans les algorithmes de plus forte complexité, on vérifie déjà la contrainte de continuité de longueur d'onde pour éviter qu'une route choisie ne bloque pas la requête en l'absence d'une longueur d'onde libre. Le cas du ECR, qui ne vérifie pas cette contrainte mais qui reste un peu plus complexe, s'explique par la présence du calcul de coût total pour chaque chemin. On aurait pu réduire ce temps en faisant calculer par l'algorithme de Yen modifié présenté précédemment, le coût de chaque chemin, mais cela n'aurait pas servi aux autres algorithmes et la base de comparaison aurait été faussée.

Après l'analyse des différents algorithmes de routage dynamique, nous abordons dans la prochaine section le comportement du système en cas de pannes.

### 3.5 Restauration en cas de pannes

Étant donné la haute fiabilité des routeurs ou des commutateurs optiques en général, nous ne prendrons en compte ici que les bris simples de liaison. Face à l'occurrence éventuelle de ces pannes, nous avons décidé d'adopter une approche réactive. La restauration de chemin a donc été choisie pour pallier les bris de liaisons. Ainsi, lorsqu'une liaison brise, tous les nœuds du réseau sont mis au courant par diffusion du message de la part du serveur central. De nouveaux chemins sont calculés et les tables de routage mises à jour. Les requêtes qui étaient acheminées sur la liaison brisée sont alors réinitialisées par les nœuds source. Le traitement réalisé lors du bris de toute liaison dans le réseau est résumé à la Figure 3.7.

En cas de panne d'une liaison  $L_{i,j}$

Prévenir le serveur central

Diffuser l'information à tous les nœuds par le biais du serveur central

Remettre les tables de routage à jour par l'exécution de l'algorithme de Yen modifié

POUR chaque requête courante acheminée sur la liaison  $L_{i,j}$

Exécuter l'algorithme de sélection de chemin pour trouver un nouveau chemin

FIN POUR

**Figure 3.7 Traitement en cas de bris de liaison**

## CHAPITRE 4

### IMPLÉMENTATION ET ÉVALUATION DE PERFORMANCE

Ce chapitre présente des détails d'implémentation et l'évaluation de performance des algorithmes de routage dynamique et d'affectation de longueurs d'onde proposés au chapitre précédent. Nous y simulons les algorithmes de routage suivants : le plus court chemin, le LLR, le FPLC, le FPLC- $k$ , le ECR et le HFPLC- $k$ . Une fois décrite l'implémentation des différents modules, nous passerons à la phase d'expérimentation et d'évaluation proprement dite. Un plan d'expériences sera proposé et les résultats analysés. Nous tiendrons essentiellement compte du taux de blocage total et du temps de calcul, comme dans la majorité des analyses de performance d'algorithmes de routage optique conduites dans la littérature (Mokhtar et al., 1998; Li et al., 1999; Zhou et al., 2002). Le but de notre évaluation sera de valider la performance de nos deux algorithmes par rapport à celle des algorithmes de routage optique classiques.

#### 4.1 Implémentation du système

Les algorithmes proposés ont été implémentés en Java. En effet, l'utilisation d'outils graphiques pour illustrer notre travail est plus simple en Java qu'en C++. Aucun matériel optique n'a été utilisé et toutes les simulations se sont faites à l'aide de *Borland JBuilder Professional version 4*.

##### 4.1.1 Analyse globale de l'implémentation

Le système se compose de trois modules principaux :

- le module topologique du réseau ;
- le module de création et de gestion des évènements ;
- le module de routage et d'affectation de longueurs d'onde.

### Module topologique

Ce module charge la topologie du réseau optique à partir d'un fichier texte spécifié en argument de notre programme. Ce fichier comprend des données sur la position des nœuds, leurs noms ainsi que les caractéristiques de chaque liaison. En l'absence de spécification de certains paramètres, des valeurs par défaut leur sont affectées pour permettre au programme de garder une certaine cohérence. Ainsi, le contenu typique d'un fichier topologique pour un réseau en anneau est illustré à la Figure 4.1.

#	name	xcoord	ycoord			
node	n0	0.3	0.0			
node	n1	0.7	0.0			
node	n2	1.0	0.3			
node	n3	1.0	0.7			
node	n4	0.7	1.0			
node	n5	0.3	1.0			
node	n6	0.0	0.7			
node	n7	0.0	0.3			
#	name	node1	node2	speed	delay	cost (optional)
link	10	n0	n1	57600	0.01	
link	11	n1	n2	57600	0.01	
link	12	n2	n3	57600	0.01	
link	13	n3	n4	57600	0.01	
link	14	n4	n5	57600	0.01	
link	15	n5	n6	57600	0.01	
link	16	n6	n7	57600	0.01	
link	17	n7	n0	57600	0.01	

Figure 4.1 Fichier de représentation topologique d'un réseau en anneau

Comme on peut le remarquer, la désignation d'un nœud se fait par l'acronyme *node* suivi par son nom ainsi que ses coordonnées spatiales normalisées entre 0 et 1. Pour les liaisons, on commence par *link* suivi du nom de la liaison et de celui des nœuds à ses deux extrémités. Ensuite, on peut spécifier la capacité de la liaison en bits par seconde, le délai de propagation en secondes et le coût de la liaison en option. Au cas où toutes ces valeurs ne seraient pas spécifiées, les valeurs par défaut sont : *speed* → 57600, *delay* → 0.001 et *cost* → 1. Le coût des liaisons est fixé au départ par l'administrateur en

fonction des caractéristiques physiques de chaque liaison et de son importance ou de sa fiabilité dans le réseau. Quant aux lignes débutant par un caractère #, elles permettent d'insérer des commentaires dans le fichier.

Le fonctionnement du module topologique est assuré par le serveur principal qui se charge de l'initialiser et de lire le contenu du fichier texte. Ensuite, les instances de nœuds et de liaisons sont créées selon leurs critères spécifiques. Les classes représentant chacune de ces entités seront présentées dans la prochaine section.

### ***Module de création et de gestion des évènements***

Ce module revêt une importance capitale de par sa place dans la simulation. Il se charge de créer le trafic aux nœuds ainsi que des bris éventuels de liaisons dans le réseau, selon le taux de pannes spécifié par l'utilisateur.

Pour générer le trafic initial, les informations telles que le taux d'arrivée initial des paquets en unités de requêtes par seconde, la durée moyenne de maintien d'une requête en secondes et la durée totale de la simulation en secondes sont fournies par l'utilisateur. Ces données nous permettent alors de créer aléatoirement des évènements inscrits dans des fichiers texte dont un exemple de contenu est présenté à la Figure 4.2.

Dans ce fichier, on note la première ligne qui désigne un commentaire comme dans le cas du fichier topologique. Ensuite, chaque événement est identifié par le sigle *event* suivi du temps auquel il survient. Les requêtes de connexion sont désignées par *pkt* accompagné des noms des nœuds source et destination. Enfin, vient le contenu en caractères ASCII du message qui sera acheminé au cours de la connexion si la requête est acceptée. On peut également remarquer que plusieurs évènements surviennent en même temps, démontrant la capacité du simulateur à traiter de manière concurrentielle les requêtes qui arrivent à différents nœuds. En réalité, dans le simulateur, tous ces évènements sont empilés dans une file d'évènements et sont défilés les uns à la suite de l'autre pour être traités.

À la fin du fichier, on peut voir quelques événements de bris de liaisons qui sont programmés. L'acronyme *state* combiné avec le mot *down* ou *up* désigne respectivement une liaison qui tombe en panne et une liaison réparée.

```
#      time  command
event 0.0 pkt n5 n7 Test_d'envoi_de_paquets_courts
event 0.0 pkt n4 n3 Test_d'envoi_de_paquets_courts
event 0.0 pkt n2 n6 Test_d'envoi_de_paquets_courts
event 0.0 pkt n1 n7 Test_d'envoi_de_paquets_courts
event 0.0 pkt n5 n2 Test_d'envoi_de_paquets_courts
event 0.0 pkt n0 n4 Test_d'envoi_de_paquets_courts
event 0.0 pkt n3 n7 Test_d'envoi_de_paquets_courts
event 0.0 pkt n1 n2 Test_d'envoi_de_paquets_courts
event 0.0 pkt n4 n7 Test_d'envoi_de_paquets_courts
event 0.0 pkt n6 n1 Test_d'envoi_de_paquets_courts
event 0.027034637741908323 pkt n3 n1 Test_d'envoi_de_paquets_courts
event 0.027034637741908323 pkt n3 n4 Test_d'envoi_de_paquets_courts
event 0.027034637741908323 pkt n5 n4 Test_d'envoi_de_paquets_courts
event 0.0 state l2 down
event 0.02 state l2 up
event 0.667 state l2 down
event 0.687 state l2 up
event 1.334 state l6 down
event 1.354 state l6 up
event 2.0010000000000003 state l1 down
event 2.0210000000000004 state l1 up
event 2.668 state l5 down
event 2.688 state l5 up
```

Figure 4.2 Exemple de fichier de génération de trafic

#### *Module de routage et d'affectation de longueurs d'onde*

Ce module se compose essentiellement de fonctions implémentant les algorithmes décrits au chapitre 3. La première fonction exécutée est l'algorithme de routage principal dénommé *algoRouting*. Elle lance alors l'algorithme de Dijkstra pour calculer le plus court chemin entre tous les nœuds et ensuite l'algorithme de Yen modifié pour calculer les autres plus courts chemins selon le nombre spécifié par

l'utilisateur. Par la suite, le support de trafic est activé et les algorithmes de sélection de chemin ainsi que d'affectation de longueurs d'onde sont exécutés.

Dans le but de mieux comprendre le système, nous allons présenter son diagramme de classes ainsi que les différentes relations qui existent entre elles.

#### 4.1.2 Modèle de classes

Le diagramme de classes utilisé est présenté à la Figure 4.3. On y retrouve six classes principales : *Network*, *Node*, *Link*, *Lambda*, *Event* et *RoutingTable*.

Les autres classes comme *NetworkMap*, *RoutingTablesView* et *StepView* nous ont principalement servi à implémenter des outils graphiques pour mieux tester nos algorithmes. En effet, la classe *NetworkMap* permet de dessiner les nœuds du réseau sur une figure aux dimensions normalisées, comme le montre l'exemple de la Figure 4.4. On y voit les différents nœuds avec leurs noms dans des carrés ainsi que les coûts des liaisons qui les relient entre eux. La classe *StepView* permet, quant à elle, de faire une simulation pas-à-pas, donc de suivre le comportement du système après chaque arrivée de requêtes. Cependant, cette option a été désactivée pour l'ensemble des tests finaux puisque la simulation avait une durée importante. Enfin, la classe *RoutingTablesView* nous permettait d'afficher le contenu des tables de routage pour le  $x^{\text{ème}}$  plus court chemin entre tous les nœuds du réseau. Le résultat graphique du plus court chemin ( $K=0$ ) est illustré à la Figure 4.5. On constate qu'il y a treize tables de routage pour autant de nœuds dans le réseau. La première colonne de chaque table indique le numéro de la destination, la seconde colonne représente le prochain nœud pour transmettre les données et la troisième donnée désigne le coût total pour chaque chemin. Lorsque la destination est identique à la source, nous décidons de mettre par convention un chemin de coût nul.

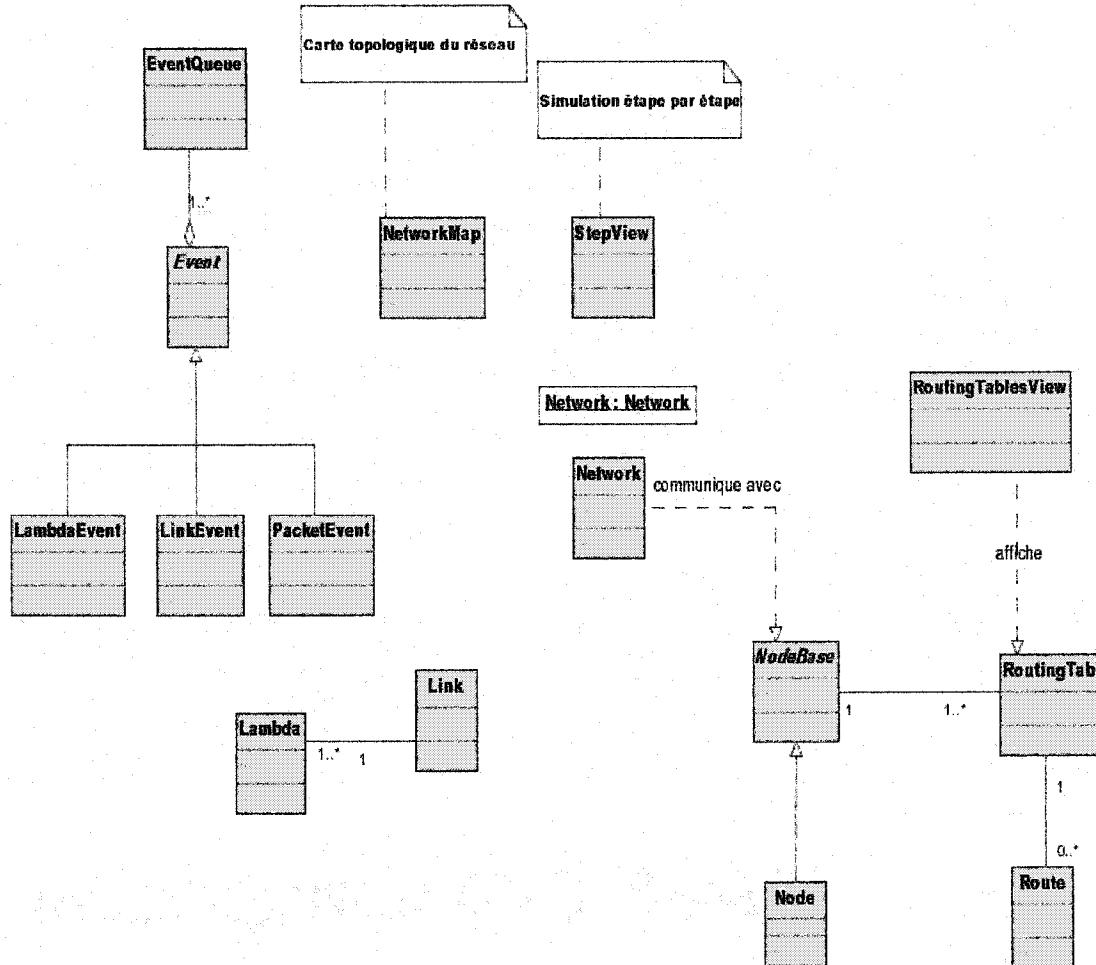


Figure 4.3 Diagramme de classes du système

La classe *Network* désigne le serveur principal de notre réseau. Elle a une instance unique qui sert à contrôler l'ensemble du réseau. Cette dernière se charge de l'initialisation du réseau ainsi que des procédures de gestion et de création d'évènements. Pour faciliter nos travaux d'expérimentation, le programme principal, qui permet de lancer l'exécution des différents algorithmes de routage testés, a été implanté au niveau de cette classe. Les résultats générés sont alors sauvegardés dans des fichiers texte soigneusement identifiés pour fins d'analyse. La classe *Network* possède également

la liste de nœuds et la liste des liaisons qui forment le réseau afin de communiquer aisément avec eux.

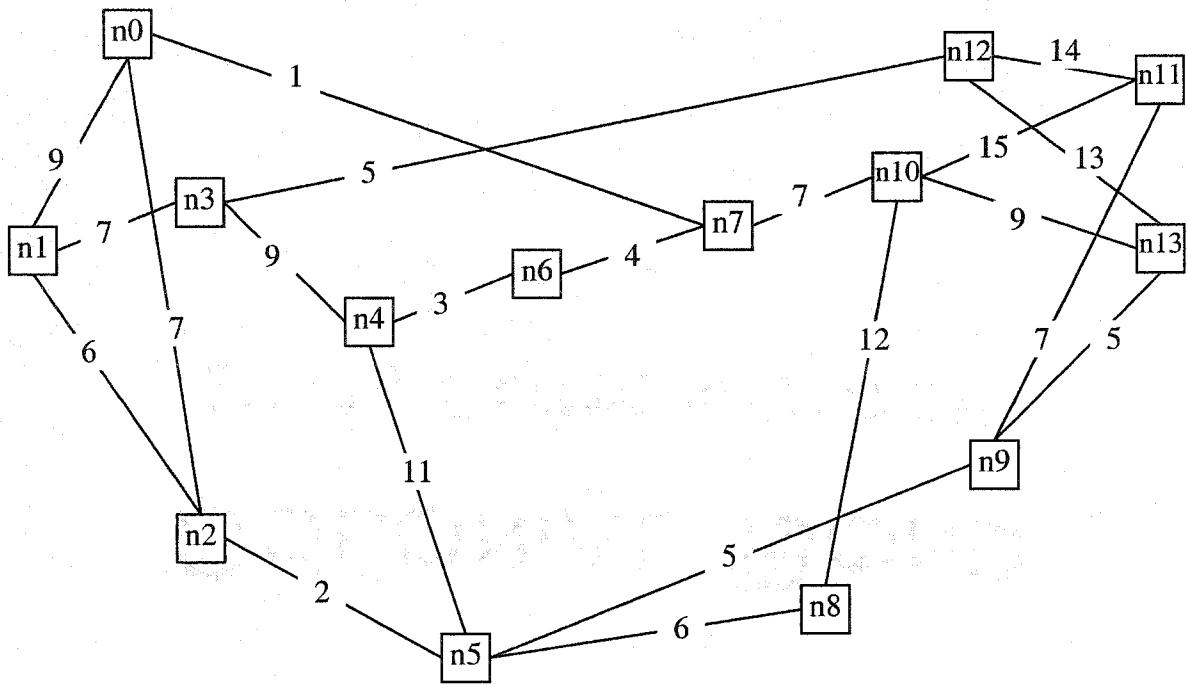


Figure 4.4 Topologie du réseau simulé pour ARPANET

La seconde classe importante est la classe *Node* qui représente les nœuds du réseau et hérite de la classe *NodeBase*. Cette dernière possède une adresse unique dans le réseau, la liste des nœuds voisins et des liaisons qui lui sont attachés ainsi qu'un ensemble de tables de routage. La classe enfant *Node* implémente alors les fonctions comme celles d'initialiser le nœud, de remplir les tables de routage représentées par la classe *RoutingTable*, et de notifier que les interfaces de liaison sont toujours actives ou en panne. Puisque nous avons adopté une approche centralisée, les nœuds n'implémentent pas de fonction de routage proprement dite et tous les calculs sont laissés au serveur central. Toutefois, lors de la réception d'un paquet, le nœud connaît la route empruntée et achemine les données sur la liaison adéquate. Ainsi, chaque route est une instance de la classe *Route* qui contient l'adresse de la destination, le coût total de la route, ainsi que les prochains nœuds sur la route jusqu'à la destination. Le diagramme de

classes illustre également le fait qu'une table de routage peut contenir zéro ou plusieurs routes selon l'état courant d'un nœud.

Routing Simulator: Routing Tables															
0	0	0	▲	0	1	9	▲	0	2	7	▲	0	1	16	▲
1	0	9		1	1	0		1	2	6		1	3	7	
2	0	7		2	1	6		2	2	0		2	1	13	
3	1	16		3	1	7		3	1	13		3	0	3	
4	7	8		4	3	16		4	5	13		4	3	9	
5	2	9		5	2	8		5	2	2		5	1	15	
6	7	5		6	0	14		6	0	12		6	4	12	
7	0	1		7	0	10		7	0	8		7	4	16	
8	2	15		8	2	14		8	5	8		8	1	21	
9	2	14		9	2	13		9	5	7		9	1	20	
10	7	6		10	0	17		10	0	15		10	4	23	
11	2	21		11	2	20		11	5	14		11	12	19	
12	1	21		12	3	12		12	1	18		12	3	5	
13	7	17		13	2	18		13	5	12		13	12	18	
Routing table: n5				Routing table: n6				Routing table: n7				Routing table: n8			
0	2	9	▲	0	7	5	▲	0	7	1	▲	0	5	15	▲
1	2	6		1	7	14		1	0	10		1	5	14	
2	5	2		2	7	12		2	0	6		2	5	8	
3	2	15		3	4	12		3	6	16		3	5	21	
4	5	11		4	6	3		4	6	7		4	5	17	
5	5	0		5	4	14		5	0	10		5	8	6	
6	2	14		6	6	0		6	7	4		6	5	20	
7	2	10		7	6	4		7	7	0		7	5	16	
8	5	6		8	4	20		8	0	16		8	8	0	
9	5	5		9	4	19		9	0	15		9	5	11	
10	2	17		10	7	11		10	7	7		10	8	12	
11	9	12		11	7	26		11	10	22		11	5	18	
12	2	20		12	4	17		12	6	21		12	5	26	
13	9	10		13	7	20		13	10	16		13	5	16	
Routing table: n10				Routing table: n11				Routing table: n12				Routing table: n13			
0	7	6	▲	0	9	21	▲	0	3	21	▲	0	10	17	▲
1	7	17		1	9	26		1	3	12		1	9	18	
2	7	15		2	9	14		2	3	18		2	9	12	
3	7	23		3	12	19		3	12	5		3	12	18	
4	7	14		4	9	23		4	3	14		4	9	21	
5	7	17		5	9	12		5	3	20		5	9	10	
6	7	11		6	10	26		6	3	17		6	10	20	
7	10	7		7	10	22		7	3	21		7	10	16	
8	10	12		8	9	18		8	3	26		8	9	16	
9	13	14		9	11	7		9	13	18		9	13	5	
10	10	0		10	11	15		10	13	22		10	13	9	
11	10	15		11	11	0		11	12	14		11	9	12	
12	13	22		12	11	14		12	12	0		12	13	13	
13	10	9		13	9	12		13	12	13		13	13	0	

Figure 4.5 Contenu des tables de routage pour le plus court chemin (K=0)

Par ailleurs, la classe *Link* désigne naturellement les liaisons du réseau. Elle a comme attributs son coût, son débit, sa latence ou délai de propagation, les adresses des deux nœuds à ses extrémités, une liste de ses longueurs d'onde initiales, son taux d'utilisation ainsi qu'une variable d'état. Elle offre de multiples fonctionnalités comme le calcul du temps mis pour transmettre le contenu d'un message sur son canal. Par ailleurs, à tout moment, elle peut fournir la liste de ses longueurs d'onde libres

permettant ainsi au serveur principal d'appliquer efficacement l'algorithme d'affectation de longueurs d'onde désiré.

La classe *Lambda* a été créée pour désigner une longueur d'onde et pour développer de manière très modulaire le réseau. Comme le montre le diagramme de classes principal, chaque liaison possède au moins une longueur d'ondes. Au cours de la simulation, ce nombre peut augmenter pour étudier l'impact du nombre de longueurs d'onde initial sur chaque liaison dans le support du trafic. Chaque instance de la classe *Lambda* possède une adresse unique sur sa liaison correspondante ainsi qu'une variable d'état. Lors de l'acheminement de données sur cette longueur d'onde, cette variable est mise à la valeur occupée.

Par ailleurs, la classe *Event* qui appartient au module de création et de gestion des évènements permet de créer des évènements dans le réseau. On distingue alors des évènements de paquets (*PacketEvent*), des évènements de liaison (*LinkEvent*) et des évènements de longueurs d'onde (*LambdaEvent*). Tous ces évènements héritent de la classe *Event* et partagent comme attributs le temps auquel ils surviennent et une fonction chargée de les exécuter. Plus spécifiquement, les évènements de paquets sont créés comme des requêtes lors de la lecture du fichier texte de génération de trafic. Ils contiennent les nœuds source et destination de la connexion courante ainsi que le contenu du message à transmettre en cas d'acceptation de la requête. Les évènements de liaison font référence, quant à elles, aux éventuelles pannes de liaisons enregistrées dans le fichier texte de génération de trafic. Les instances de cette classe ont pour attributs spécifiques la liaison mise en cause et son état au moment de l'occurrence de l'évènement. Enfin, les évènements de longueurs d'onde permettent de spécifier les moments auxquels les longueurs d'onde sont occupées ou sont libérées pour d'autres connexions. Ces évènements sont lancés dès qu'une requête est acceptée et qu'un chemin et une longueur d'onde lui ont été affectés. Une approche par évènements pour affecter l'état des longueurs d'onde semble à première vue inutile mais, quand on se ramène à l'idée de la possible arrivée concurrentielle de plusieurs requêtes, il est

important de traiter les différents états des longueurs d'onde en tenant compte du facteur temporel.

La classe *EventQueue* représente une liste ordonnée des futurs événements dans le réseau. Chaque fois qu'un nouvel événement est créé pour survenir à un temps donné  $t$ , il est inséré dans cette liste. Si son temps d'occurrence est inférieur au dernier événement survenu, c'est-à-dire s'il survient en décalage dans le passé par rapport au présent, il n'est pas ajouté dans la liste et un message d'erreur est lancé. Par contre, si  $t$  est supérieur ou égal au moment présent, l'événement est inséré dans la liste ordonnée de manière chronologique.

Une fois défini l'ensemble des classes utilisées lors de l'implémentation, nous passons maintenant aux méthodes d'implémentation des algorithmes de routage abordés.

#### 4.1.3 Scénario d'implémentation des algorithmes de routage

Pour illustrer l'implémentation des algorithmes de routage dans le système, un diagramme de séquences général est présenté à la Figure 4.6. Dans un premier temps, on y voit l'unique objet de la classe *EventQueue* qui dépile un événement de requête de connexion et lance son exécution. Une requête de connexion arrive donc à un nœud  $i$  du réseau. La requête est acheminée au serveur central par la demande d'exécution de sa fonction *deliver* qui prend en paramètres l'adresse des nœuds source et destination et le contenu du message à transmettre formulé ici sous la forme d'une chaîne de caractères. Le serveur central détermine alors les différents chemins pour atteindre la destination en consultant les tables de routage du nœud source. Sur la base de cette information, la fonction de sélection de chemin *choosePath* est lancée avec en paramètres le nom de l'algorithme de routage désiré (FPLC, LLR, ECR, HFPLC-k, etc..) ainsi que l'ensemble des routes prédéterminées. L'algorithme de routage courant essaie alors de choisir la meilleure route selon ses critères et retourne l'indice du chemin choisi au serveur. Ensuite, ce dernier lance la fonction de recherche d'une longueur d'onde sur le chemin choisi en tenant compte de la stratégie d'affectation courante. Si une longueur d'onde libre est trouvée, un événement d'occupation de la longueur d'onde et un autre

évènement de libération de cette même longueur d'onde sont créés sur toutes les liaisons avec des temps de début et de fin de transmission du message. L'événement d'occupation de la longueur d'onde est alors défilé et exécuté par la pile d'évènements, et la longueur d'onde correspondante est mise dans l'état occupé. Ensuite, l'autre évènement de libération est exécuté et la longueur d'onde libérée. Le trafic d'entrée dans le réseau est ainsi traité requête par requête en utilisant un marqueur de temps d'une précision de  $10^{-16}$  unités de temps.

#### 4.2 Plan d'expériences

Dans cette section, nous présentons la stratégie adoptée pour réaliser notre évaluation de performance ainsi que les différents paramètres que nous faisons varier pour mesurer leur impact dans les performances du réseau. Notre plan d'expériences sera du type *un facteur à la fois*, c'est-à-dire que nous évaluerons les indices de performance du réseau en faisant varier les niveaux de paramètres les uns après les autres. Pour chaque session d'expériences, nous effectuerons 25 simulations et ensuite nous prendrons les moyennes arithmétiques des valeurs générées pour présenter nos résultats. La plate-forme qui sert de support à nos simulations est un ordinateur de bureau utilisant un processeur Pentium III cadencé à 600 MHz et le système d'exploitation *Windows 2000 Professional*. Les différents résultats présentés ont été obtenus alors que l'ordinateur fonctionnait en mode fermé avec un nombre minimum de processus critiques pour limiter l'interférence de ces derniers dans le temps d'exécution des algorithmes.

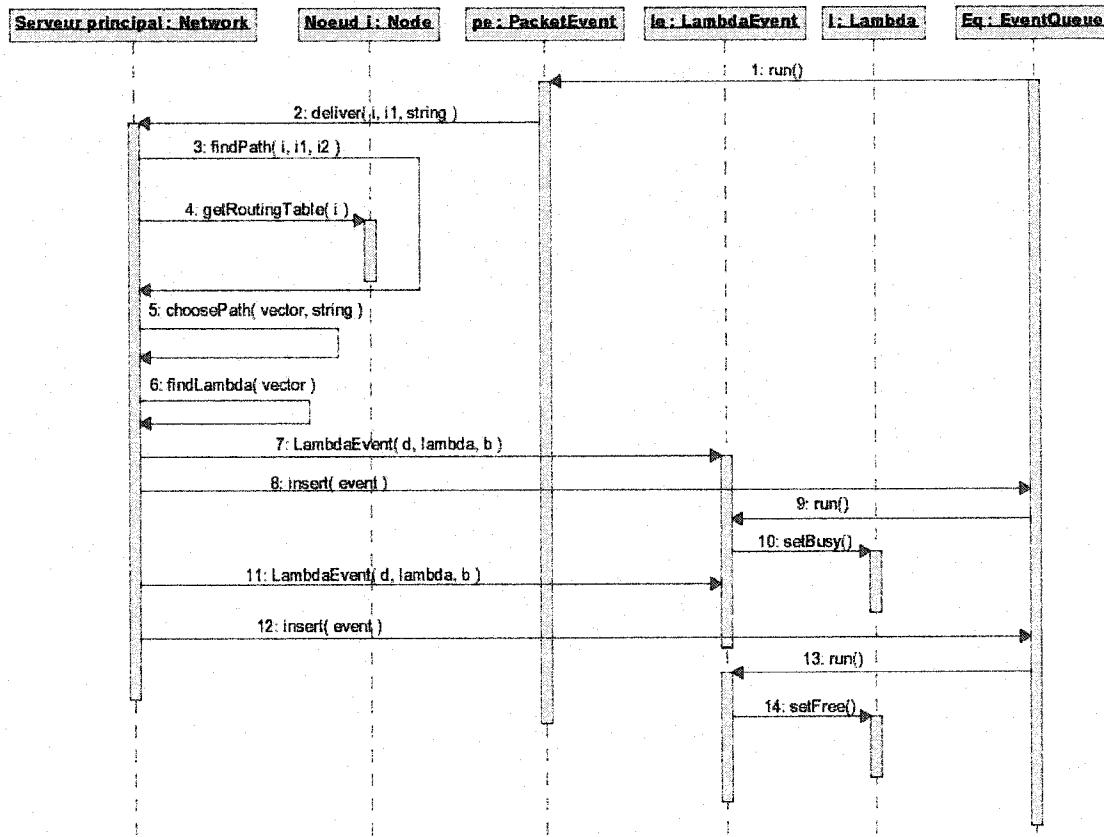


Figure 4.6 Diagramme de séquence du traitement d'une requête

#### 4.2.1 Facteurs primaires

Nous étudierons l'importance des facteurs suivants dans le réseau :

- la *topologie* du réseau ;
- le *niveau du trafic d'entrée* dans le réseau ;
- le *nombre initial de longueurs d'onde* par fibre dans le réseau ;
- les différents *algorithmes de routage* implémentés ;
- les *algorithmes d'affectation de longueurs d'onde*.

Afin d'avoir une vue globale des performances de nos algorithmes de routage, nous avons choisi de les implémenter pour divers réseaux optiques de topologies différentes. Ceci nous permettra de généraliser nos résultats et de ne pas nous enfermer

dans des cas particuliers. Nous étudierons donc des réseaux en anneau, maillé et enfin un réseau général comme celui de l'ARPANET.

Le niveau de trafic est proportionnel au taux d'arrivée initial des requêtes spécifié par l'utilisateur. Les requêtes acceptées transmettent dans le programme, au cours de leur temps de connexion, le même message standard représenté par une phrase.

Le nombre de longueurs d'onde initial pour chaque fibre dans le réseau variera de un à six. Nous avons décidé de nous limiter à ce nombre à cause des limites physiques actuelles des fibres optiques ainsi que des articles relatifs au sujet qui évaluent les réseaux optiques avec un maximum de huit longueurs d'onde par liaison mono-fibre.

Les six algorithmes de routage à savoir le PCC, le FPLC, le LLR, le FPLC- $k$ , le ECR et le HFPLC- $k$  seront évalués ainsi que les quatre algorithmes d'affectation de longueurs d'onde : le *first fit*, le *most used*, le *least used* et le *random*.

#### 4.2.2 Indices de performance

Nous considérons trois indices de performance importants dans notre évaluation :

- le *taux de blocage* qui mesure le nombre de requêtes bloquées par rapport au nombre total de requêtes entrantes à chaque nœud ;
- le *temps de réponse* du réseau qui mesure la durée totale pendant laquelle le réseau essaie de satisfaire tout son trafic d'entrée aux nœuds ;
- et le *coût total* des requêtes acceptées par le réseau.

### 4.3 Résultats

Nous présentons les différents résultats obtenus pour différentes topologies de réseau optique et nous faisons une étude comparative entre algorithmes de routage.

#### 4.3.1 Réseau en anneau

Le réseau en anneau présenté à la Figure 4.7 est composé de 8 nœuds et 16 liaisons. Entre chaque paire de nœuds, il existe initialement deux chemins possibles.

Pour ce réseau, nous avons mis des coûts unitaires sur chaque arc et nous utilisons les variables initiales suivantes :

- taux d'arrivée initial des requêtes  $\Rightarrow 3$  par minute
- taux de bris des liaisons  $\Rightarrow 0.2$
- durée totale de la simulation  $\Rightarrow 10$  minutes.

Les algorithmes de routage tels que le FPLC- $k$  et le HFPLC- $k$  ont été exécutés avec  $k=2$  liaisons à analyser.

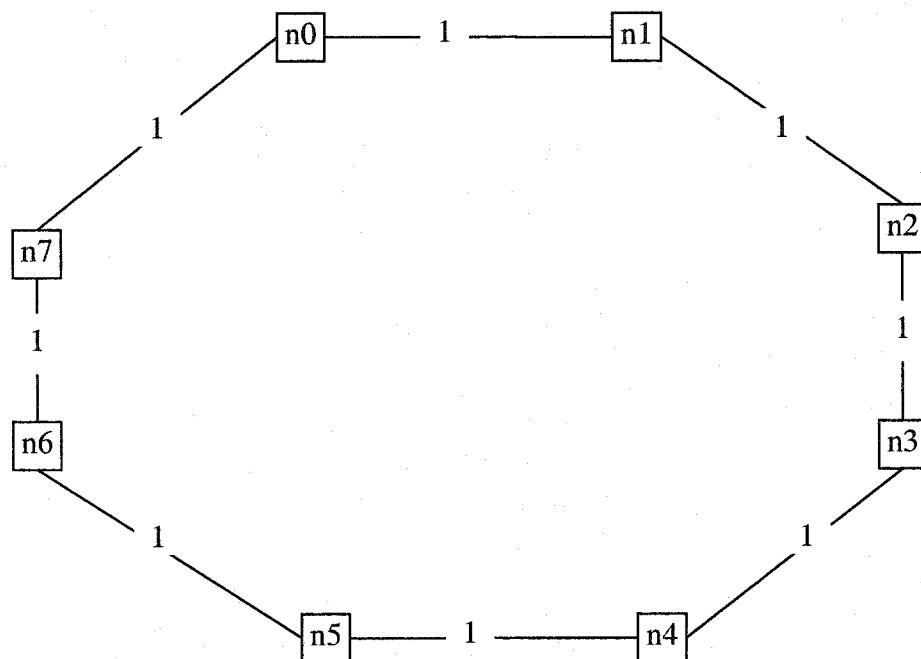


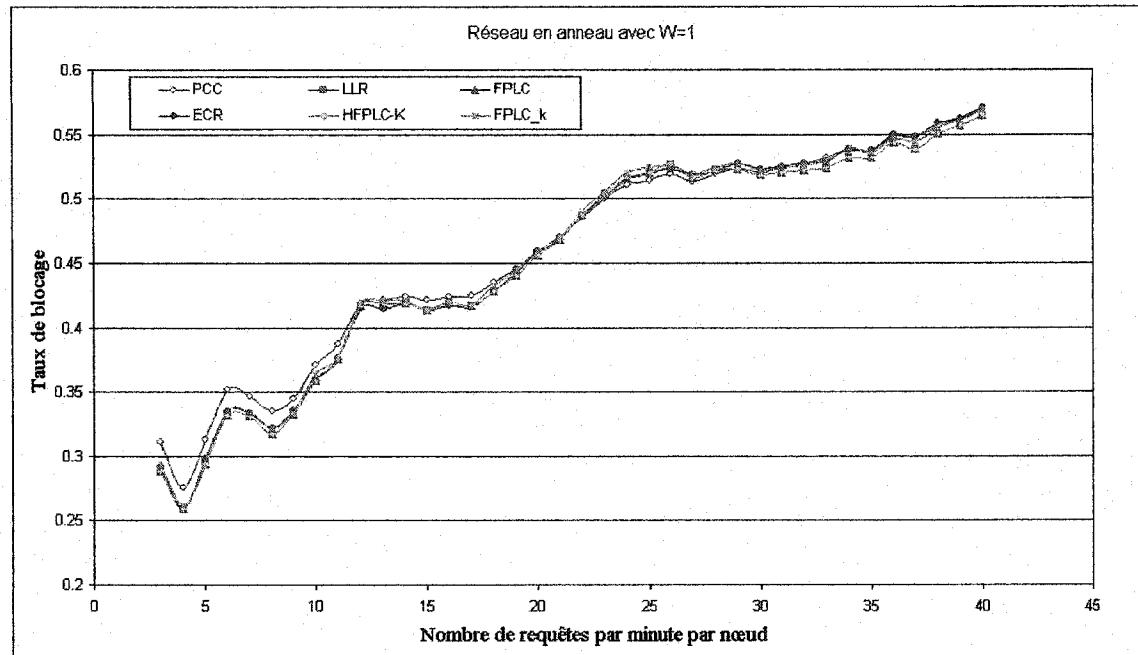
Figure 4.7 Topologie du réseau optique en anneau simulé

#### 4.3.1.1 Évaluation du taux de blocage

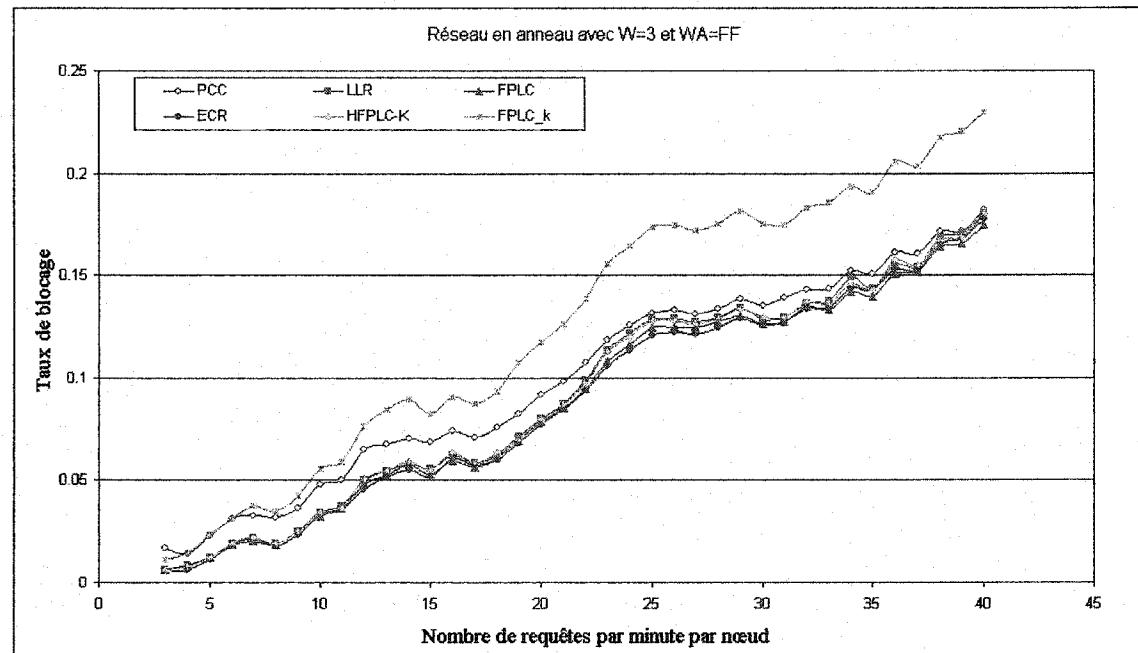
Dans le cas d'un réseau avec des liaisons mono fibres ayant une seule longueur d'onde, les performances de tous les algorithmes de routage ne présentent pas de différences remarquables comme le montre la Figure 4.8. Le ECR et le HFPLC- $k$  se comportent très bien. On note au départ les meilleurs résultats obtenus par les algorithmes dynamiques versus l'algorithme du plus court chemin ; mais quand le trafic augmente en intensité, le taux de blocage augmente proportionnellement et atteint plus

de 50 % pour tous les algorithmes de routage. Cela s'explique par le fait que la seule longueur d'onde disponible par liaison est souvent monopolisée par une requête alors que plusieurs autres arrivent aux noeuds. Les différents algorithmes d'affectation de longueurs d'onde n'ont évidemment aucun effet sur le taux de blocage des requêtes puisque le choix de la longueur d'onde est unique.

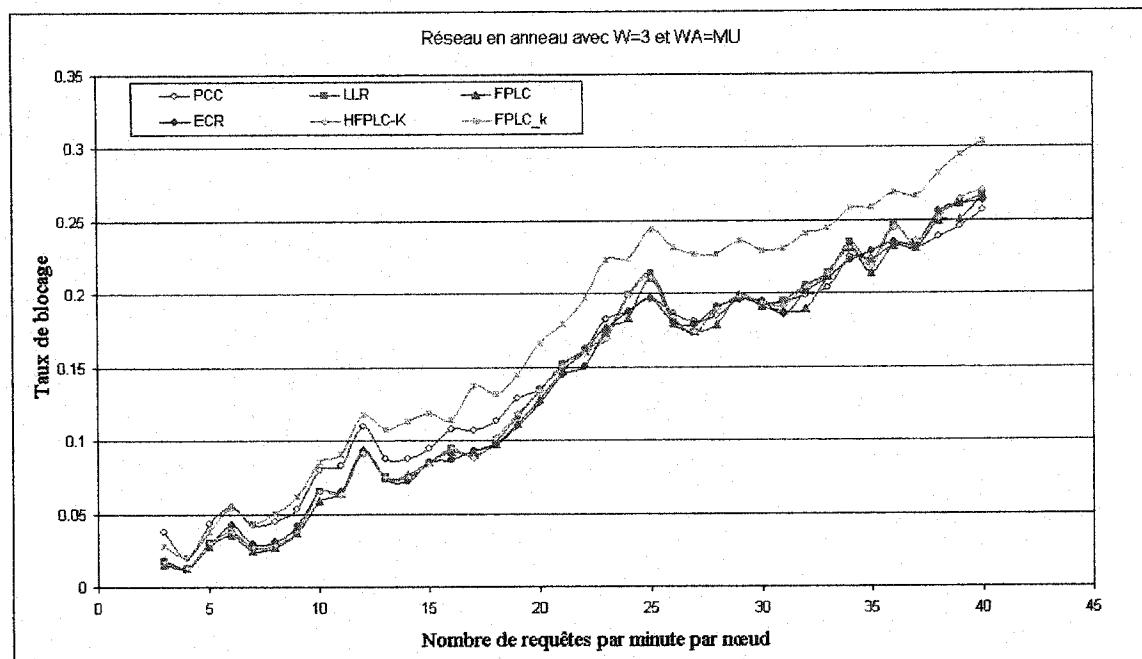
Par souci de simplification, nous analyserons par la suite les résultats obtenus avec des liaisons de trois et ensuite de six longueurs d'onde initiales. Pour le réseau avec des liaisons comportant trois longueurs d'onde, les résultats des différents algorithmes de routage varient comme illustré aux Figures 4.9 à 4.12. Globalement, on note un taux de blocage moins élevé que dans le cas précédent, avec des plafonds variant entre 17 et 35%. On note également une mauvaise performance de la part du FPLC- $k$  qui reste le pire algorithme quelle que soit la stratégie d'affectation de longueurs d'onde proposée. La topologie particulière du réseau implique que les longueurs des chemins entre deux paires de noeuds varient de 1 à N-1 (c'est-à-dire 7) ; dans ce cas, l'analyse de deux liaisons pour effectuer une décision s'avère insuffisante. Par contre, notre algorithme, le HFPLC- $k$ , qui analyse également deux liaisons, offre dans la même situation de très bons résultats car il évalue les goulets d'étranglement de chaque chemin. De même, quand le nombre de requêtes qui arrivent aux noeuds ne dépasse pas 20, on constate que nos deux algorithmes de routage proposés rivalisent très bien avec les autres. L'algorithme du ECR se démarque d'ailleurs des autres avec la stratégie d'affectation du *Least Used* où le taux de blocage du réseau est parfois meilleur de 1% à ceux des autres algorithmes de routage. Le LLR et le FPLC enregistrent dans l'ensemble des performances moyennes, particulièrement dans le cas des stratégies d'affectation *random* et *least used*. Par ailleurs, l'algorithme du plus court chemin offre de très bonnes performances dues à la longueur réduite des plus courts chemins et au temps de connexion très peu élevé des requêtes acceptées.



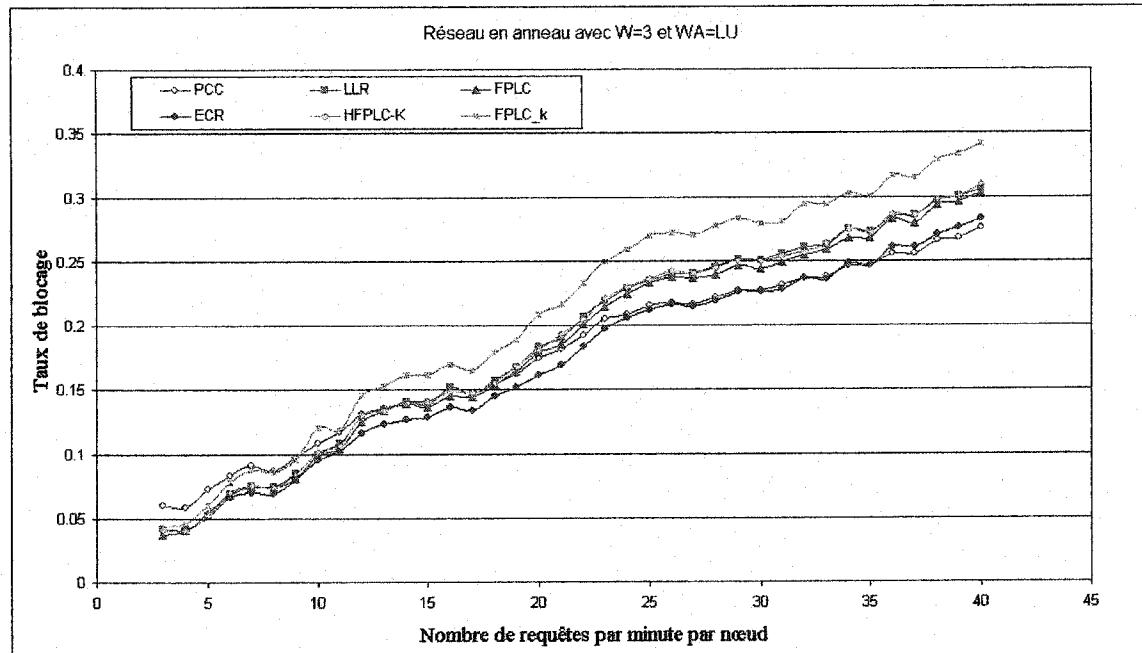
**Figure 4.8 Évaluation du taux de blocage en fonction du trafic pour  $W=1$**



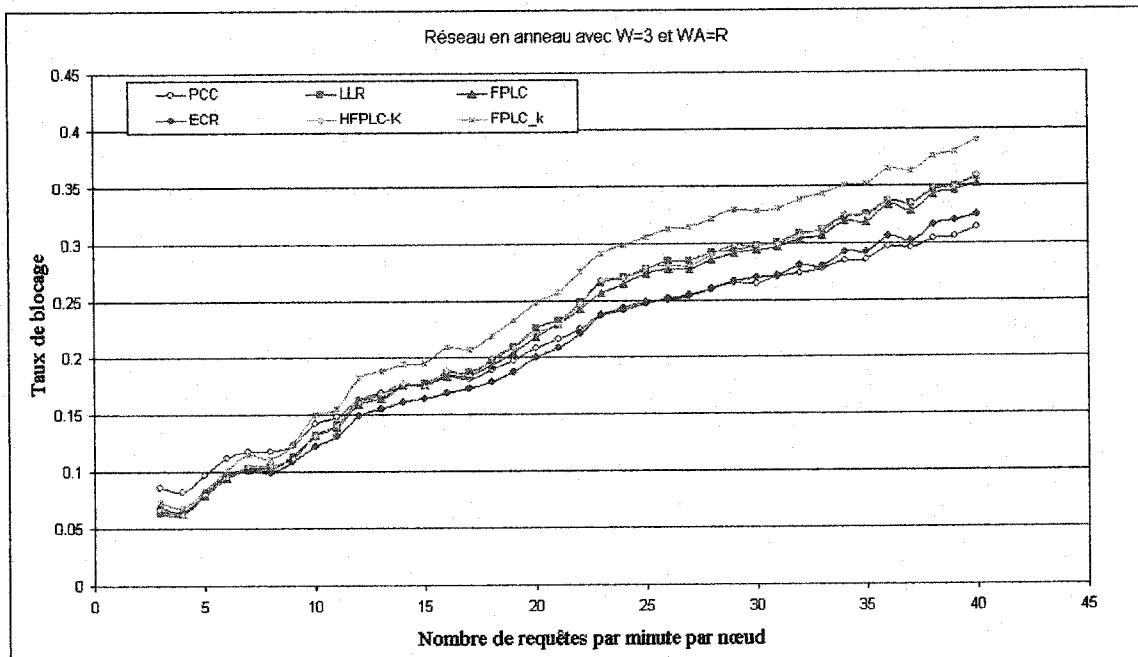
**Figure 4.9 Évaluation du taux de blocage en fonction du trafic pour  $W=3$  et  $WA=First\ Fit$**



**Figure 4.10 Évaluation du taux de blocage en fonction du trafic pour  $W=3$  et  $WA=$ Most Used**



**Figure 4.11 Évaluation du taux de blocage en fonction du trafic pour  $W=3$  et  $WA=$ Least Used**



**Figure 4.12 Évaluation du taux de blocage en fonction du trafic pour W=3 et WA=Random**

Le réseau en anneau avec des liaisons comportant chacune six longueurs d'onde initiales présentent de meilleurs résultats et un taux moins élevé de blocage dans l'ensemble, à cause du nombre de requêtes admissibles par chemin plus grand. Tous les algorithmes de routage présentent un taux de blocage inférieur à 10% dans le pire cas montré aux Figures 4.13 à 4.16. Cela nous laisse présumer qu'un nombre initial de longueurs d'onde par liaison plus grand aurait suffi à satisfaire tout le trafic d'entrée dans le réseau. Cependant, contrairement aux résultats présentés auparavant, l'algorithme du plus court chemin n'offre pas toujours de bons résultats par rapport aux autres algorithmes. Le FPLC-*k* reste le pire algorithme de routage ; par contre, le FPLC améliore nettement ses performances avec les stratégies d'affectation *first fit* et *most used*. Le ECR offre pour sa part des résultats similaires aux précédents et prouve son efficacité lorsque combiné aux algorithmes d'affectation *least used* ou *random*. Grâce à sa combinaison avec ces derniers, la charge du réseau est répartie sur toutes les liaisons

puisque plus certaines longueurs d'onde disponibles sont utilisées, plus la probabilité est grande que les chemins utilisant cette longueur d'onde finissent par être congestionnés. Quant au HFPLC- $k$ , il voit ses performances se dégrader en combinaison avec le *least used* et le *random*, car il peut souvent arriver que le chemin choisi, en respectant ses propres critères, ne possède pas la longueur d'onde la moins utilisée de libre.

On constate donc globalement que, dans le cas du réseau en anneau, les algorithmes de routage offrent un taux de blocage moins élevé avec un nombre initial de longueurs d'onde plus grand. Par ailleurs, bien qu'utilisant le même nombre de liaisons, en l'occurrence deux, pour faire le choix du chemin, l'algorithme du FPLC- $k$  et celui du HFPLC- $k$  que nous avons proposé ne donnent pas les mêmes résultats. Notre algorithme est plus efficace par le fait qu'il identifie les liaisons les plus congestionnées de chaque chemin avant de faire son choix définitif. Les stratégies d'affectation de longueurs d'onde *first fit* et *most used* combinées à tous les algorithmes de routage donnent de bons résultats alors que seuls le ECR et le plus court chemin offrent de meilleures performances lorsqu'ils sont utilisés avec le *least used* et le *random*.

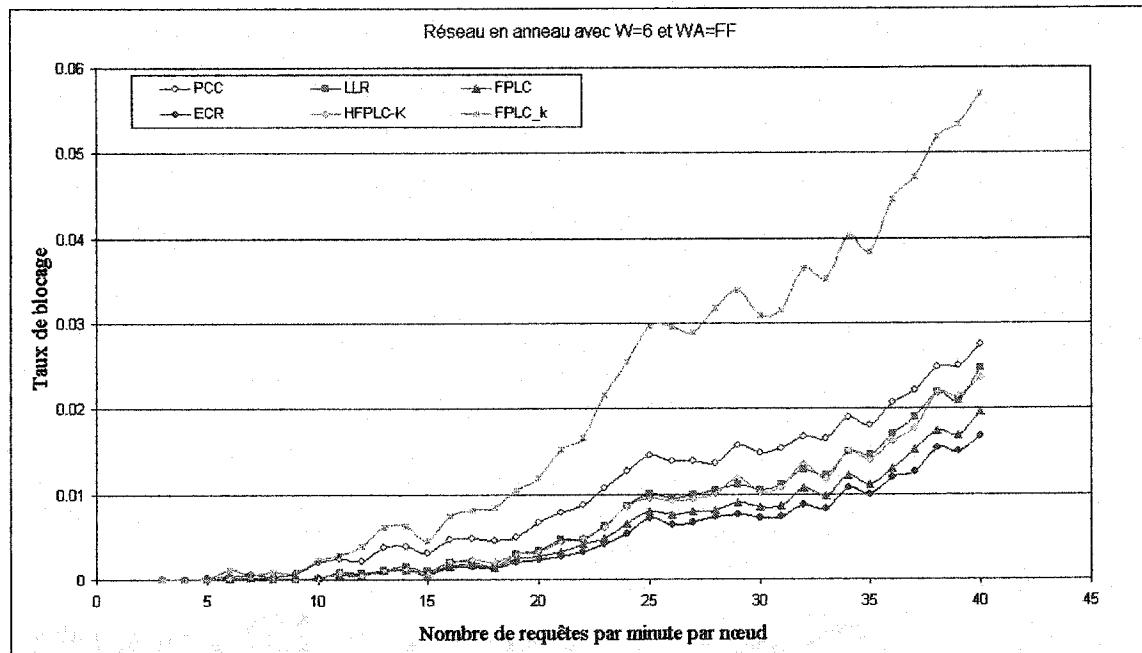


Figure 4.13 Évaluation du taux de blocage en fonction du trafic  
pour W=6 et WA= First Fit

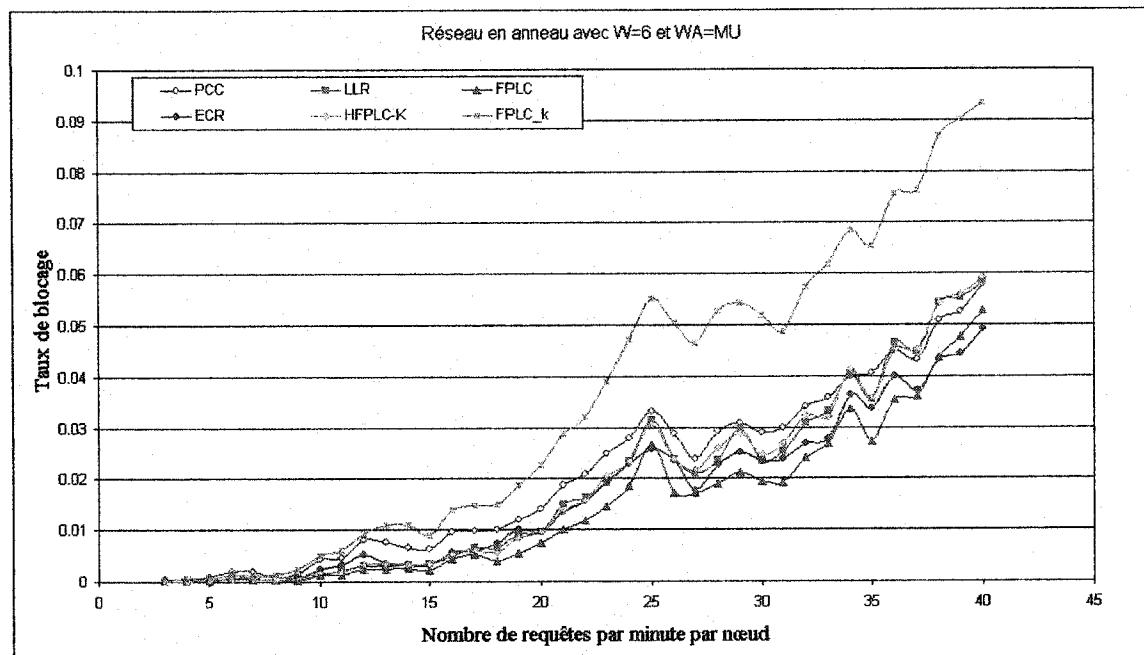


Figure 4.14 Évaluation du taux de blocage en fonction du trafic pour  $W=6$  et  $WA=$  Most Used

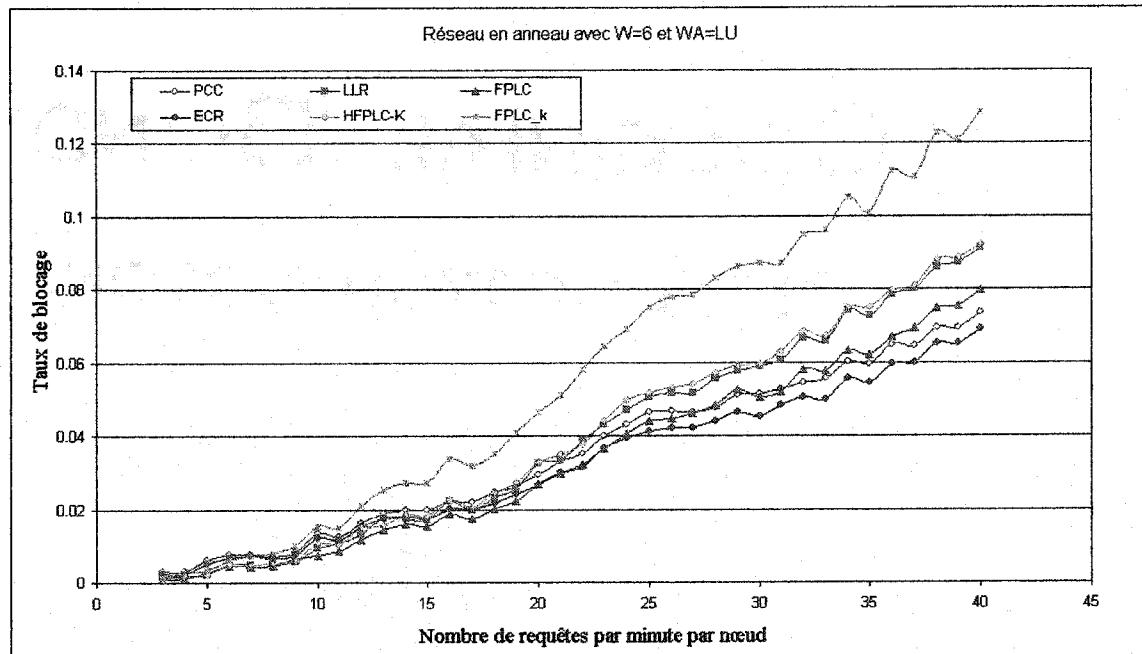
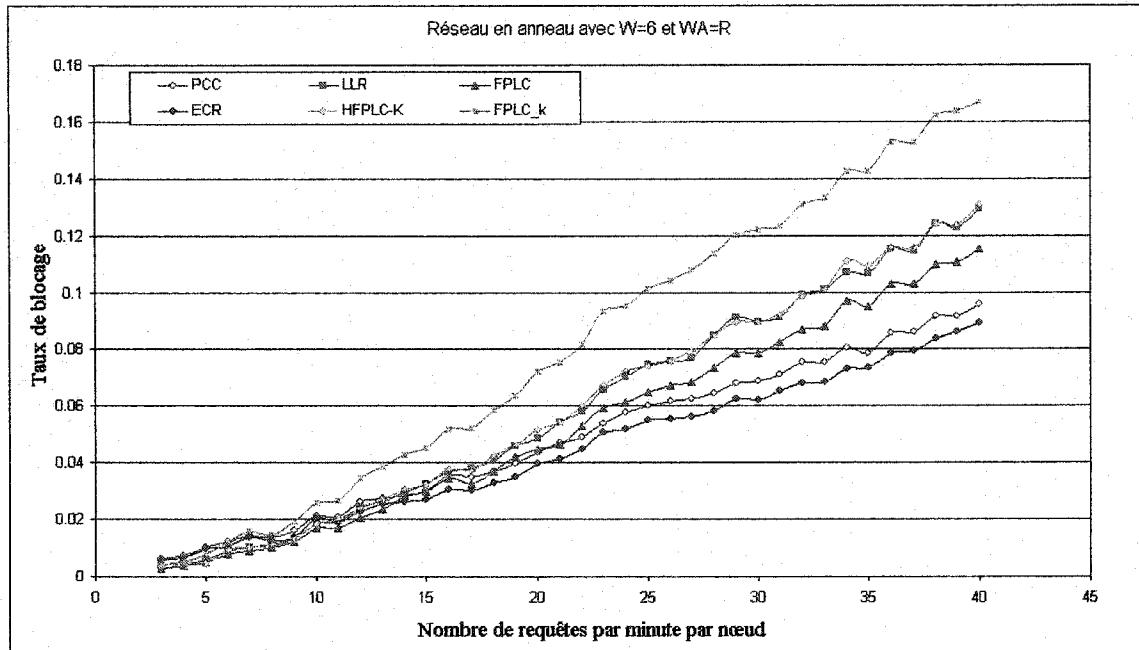


Figure 4.15 Évaluation du taux de blocage en fonction du trafic pour  $W=6$  et  $WA=$  Least Used



**Figure 4.16 Évaluation du taux de blocage en fonction du trafic pour W=6 et WA= Random**

#### 4.3.1.2 Évaluation du temps de réponse

Les différentes variations du nombre de longueurs d'onde initial par liaison dans le réseau n'ont pas changé les résultats observés par tous les algorithmes de routage. Ainsi, on constate à la Figure 4.17 que le ECR et le HFPLC-*k* sont les algorithmes les plus coûteux en temps de calcul. Le cas particulier du ECR va se confirmer par la suite pour les autres configurations étudiées puisque les opérations flottantes réalisées par cet algorithme consomment beaucoup de temps au niveau du processeur. Quant au HFPLC-*k*, le temps supplémentaire mis pour déterminer les liaisons les plus congestionnées avant d'effectuer un choix justifie la différence de l'ordre de 50 ms observée par rapport aux autres algorithmes. L'algorithme du plus court chemin a évidemment le temps de réponse le plus court puisque chaque fois qu'une requête arrive, le chemin à choisir est calculé et connu d'avance. Tous les autres algorithmes de routage offrent des performances plutôt moyennes et le reste des résultats est présenté en annexe de ce document.

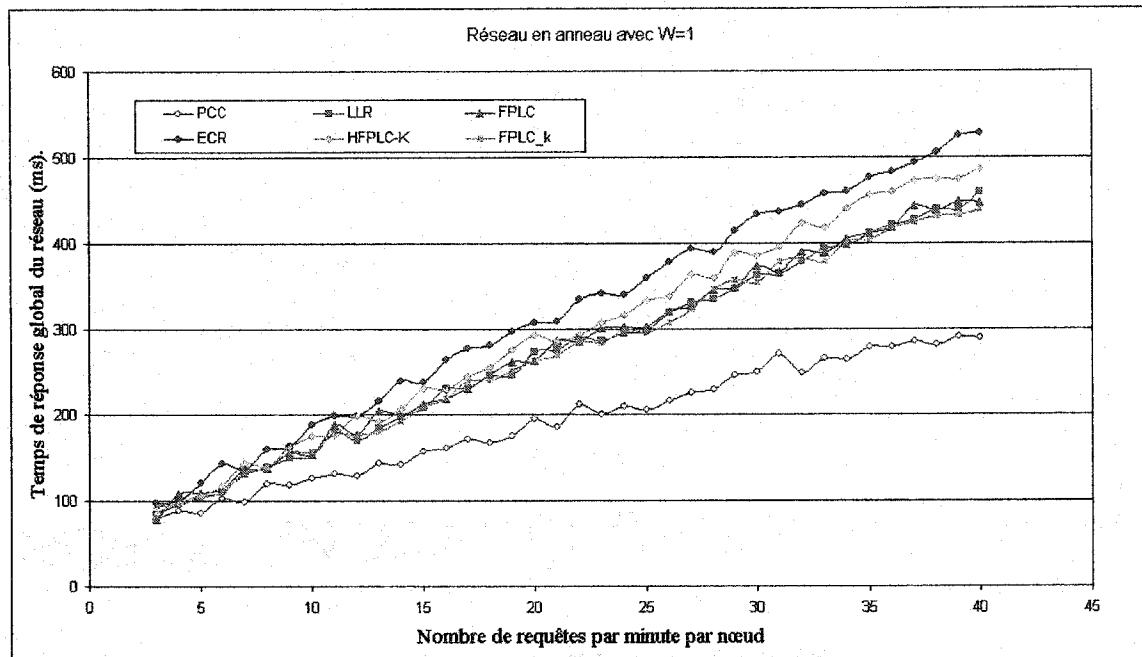


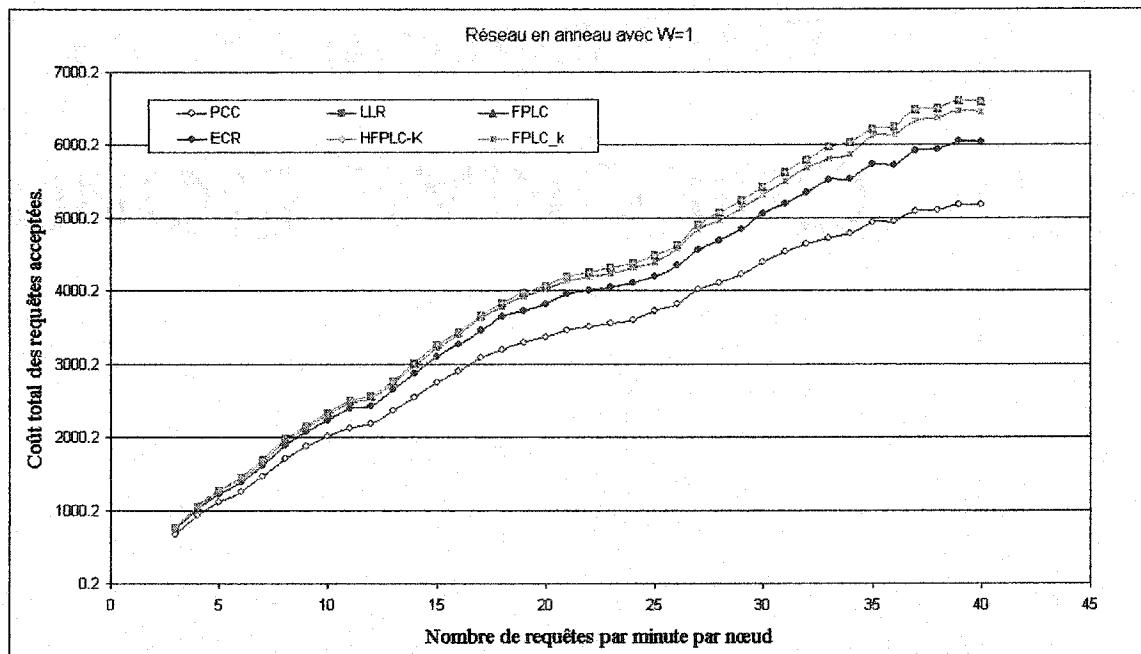
Figure 4.17 Évaluation du temps de réponse en fonction du trafic pour  $W=1$

#### 4.3.1.3 Évaluation du coût total

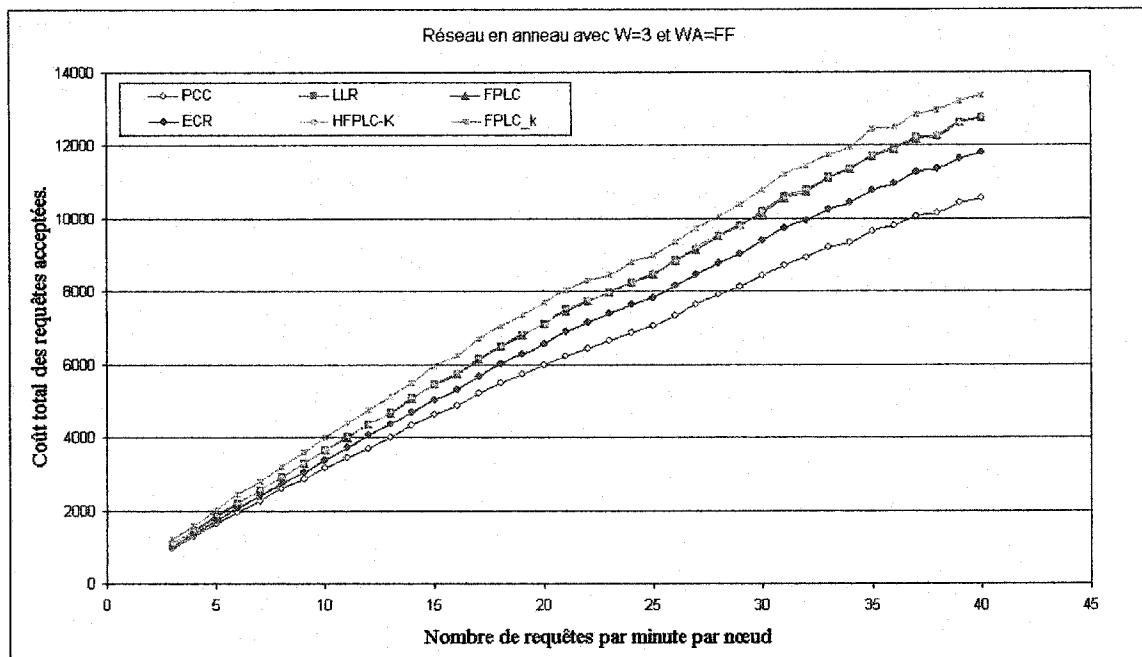
Les Figures 4.18 à 4.20 reportent les résultats obtenus pour le cas des liaisons avec une et trois longueurs d'onde initiales par liaison. Dans le premier cas, on constate naturellement que l'algorithme du plus court chemin offre les meilleurs coûts puisqu'il base le choix des chemins uniquement sur le critère du coût. Le second meilleur algorithme est le ECR qui utilise également une variable de coût dans son processus de décision. Les autres algorithmes de routage ont des coûts semblables, ce qui est normal si on se rapporte aux taux de blocage enregistrés précédemment avec cette architecture.

Les algorithmes de routage offrent des performances similaires dans le cas de trois longueurs d'onde par liaison, même si on voit les résultats du FPLC- $k$  se dégrader. Cela s'explique par le fait que, lorsqu'il existe deux chemins possibles entre deux nœuds voisins avec l'un qui comporte par exemple une seule longueur d'onde libre et l'autre qui en possède deux de libres sur ces deux premières liaisons, l'algorithme du FPLC- $k$  va choisir le deuxième chemin. Pourtant, le premier chemin aurait coûté moins cher et aurait respecté le critère de continuité de longueur d'onde. Le HFPLC- $k$  aurait pu

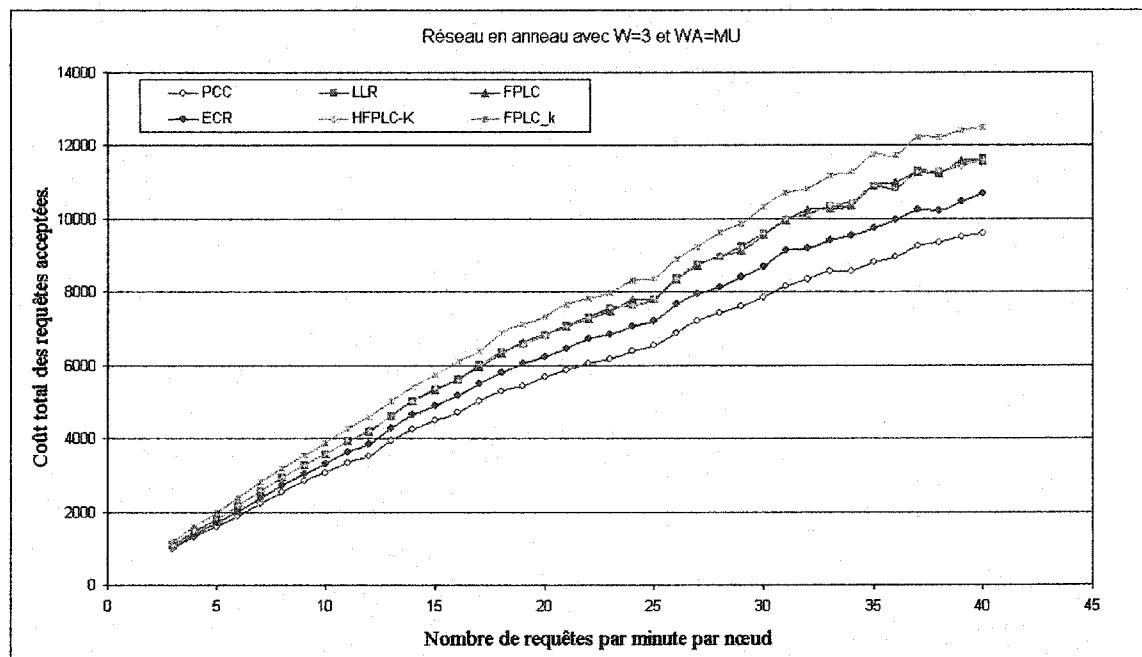
sombrer dans les mêmes erreurs mais les risques s'amenuisent avec la probabilité de trouver une longueur d'onde de moins dans celles qui sont disponibles sur le second chemin, par l'analyse des liaisons les plus chargées. On remarque par ailleurs que les coûts des algorithmes FPLC, LLR et HFPLC-*k* restent pratiquement identiques selon chaque stratégie d'affectation de longueurs d'onde utilisée. Les autres résultats obtenus pour le coût total suivent le même comportement et sont annexés au présent document pour fin de consultation.



**Figure 4.18 Évaluation du coût total des requêtes en fonction du trafic pour W=1**



**Figure 4.19** Évaluation du coût total des requêtes en fonction du trafic pour  $W=3$  et  $WA=First\ Fit$



**Figure 4.20** Évaluation du coût total des requêtes en fonction du trafic pour  $W=3$  et  $WA=Most\ Used$

### 4.3.2 Réseau maillé

Le réseau maillé étudié au cours de notre évaluation est illustré à la Figure 4.21 et comprend 13 nœuds reliés par 56 liaisons. Les coûts des liaisons varient entre 1 et 2 en fonction de leur rattachement ou non au nœud principal désigné ici par  $n0$ . Les variables initiales entrées dans le système restent identiques à celles du réseau en anneau. Une valeur de  $k=3$  a été utilisée pour les algorithmes HFPLC- $k$  et FPLC- $k$ . Les différents résultats obtenus sont présentés dans cette section.

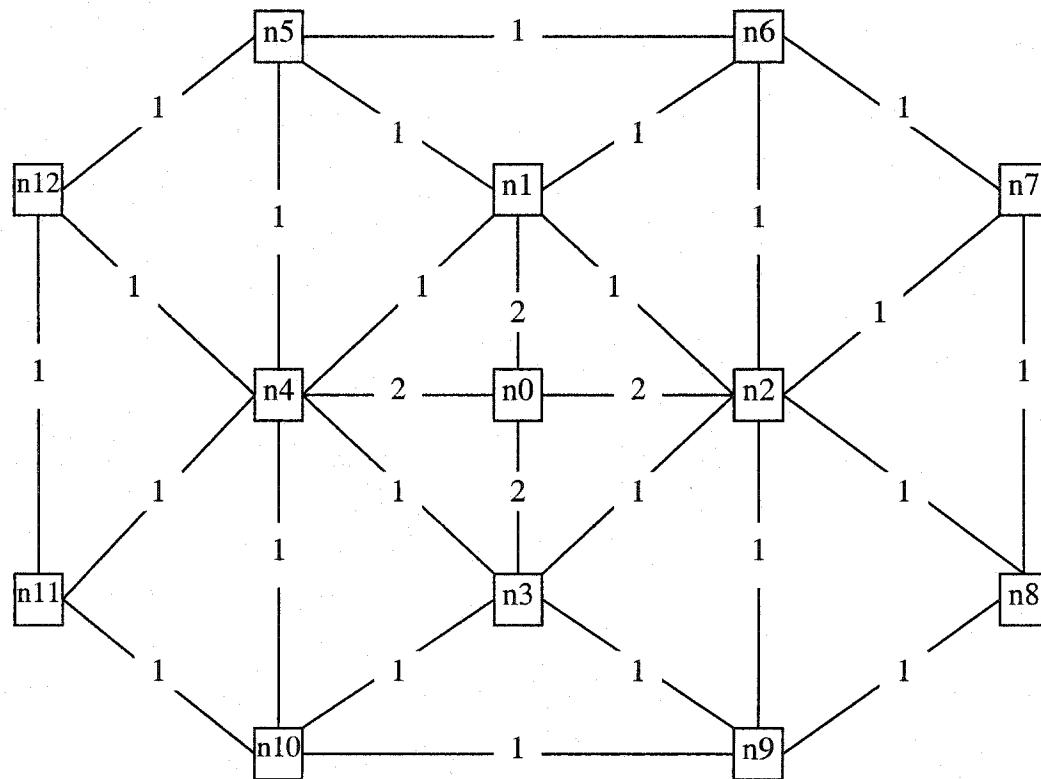


Figure 4.21 Topologie du réseau optique maillé

#### 4.3.2.1 Évaluation du taux de blocage

Les Figures 4.22 à 4.24 montrent les résultats obtenus. On constate que le ECR offre des performances plutôt moyennes alors que le HFPLC- $k$  donne les mêmes résultats que les autres algorithmes pour le cas d'une seule longueur d'onde par liaison. Par contre, quand le nombre de longueurs d'onde passe à 3, les deux algorithmes que

nous proposons rivalisent bien avec les autres et le HFPLC- $k$  bat le LLR quand le trafic devient plus important. Le FPLC- $k$  se montre très performant en donnant les mêmes résultats que le FPLC et ce, avec trois longueurs d'onde analysées. Les mêmes tendances s'observent quand on utilise six longueurs d'onde par fibre et on constate que le ECR est beaucoup moins adapté à la topologie de ce réseau. La meilleure stratégie d'affectation de longueur d'onde qui se combine le mieux à nos algorithmes reste le *first fit*. Les résultats obtenus avec les autres stratégies sont mis en annexe.

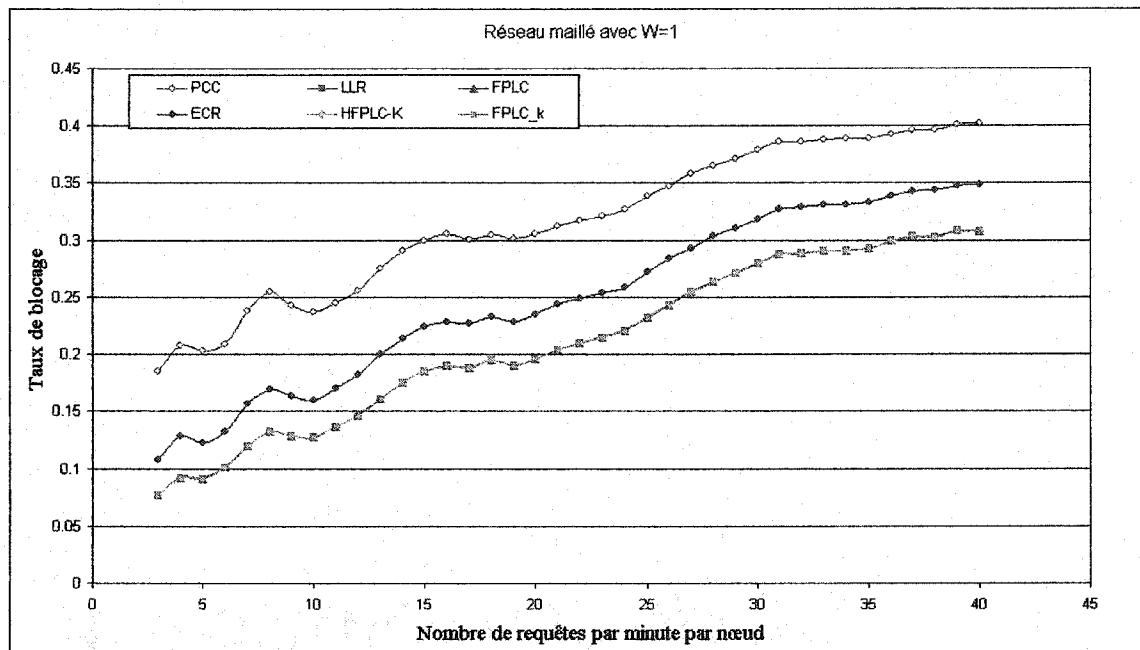


Figure 4.22 Évaluation du taux de blocage en fonction du trafic pour  $W=1$

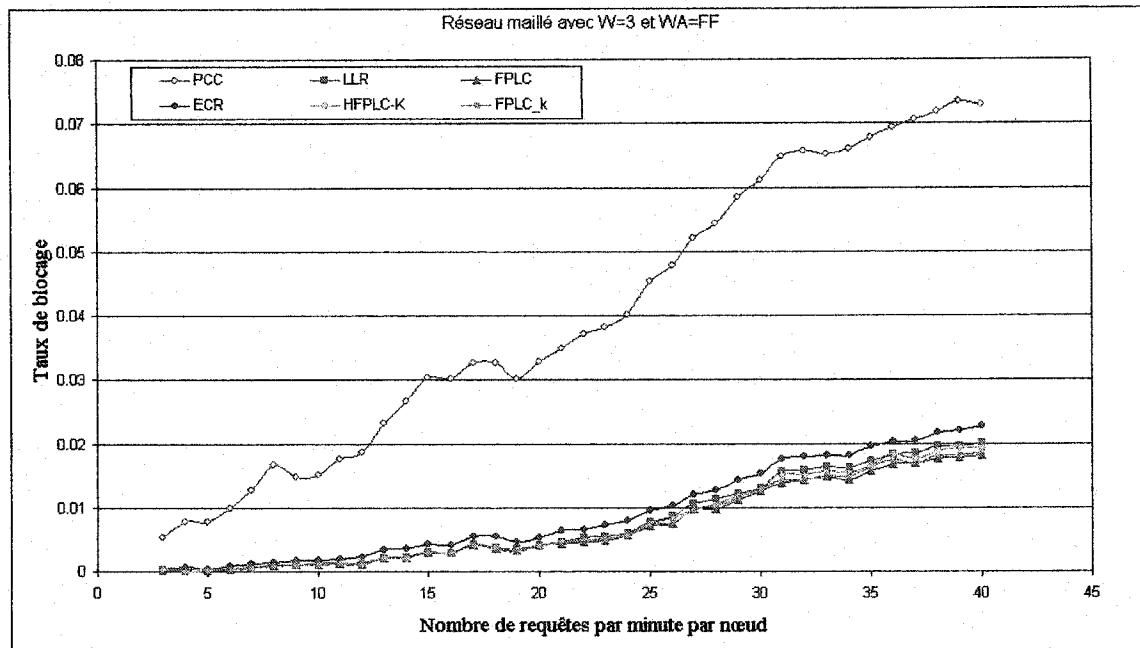


Figure 4.23 Évaluation du taux de blocage en fonction du trafic pour  $W=3$  et  $WA=First\ Fit$

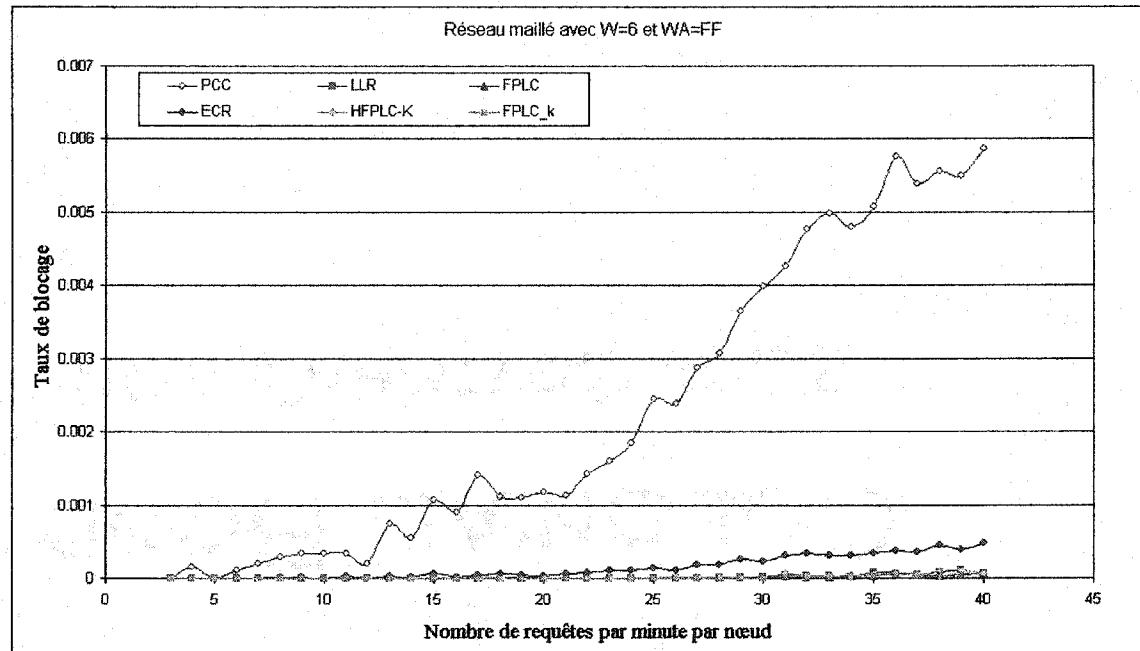


Figure 4.24 Évaluation du taux de blocage en fonction du trafic pour  $W=6$  et  $WA=First\ Fit$

L'évaluation du temps de réponse et du coût total donne des résultats identiques aux précédents observés dans la section 4.3.1 pour les raisons que nous avons citées :

- les opérations engendrées dans le ECR en font un algorithme coûteux en temps de calcul ;
- le HFPLC- $k$  demeure le deuxième algorithme le moins rapide car il effectue des opérations de tri supplémentaires.

### 4.3.3 Réseau ARPANET

Le seul réseau non standard étudié a été celui de ARPANET dont la topologie est représentée à la Figure 4.4 de ce chapitre. On y retrouve 14 nœuds et 44 liaisons dont les coûts ont été affectés de manière aléatoire, tout en veillant à ce que deux liaisons unidirectionnelles reliant la même paire de noeuds ait le même coût. Le taux d'arrivée initial des requêtes à chaque nœud a été fixé à 4 au lieu de 3, comparativement aux réseaux étudiés ci-dessus, mais les autres variables initiales restent identiques. Nous analysons ici les résultats de la simulation.

#### 4.3.3.1 Évaluation du taux de blocage

Le taux de blocage est très élevé dans le cas de l'existence d'une seule longueur d'onde par liaison, comme le montre la Figure 4.25. Les algorithmes de routage dynamique sauf le plus court chemin rejettent 60 à 70% des requêtes lorsque ces dernières augmentent. Le plus court chemin offre les pires performances avec une différence de 4% environ par rapport aux autres algorithmes. Le taux élevé de blocage s'explique tout simplement par le grand *diamètre* du réseau et l'insuffisance de longueurs d'ondes continues pour acheminer les requêtes. Les stratégies d'affectation de longueurs d'onde n'ont pas ici non plus d'influence sur le taux de blocage puisque le choix ne se fait que par rapport à une seule longueur d'onde par liaison.

Pour le même réseau avec trois longueurs d'onde par liaison, les taux de blocage diminuent considérablement et montent jusqu'à 54% pour tous les algorithmes, excepté celui du plus court chemin. On constate sur les Figures 4.26 et 4.27 que ce dernier est

largement dépassé par tous les autres algorithmes. Par ailleurs, le HFPLC-*k* offre de meilleurs résultats que le ECR mais reste comparable au LLR et au FPLC-*k*. Ce dernier le dépasse légèrement de 1 ou 2%. Quant au FPLC, c'est le meilleur algorithme quelle que soit la stratégie d'affectation de longueur d'onde car, contrairement aux autres algorithmes, il tient compte de la continuité de longueur d'onde sur toutes les liaisons de chaque chemin potentiel avant d'effectuer son choix. On note enfin la mauvaise performance du ECR en combinaison avec d'autres algorithmes d'affectation de longueurs d'onde, excepté le *first fit*.

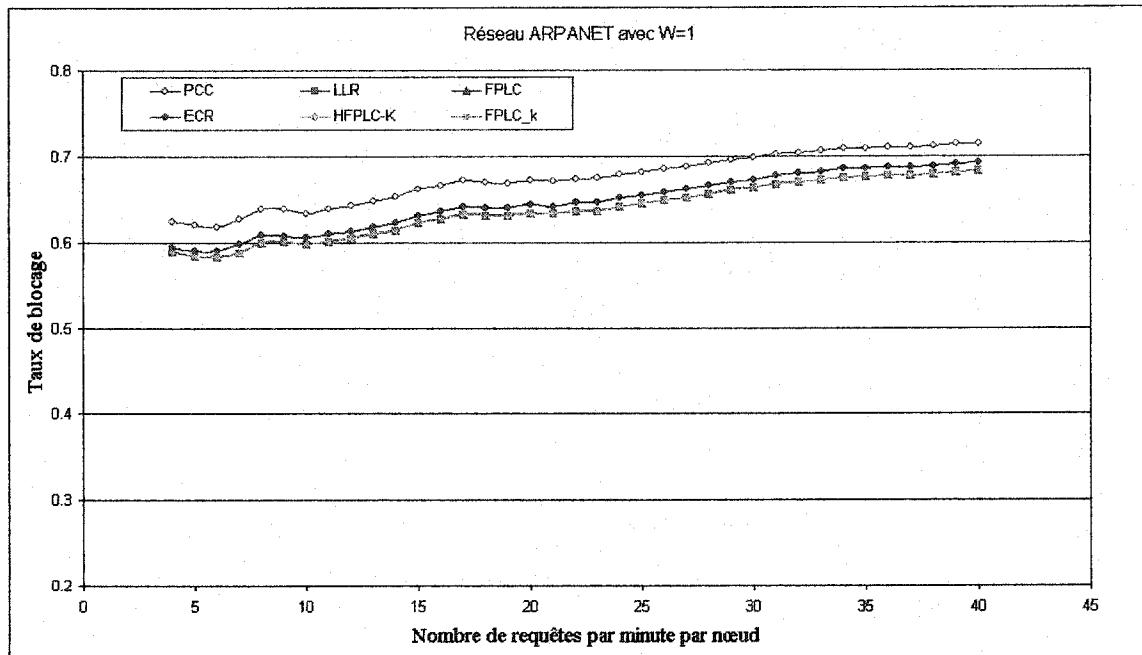
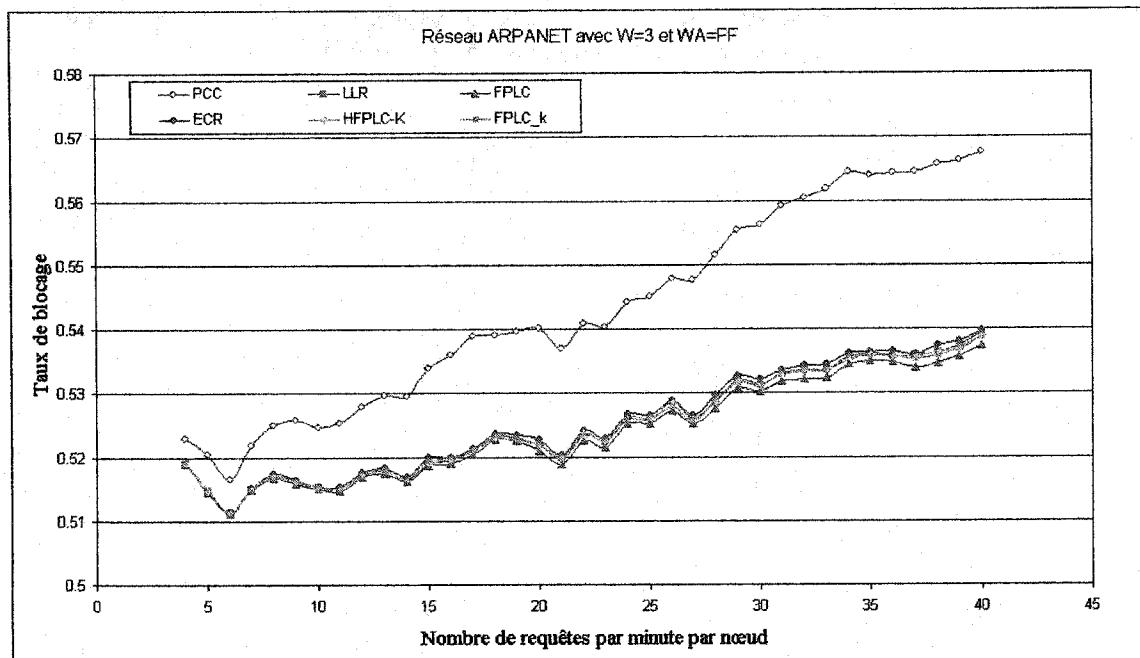
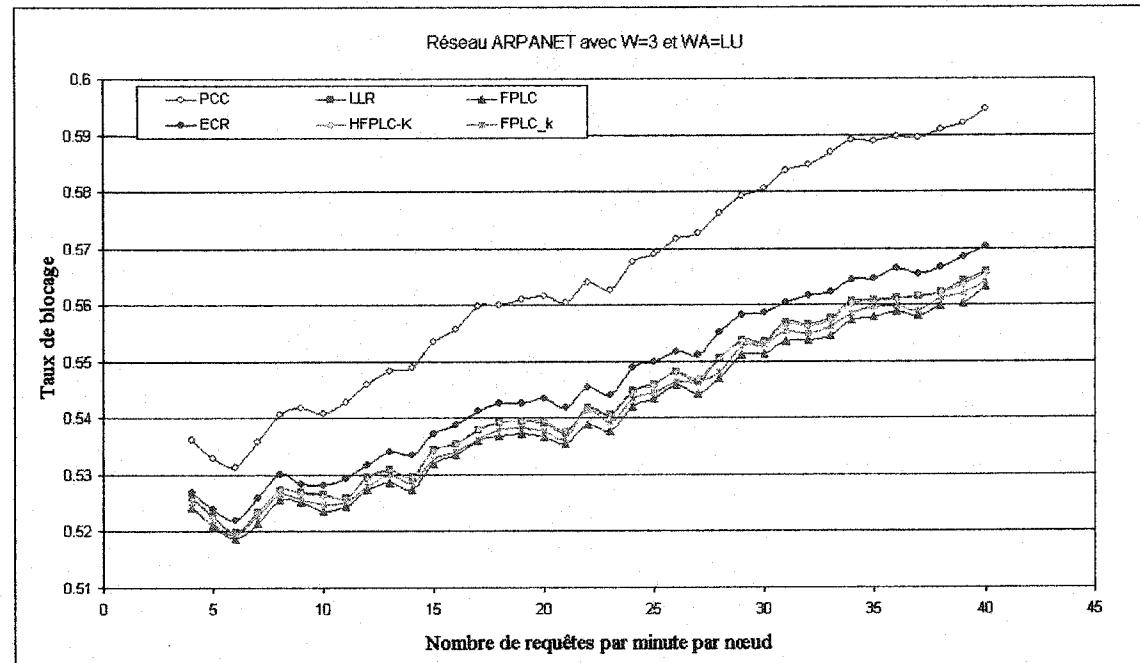


Figure 4.25 Évaluation du taux de blocage en fonction du trafic pour W=1



**Figure 4.26 Évaluation du taux de blocage en fonction du trafic pour  $W=3$  et  $WA=First\ Fit$**



**Figure 4.27 Évaluation du taux de blocage en fonction du trafic pour  $W=3$  et  $WA=Least\ Used$**

#### 4.3.3.2 Évaluation du temps de réponse

Comme pour les autres résultats observés précédemment à la même étape, nos deux algorithmes proposés sont les plus coûteux en temps de calcul. Cependant, dans le cas du réseau avec une seule longueur d'onde par liaison illustré à la Figure 4.28, seul le ECR connaît des variations sensibles de l'ordre d'une vingtaine de millisecondes par rapport aux autres algorithmes dynamiques. Le plus court chemin demeure le meilleur en temps de calcul et, grâce à l'écart mesuré entre lui et les autres algorithmes, on peut déterminer le temps affecté au choix du meilleur chemin pour chaque requête. Quand le nombre de longueurs d'onde augmente, on constate que le temps de calcul de chaque algorithme augmente très peu. Les courbes correspondantes se retrouvent en annexe de notre document.

Les performances moyennes de l'ECR et du HFPLC- $k$  peuvent être rehaussées par l'utilisation d'un processeur spécialisé en opérations flottantes et par l'implantation d'un algorithme de tri plus efficace.

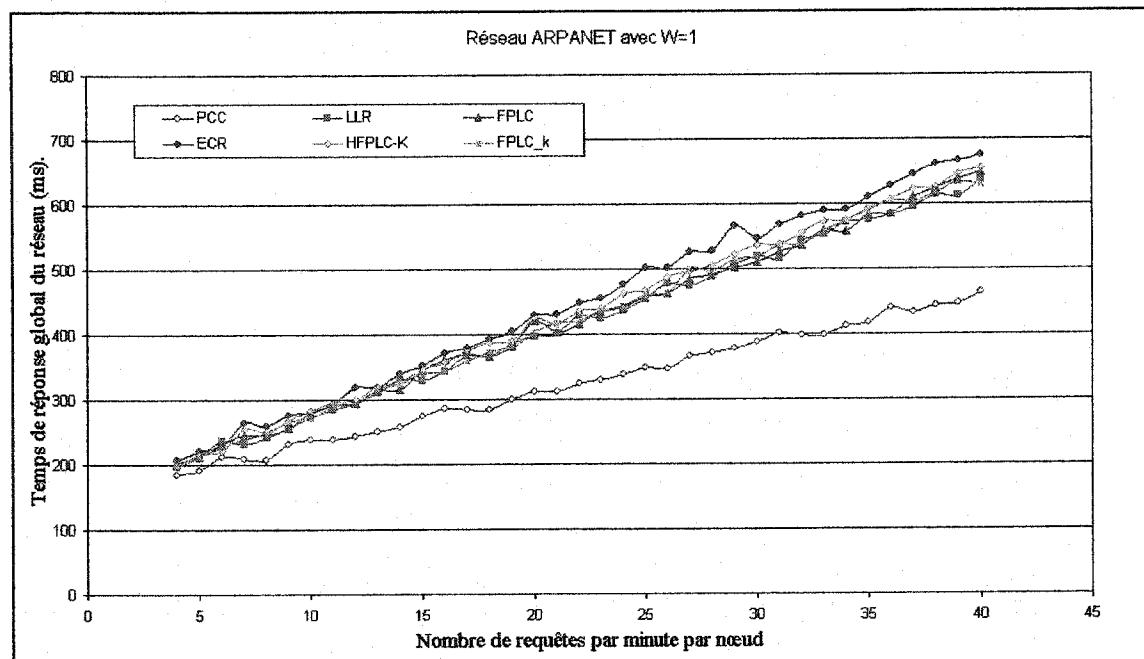


Figure 4.28 Évaluation du temps de réponse en fonction du trafic pour  $W=1$

#### 4.3.3.3 Évaluation du coût total

Nous remarquons ici que les coûts de tous les algorithmes, sauf celui du plus court chemin et du ECR, restent pratiquement identiques entre eux quel que soit le nombre de longueurs d'onde données aux liaisons du réseau. Le ECR et le PCC sont les moins coûteux puisqu'ils bloquent plus de requêtes que les autres, donc ils utilisent moins de liaisons au bout de l'expérimentation. Quant aux autres algorithmes, le fait d'avoir des coûts relativement uniformes nous prouve que l'algorithme du HFPLC- $k$  est capable d'effectuer les bons choix de chemins pour un ensemble de requêtes puisqu'il donne un coût global semblable au FPLC et au LLR.

À la fin de cette section de résultats, nous constatons donc l'efficacité de notre premier algorithme, le ECR, dans des réseaux de faible densité – les réseaux en anneau, alors que le HFPLC- $k$  se distingue en rivalisant avec le LLR et le FPLC- $k$  dans des réseaux plus larges et plus denses.

## CHAPITRE 5

### CONCLUSION

Dans ce mémoire, nous avons traité du problème de routage optique dynamique en proposant deux nouveaux algorithmes de routage basés sur des approches différentes. La première s'inspire d'une variable de congestion affectée à chaque chemin candidat, alors que la seconde tire sa source du FPLC- $k$  auquel nous avons ajouté un critère additionnel de congestion de liaison. Dans ce chapitre, nous faisons tout d'abord une synthèse de l'ensemble de nos travaux et de nos contributions essentielles. Par la suite, nous discutons des limitations de nos réalisations et nous terminons par une ouverture vers des indications de recherches futures susceptibles de compléter nos travaux.

#### 5.1 Synthèse des travaux

Le problème de routage dynamique dans les réseaux tout-optiques consiste à satisfaire les requêtes de connexion qui arrivent aux nœuds et ce, en acheminant les données le long d'un chemin optique utilisant une seule longueur d'onde. À cause du nombre limité de nœuds et de longueurs d'onde portées par les liaisons dans le réseau, ce problème s'avère difficile (NP-complet). Les algorithmes utilisés à ce jour ont été évalués et présentent certaines limitations que nous avons essayées de combler au cours de ce mémoire. Pour ce faire, nous avons utilisé deux approches pour estimer la congestion dans les réseaux optiques. La première a été proposée par nous et utilise une fonction qui estime la congestion sur chaque chemin afin de prendre une décision de routage. La deuxième se base sur les travaux effectués par Li et Soman (1999) en introduisant un critère additionnel de liaison congestionnée. Nous en avons alors tiré deux algorithmes dont la complexité est parfois plus élevée que celle de leurs prédecesseurs dans les pires scénarios.

L'implémentation de tous les algorithmes de routage antérieurs et des deux que nous proposons nous a permis de concevoir l'architecture globale des réseaux optiques.

Nous avons utilisé des nœuds pourvus d'une quantité de mémoire limitée mais suffisante pour échanger des messages avec un serveur principal et traiter les données qui les traversent. Le serveur principal est considéré comme le maître d'œuvre du réseau et se charge de coordonner les actions de tous les routeurs actifs du réseau. Le trafic qui entre dans le réseau est aléatoirement généré avec un certain taux d'arrivée. Une stratégie de sauvegarde des données en cas de panne a été également instaurée, en l'occurrence la restauration de chemin. Les résultats fournis par les différents algorithmes de routage et d'affectation de longueurs d'onde ont été ensuite recueillis dans des fichiers que nous avons analysés.

Pour évaluer les performances de nos deux algorithmes, nous avons réalisé une analyse de complexité théorique ainsi qu'un ensemble de simulations qui nous ont permis de les confronter aux autres algorithmes. Les résultats générés lors des séances de simulation nous ont aidé à tester l'influence de l'architecture du réseau et de la stratégie d'affectation des longueurs d'onde sur le comportement global du réseau. Nous en sommes alors arrivés aux conclusions suivantes :

- Pour le taux de blocage, les performances de nos deux algorithmes ont varié en fonction de la topologie du réseau et de la stratégie d'affectation de longueurs d'onde auxquelles ils ont été combinés. Ainsi, le ECR a été très bon avec le réseau en anneau alors qu'il a connu de piétres performances avec le réseau ARPANET. Par contre, le HFPL- $k$  est resté stable dans ses performances quel que soit le réseau considéré et génère la plupart du temps des résultats égaux ou supérieurs à ceux du LLR et du FPLC- $k$ .
- Le temps d'exécution des algorithmes ECR et HFPLC a été dans l'ensemble légèrement supérieur aux algorithmes de référence, à cause du nombre élevé de calcul flottant pour le premier et de la complexité de l'algorithme de tri utilisé par le second.
- Le coût total des requêtes acceptées a permis pour sa part de constater que parfois, malgré une légère différence en taux de blocage avec les autres

algorithmes, nos algorithmes ont souvent choisi des chemins avec des coûts relativement faibles.

Nos deux algorithmes offrent donc en général de bonnes performances et nous notons la parfaite performance du ECR dans un réseau à faible densité comme celui en anneau, alors que le HFPL- $k$  se révèle à la fois mieux adapté aux réseaux à moyenne densité. La légère différence notée dans les temps de calcul comparativement aux autres algorithmes peut se résoudre facilement avec les solutions que nous avons proposées auparavant. La complexité de nos deux algorithmes peut donc être améliorée pour mieux répondre aux besoins spécifiques de délais dans les réseaux optiques. Enfin, on constate que les stratégies d'affectation *first fit* donne de bons résultats en général avec nos deux algorithmes ainsi que celui du *most used*. L'unique situation dans laquelle les stratégies *least used* et *random* ont été performantes est celle du réseau en anneau lorsqu'elles ont été combinées avec le ECR.

Cependant, nos algorithmes ne sont pas parfaits et comportent quelques limites à souligner.

## 5.2 Limitations des travaux

En dépit des bons résultats obtenus, les performances de nos deux algorithmes dépendent en partie de la conception de l'architecture du réseau. Ainsi, nous considérons un serveur principal dans notre réseau sans tenir compte des mécanismes de communication entre ce dernier et l'ensemble des nœuds du réseau. De plus, aucun serveur redondant n'a été implémenté, donc le réseau peut être mis hors service si le serveur principal recevait une attaque. Nous n'avons pas établi de protocole de réservation des longueurs d'onde le long des chemins choisis mais nous utilisons le critère temporel pour garantir que les événements se déroulent dans leur ordre chronologique d'occurrence. Par ailleurs, les données transmises lors des établissements de connexion ne sont pas des données réelles et étaient d'un seul type, à savoir des chaînes de caractère. Les données telle que la voix ou la vidéo n'ont pas été considérées.

En outre, nous n'avons pas eu l'opportunité d'implanter nos algorithmes dans un vrai réseau optique pour étudier leurs performances. Toutes les composantes réelles de réseaux optiques à savoir les commutateurs, les répéteurs, etc., n'ont pas été prises en compte et l'intégration de leurs pannes éventuelles dans les scénarios étudiés n'a pas été réalisée. Les différentes stratégies de restauration n'ont pas été non plus étudiées car elles n'étaient pas toutes adaptées à notre implémentation.

Toutefois, puisque ces remarques s'appliquent à l'ensemble des algorithmes de routage implémentés, la comparaison des résultats obtenus par nos deux algorithmes avec celle des autres algorithmes de routage dynamique reste entièrement valable.

Les performances de nos deux algorithmes en terme de temps d'exécution peuvent être améliorées si on utilise les ressources adéquates dans le cas du ECR ou en utilisant un meilleur algorithme de tri dans celui du HFPLC- $k$ .

### 5.3 Indications de recherches futures

L'approche basée sur l'existence d'un serveur central a été testée dans le cadre de ce travail. Il serait intéressant de développer une architecture de réseau optique dans laquelle tous les nœuds sont indépendants et se chargent d'effectuer de manière autonome toutes les opérations liées au routage afin de comparer les différentes performances qui s'en dégagent.

Dans le cas de l'algorithme du ECR, la fonction d'estimation de la congestion des chemins pourrait être raffinée par des outils analytiques afin d'améliorer le taux de blocage obtenu dans les réseaux denses. Le défi réside alors dans la découverte et l'utilisation de critères plus valables pour estimer la congestion. Quant au HFPLC- $k$ , il faudrait trouver un algorithme de tri des liaisons chargées moins complexe et plus rapide pour améliorer le temps d'exécution général de notre algorithme. Au lieu d'étudier la continuité de longueur d'onde sur les  $k$  premières liaisons comme le fait le FPCL- $k$  pour choisir le meilleur chemin, on pourrait également étudier cette même continuité, mais sur les  $k$  dernières liaisons de chaque chemin. Cette perspective pourrait permettre de diminuer la congestion aux nœuds de destination et leur permettre d'accommoder plus

de trafic. On pourrait aussi cibler certains nœuds essentiels dans le réseau qui auraient le privilège d'avoir une analyse sur les liaisons qui les entourent au lieu de choisir les  $k$  premières ou dernières liaisons de chaque chemin. Par ailleurs, de manière générale, le critère de qualité de service pourrait être introduit dans les décisions de routage pour nous permettre de mieux nous adapter aux conditions actuelles de transmission de données.

## BIBLIOGRAPHIE

Aneja Y. P., « Routing in Wavelength Routed Optical Networks », IEEE Workshop on High Performance Switching and Routing, May 2001, pp. 155-158.

Jue J.P. et Gaoxi X., « An adaptive routing algorithm for wavelength-routed optical networks with a distributed control scheme », Computer Communications and Networks Proceedings, 9<sup>th</sup> International Conference, Oct. 2000, pp.192-197.

Kodialam M. et Lakshman T. V., « Dynamic Routing of Locally Restorable Bandwidth Guaranteed Tunnels using Aggregated Link Usage Information », 20<sup>th</sup> Annual Joint Conference of the IEEE Computer and Communications Society, Vol. 1, Apr. 2001, pp. 376-385.

Kodialam M. et Lakshman T. V., « Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks », 20<sup>th</sup> Annual Joint Conference of the IEEE Computer and Communications Society, Vol. 1, Apr. 2001, pp. 358-366.

Lang J. P., Sharma V. et Varvarigos E. A., « An Analysis of Oblivious and Adaptive Routing in Optical Networks with Wavelength Translation », IEEE/ACM Trans. on Networking, vol. 9, No. 4, Aug. 2001, pp. 503-517.

Li L. et Somani A. K., « Dynamic Wavelength Routing Using Congestion and Neighbourhood Information », IEEE/ACM Trans. on Networking, vol. 7, No. 5, Oct. 1999, pp. 779-786.

Martins E. Q. V., Pascoal M. M. B., « A new implementation of Yen's ranking loopless paths algorithm », Oct. 2000.

[http://www.mat.uc.pt/~marta/Publicacoes/new\\_yen.ps.gz](http://www.mat.uc.pt/~marta/Publicacoes/new_yen.ps.gz)

Mohan G., Siva Ram Murthy C. et Soman A. K., « Efficient Algorithms for Routing Dependable Connections in WDM Optical Networks », IEEE/ACM Trans. on Networking, vol. 9, No. 5, Oct. 2001, pp. 553-566.

Mokhtar A. et Azizoglu M., « Adaptive Techniques for Routing and Wavelength Assignment in All-Optical Wans », IEEE Proceedings of the 39th Midwest Symposium on Circuits and Systems, vol. 3, Aug. 1996, pp. 1195-1198.

Mokhtar A. et Azizoglu M., « Adaptive Wavelength Routing in All-Optical Networks », IEEE/ACM Trans. on Networking, vol. 6, No. 2, Apr. 1998, pp. 197-206.

Mukherjee B., « WDM Optical Communications Networks: Progress and Challenges », IEEE Journal on Selected Areas in Communications, vol. 18, No. 10, October 2000, pp. 1810-1824.

Narula-Tam A., Lin P. J. et Modiano E., « Efficient Routing and Wavelength Assignment for Reconfigurable WDM Networks », IEEE Journal on Selected Areas in Commun., vol. 20, No. 1, Jan. 2002.

Poppe F., De Neve H. et Petit G. H., « Constrained Shortest Path First Algorithm for Lambda-Switched Mesh Optical Networks with Logical Overlay OCh/SP Rings », IEEE Workshop on High Performance Switching and Routing, May 2001, pp. 150-154.

Sridharan A., Sivarajan K. N., « Blocking in All-Optical Networks », 9<sup>th</sup> Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, March 2000, pp. 990-999.

Stern T. E. et Bala K., *Multiwavelength Optical Networks : A Layered Approach*, Éd. Prentice-Hall, 2000, 486 p.

Subramaniam S., Azizoglu M. et Somani A. K., « All-Optical Networks with Sparse Wavelength Conversion », *IEEE/ACM Transactions on Networks*, Vol. 4, No. 4, Aug. 1996, pp. 544-557.

Suurballe J. et Tarjan R. E., « A Quick Method for Finding Shortest Pairs of Disjoint Paths », *IEEE*, Vol. 14, No. 4, 1984, pp. 325-336.

Xu S., Li L. et Wang S., « Dynamic Routing and Assignment of Wavelength Algorithms in Multifiber Wavelength Division Multiplexing Networks », *IEEE Journ. on Selected Areas in Commun.*, vol. 18, No. 10, Oct. 2000, pp. 2130-2137.

Zang H., Jue J. P., Sahasrabuddhe L., Ramamurthy R. et Mukherjee B., « Dynamic Lightpath Establishment in Wavelength-Routed WDM Networks », *IEEE Commun. Mag.*, vol. 39, No. 9, Sept. 2001, pp. 100-108.

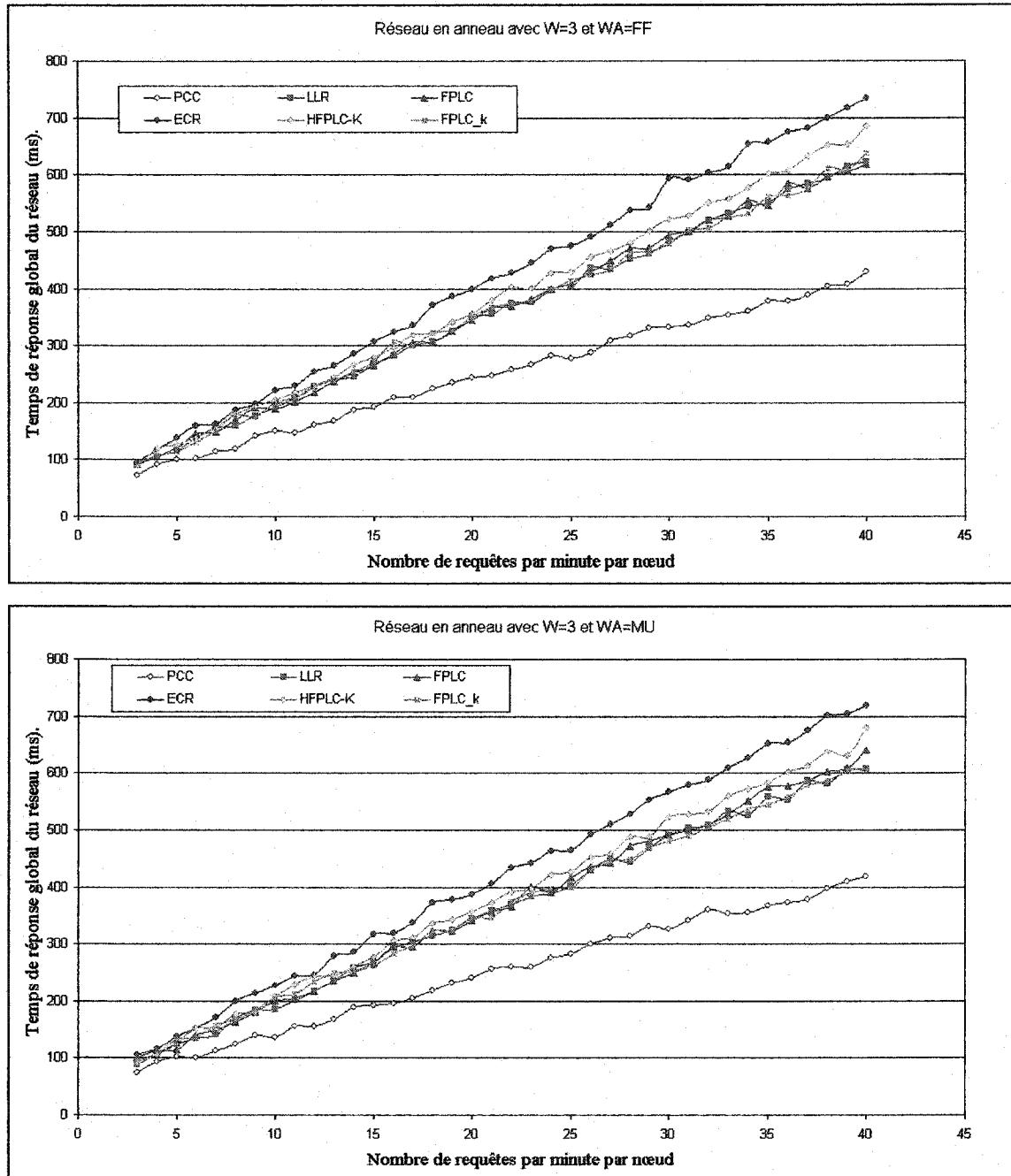
Zhou B. et Mouftah H. T., « Adaptive Least Loaded Routing for Multi-fiber WDM Networks Using Approximate Congestion Information », *Proceedings of IEEE ICC*, vol. 5, 2002, pp. 2745-2749.

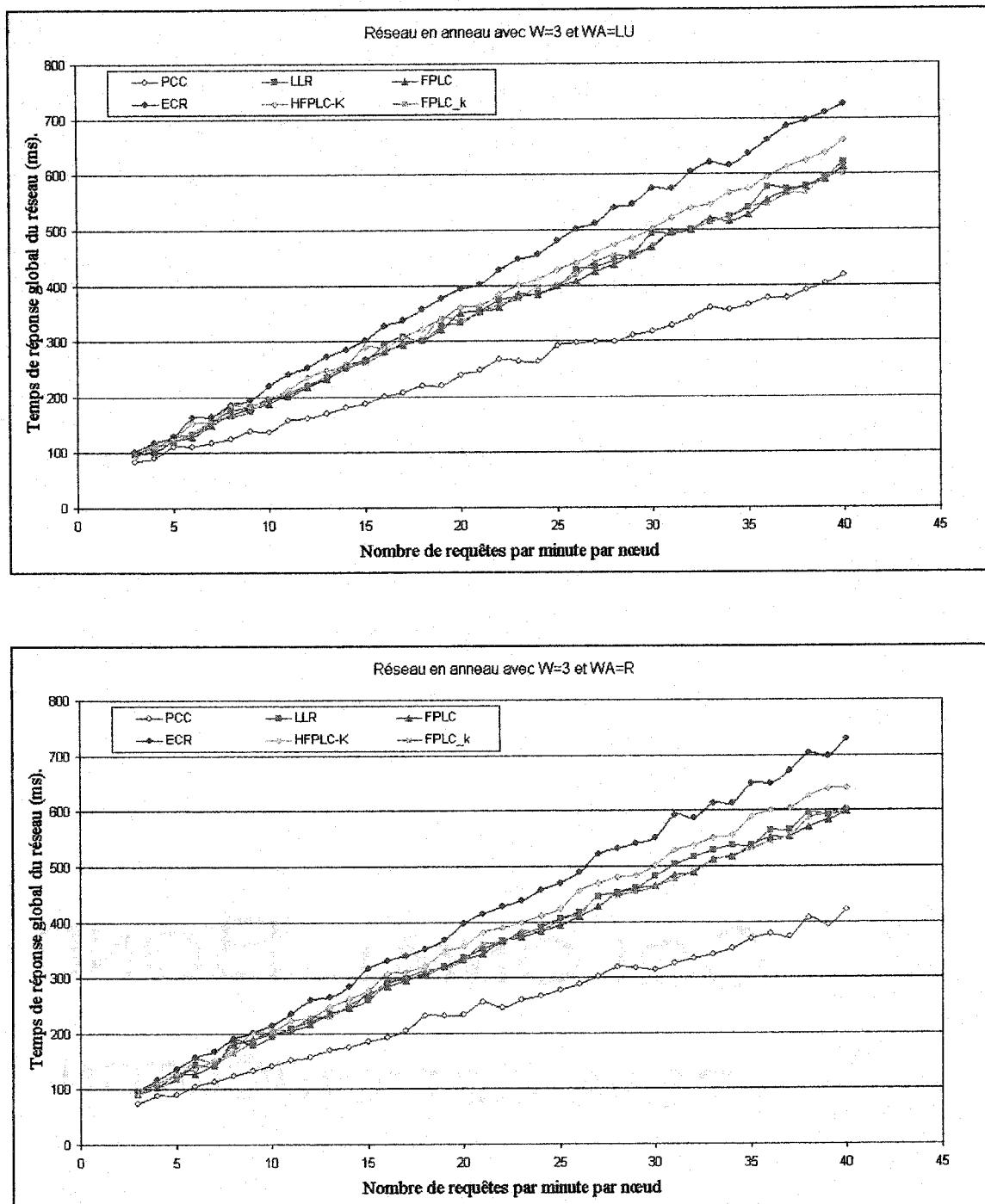
Zhu Y., Rouskas G. N. et Perros G., « A Path Decomposition Approach for Computing Blocking Probabilities in Wavelength-Routing Networks », *IEEE/ACM Trans. on Networking*, vol. 8, No. 6, Dec. 2000, pp. 747-761.

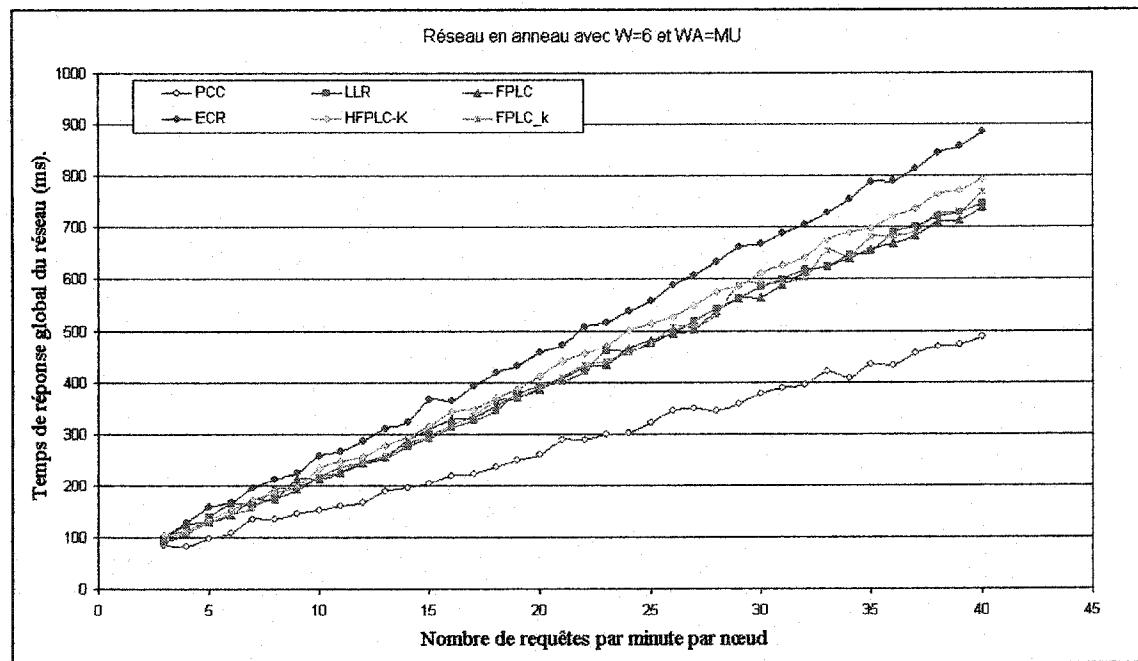
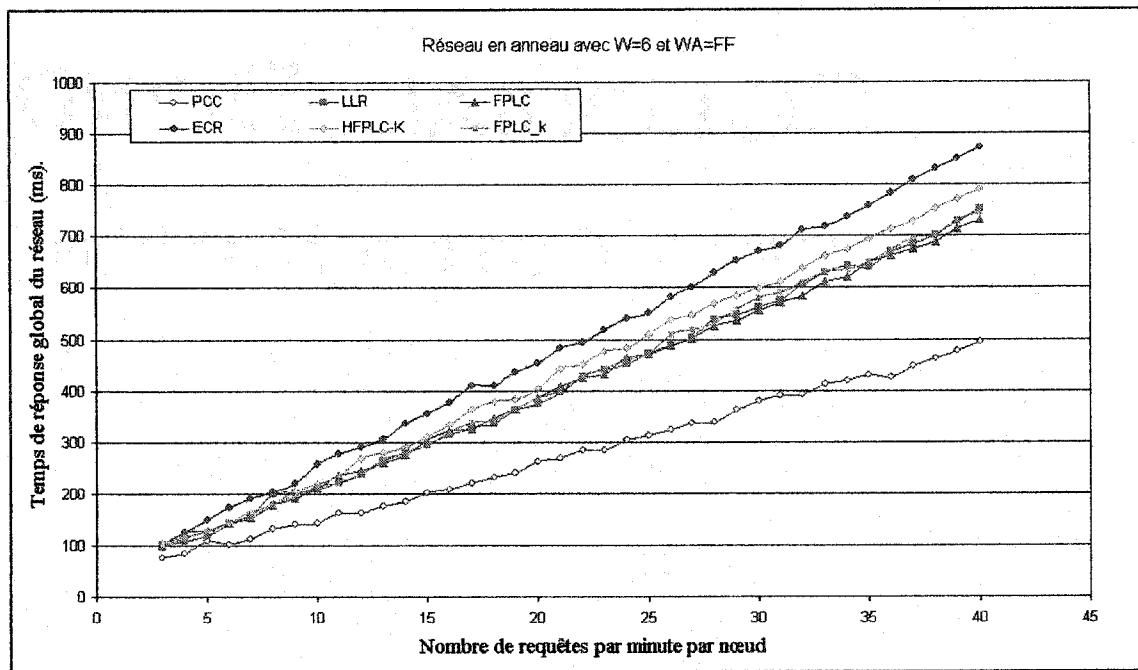
## ANNEXES

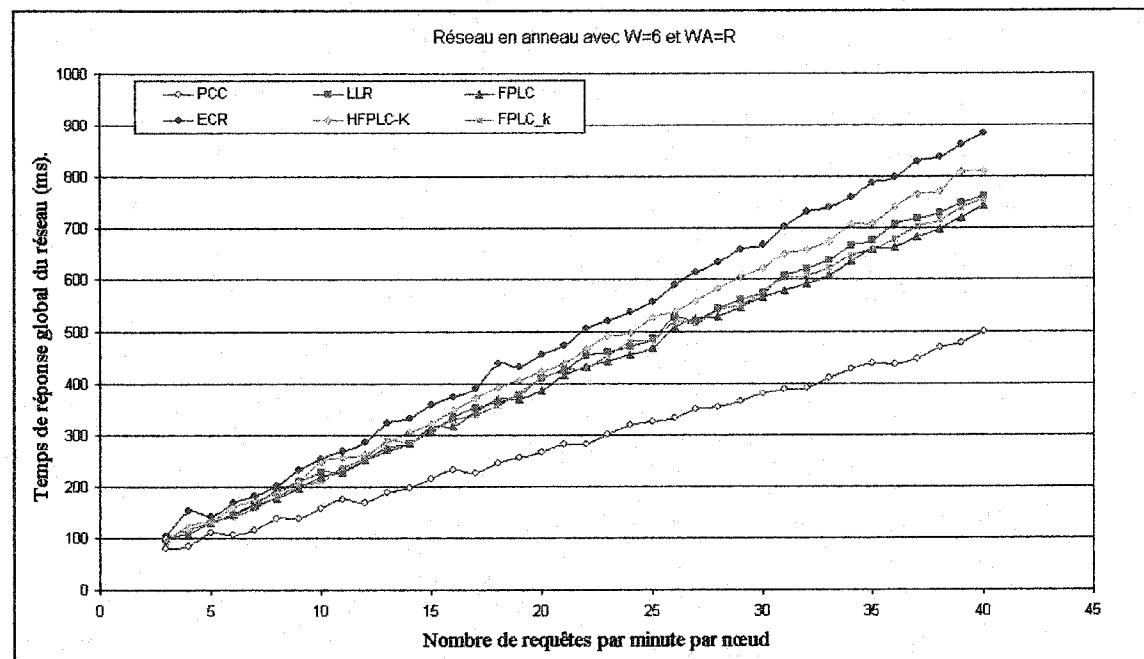
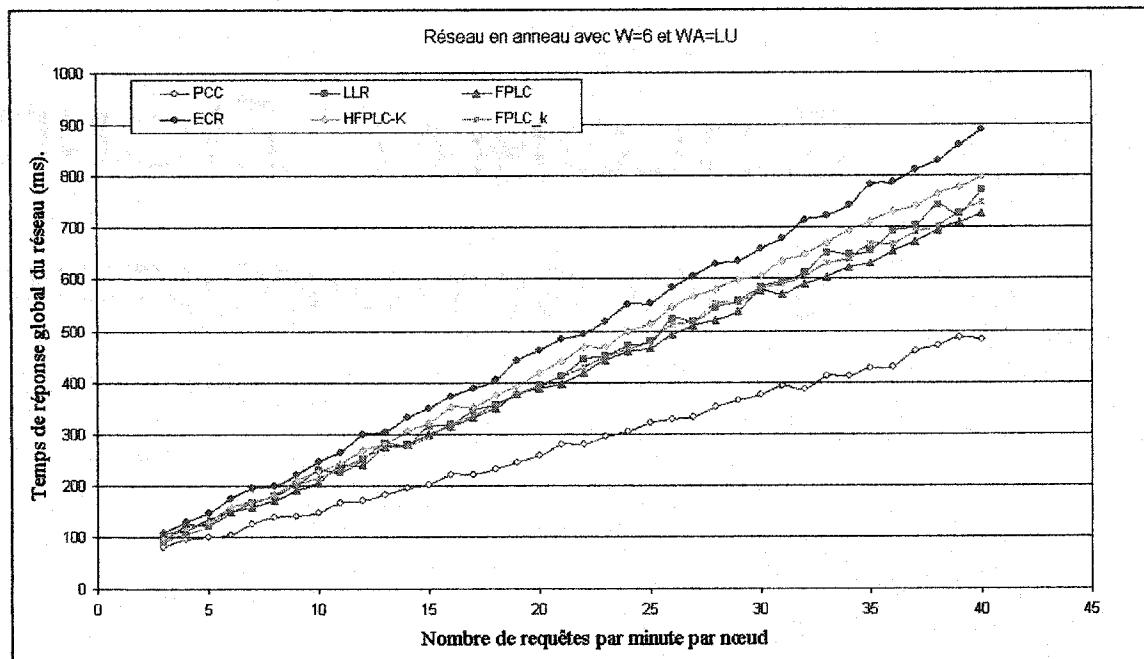
### Résultats additionnels obtenus pour le réseau en anneau

#### Évaluation du temps de réponse pour différents nombres de longueurs d'onde et stratégies d'affectation de longueurs d'onde

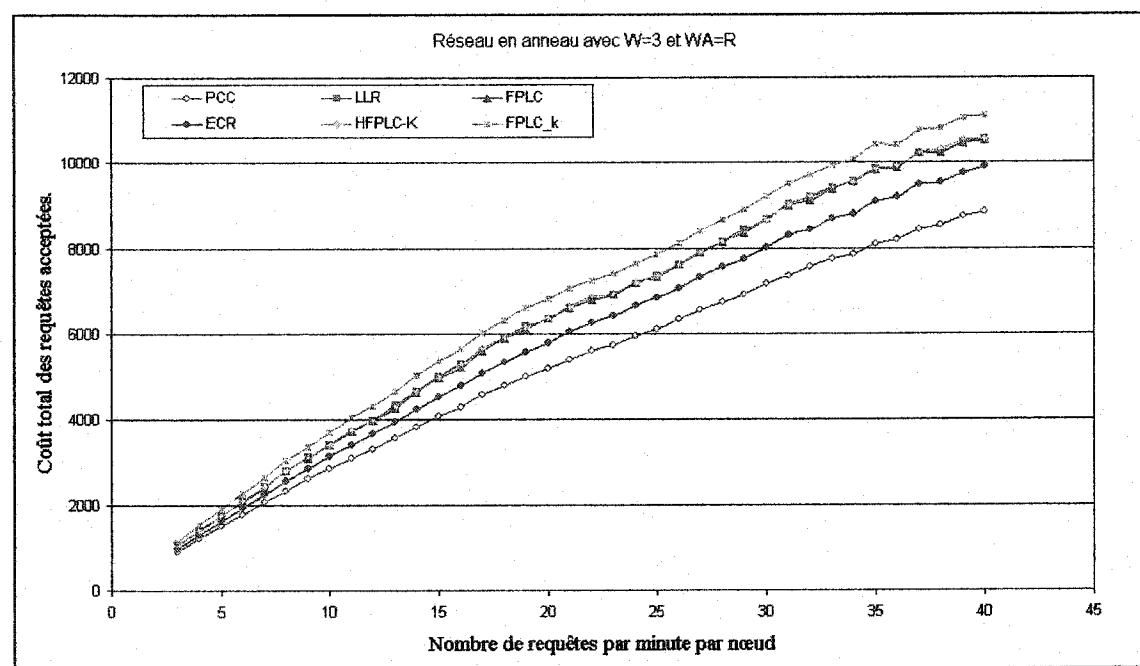
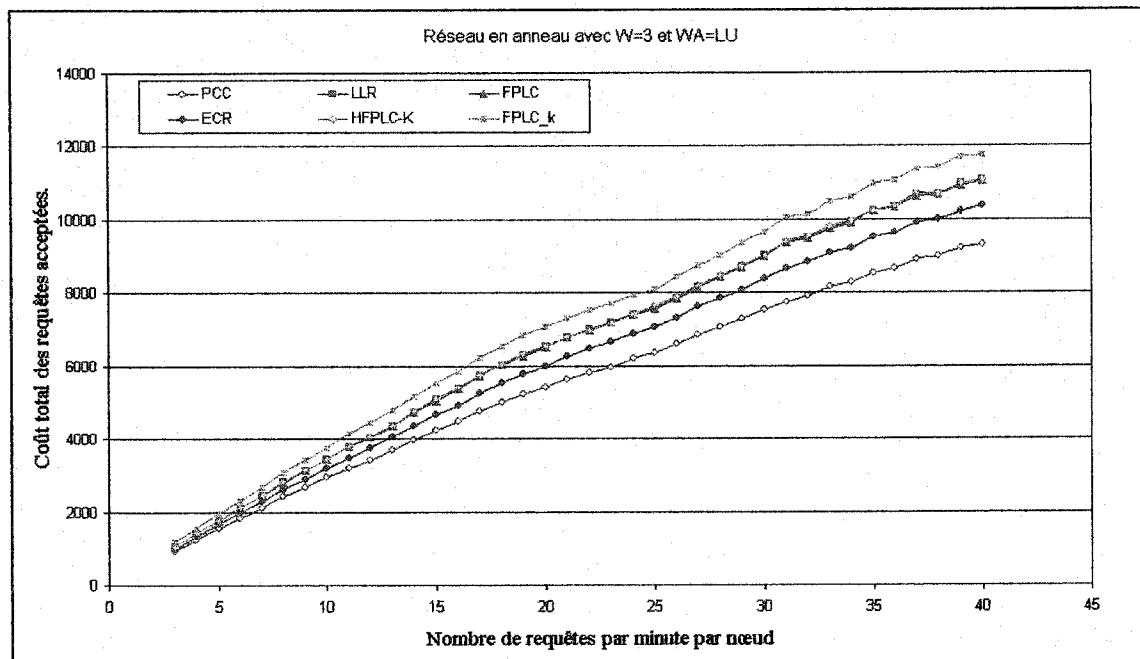


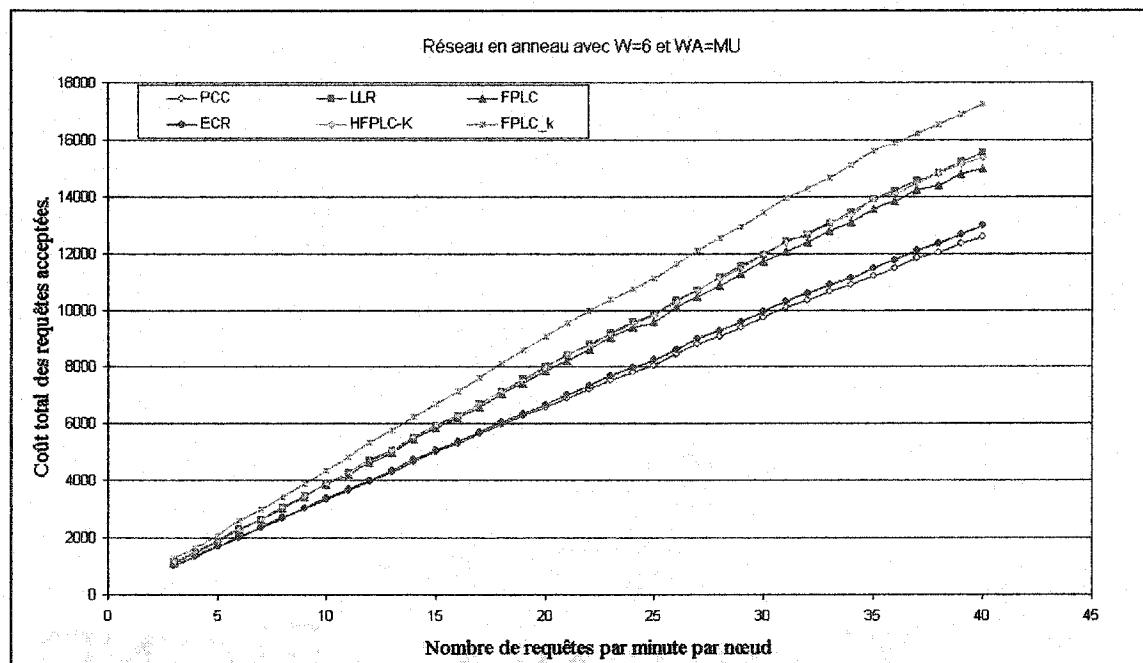
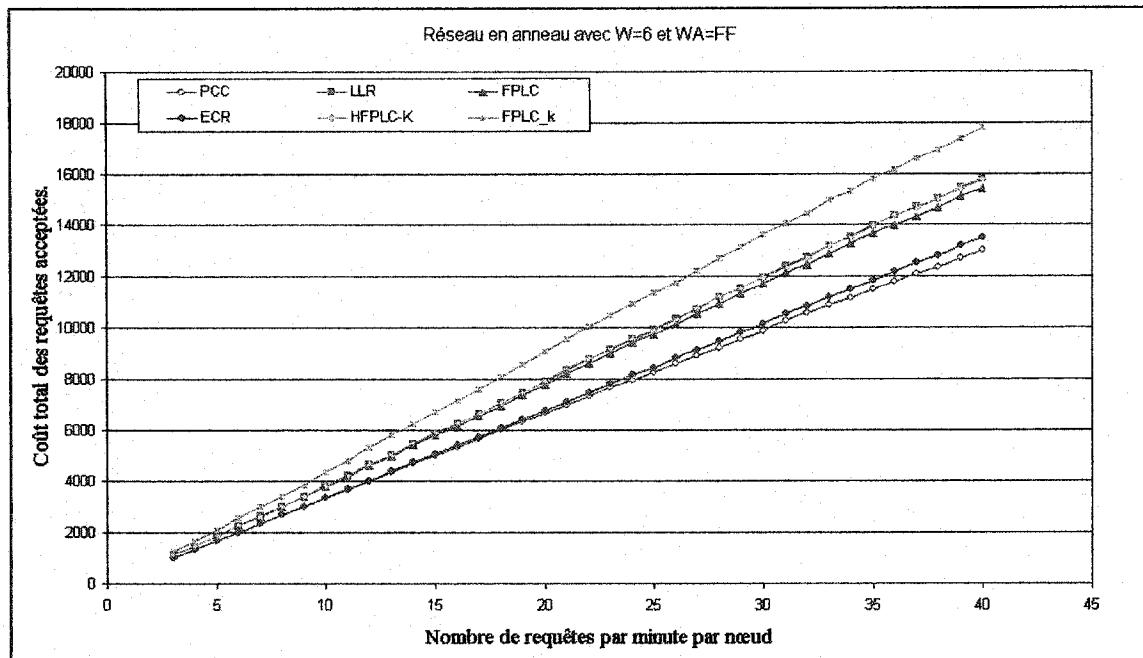


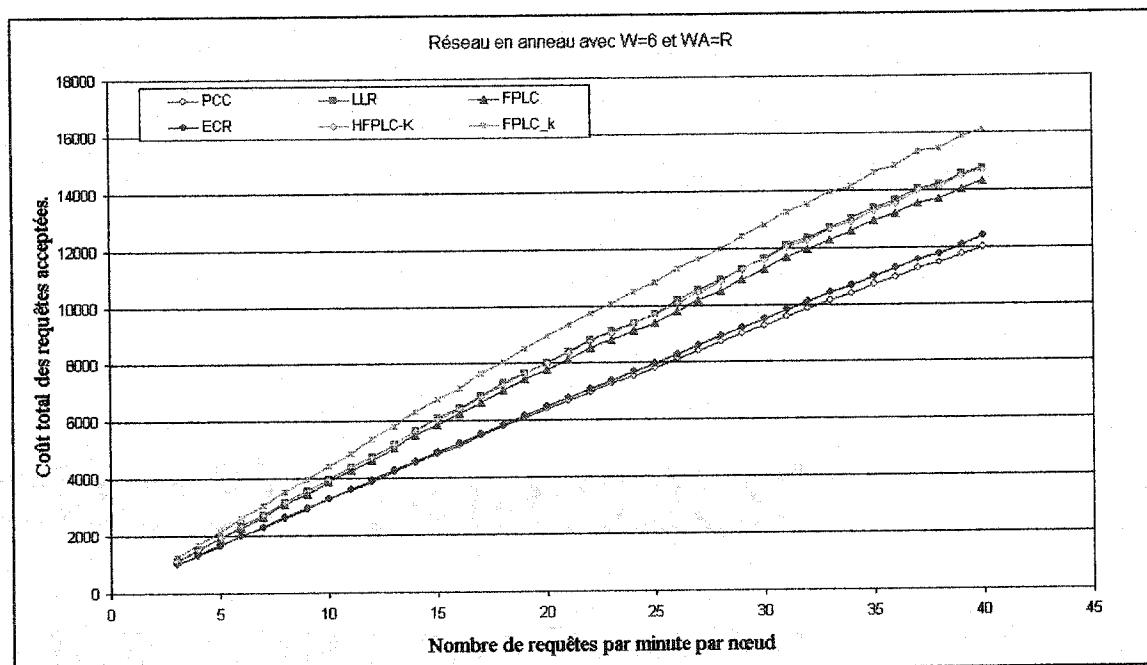
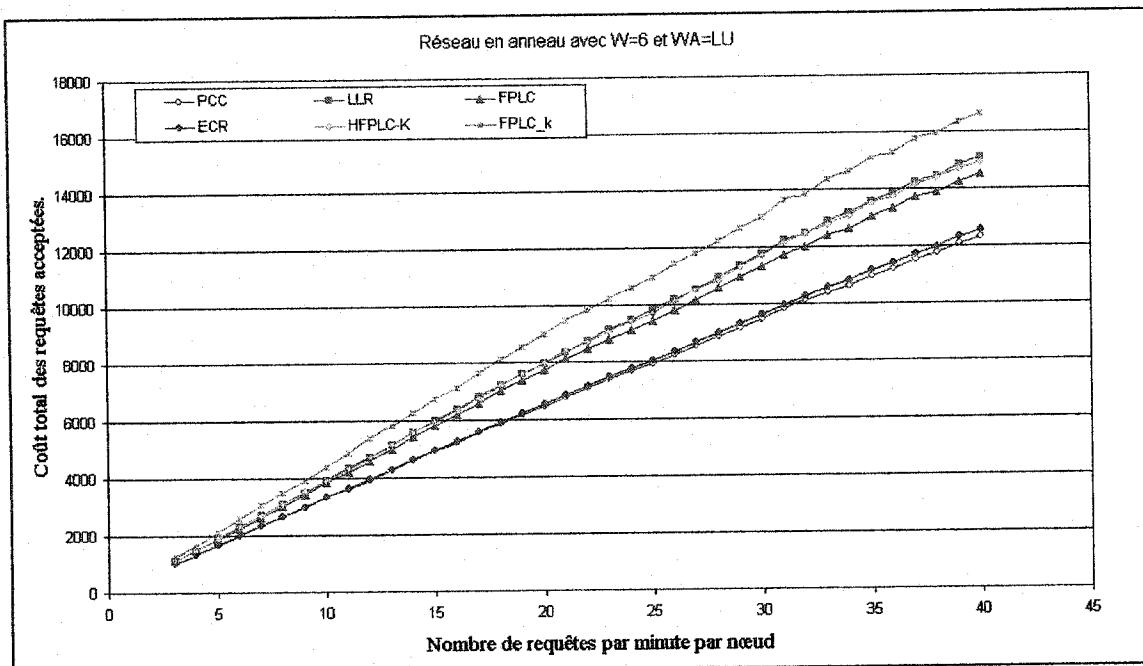




## Évaluation du coût total des requêtes pour différents nombres de longueurs d'onde et stratégies d'affectation de longueurs d'onde

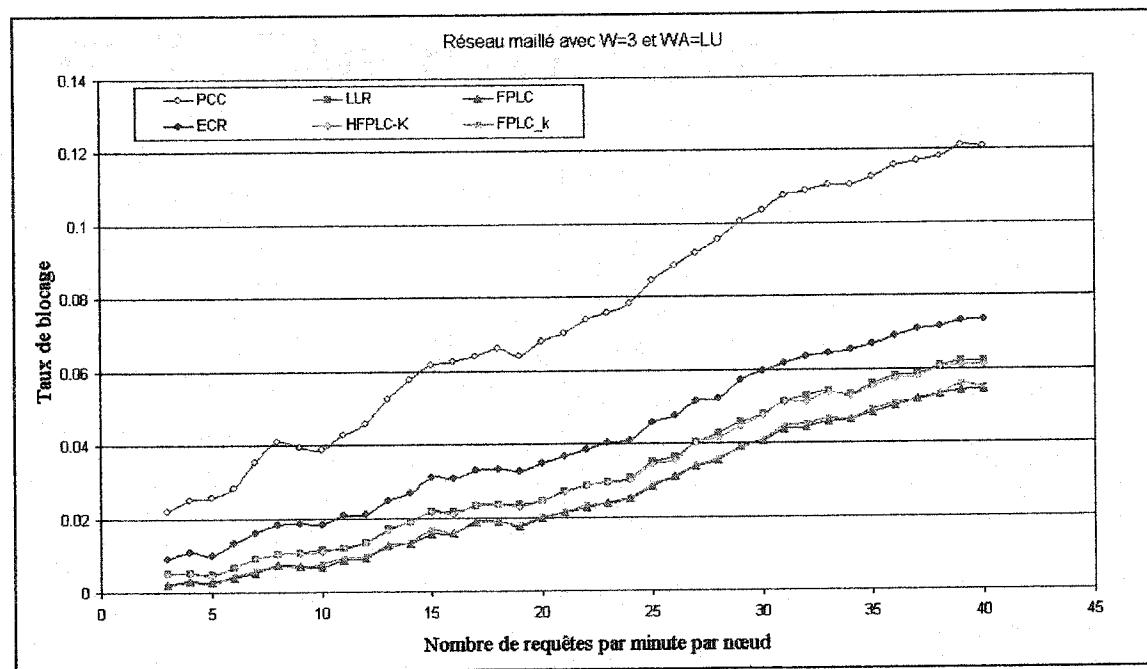
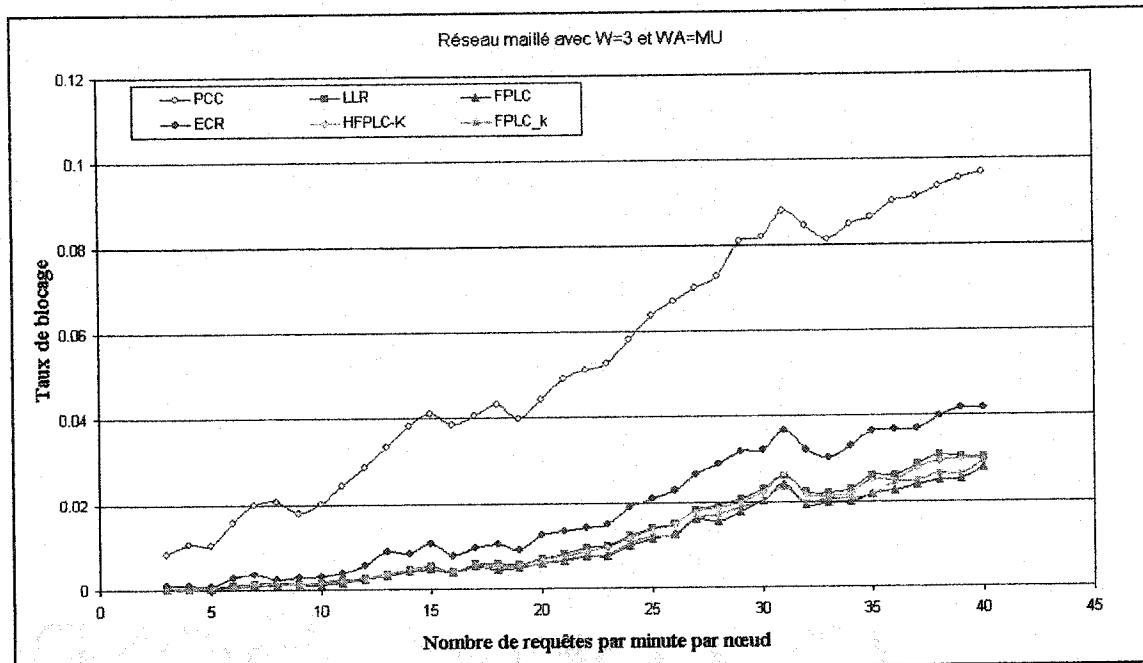


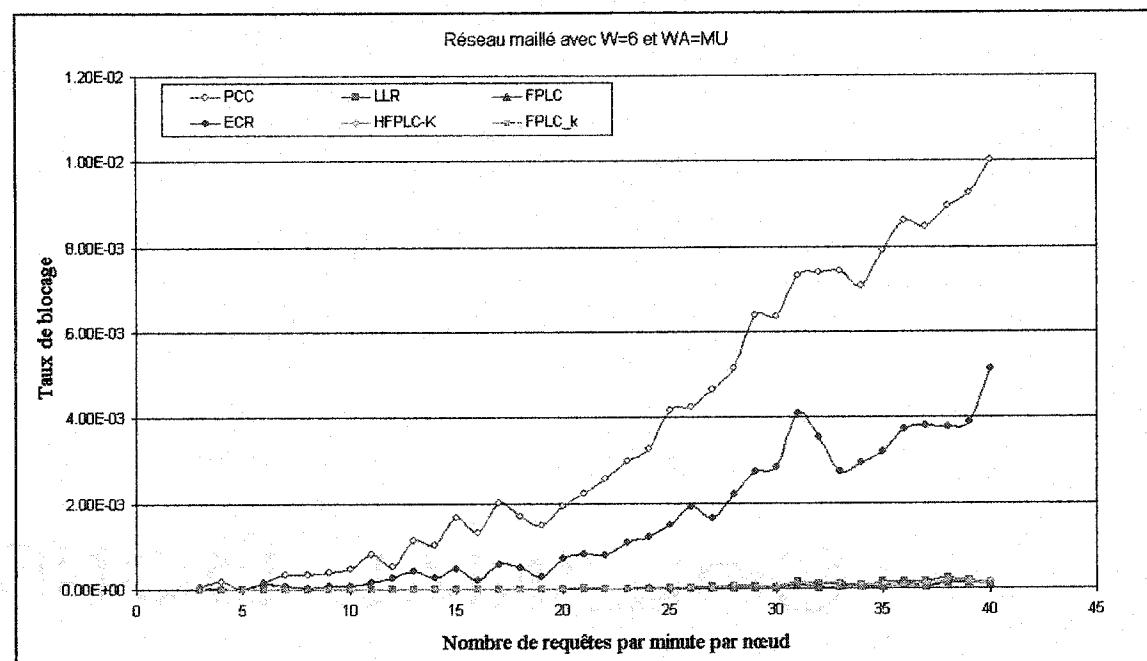
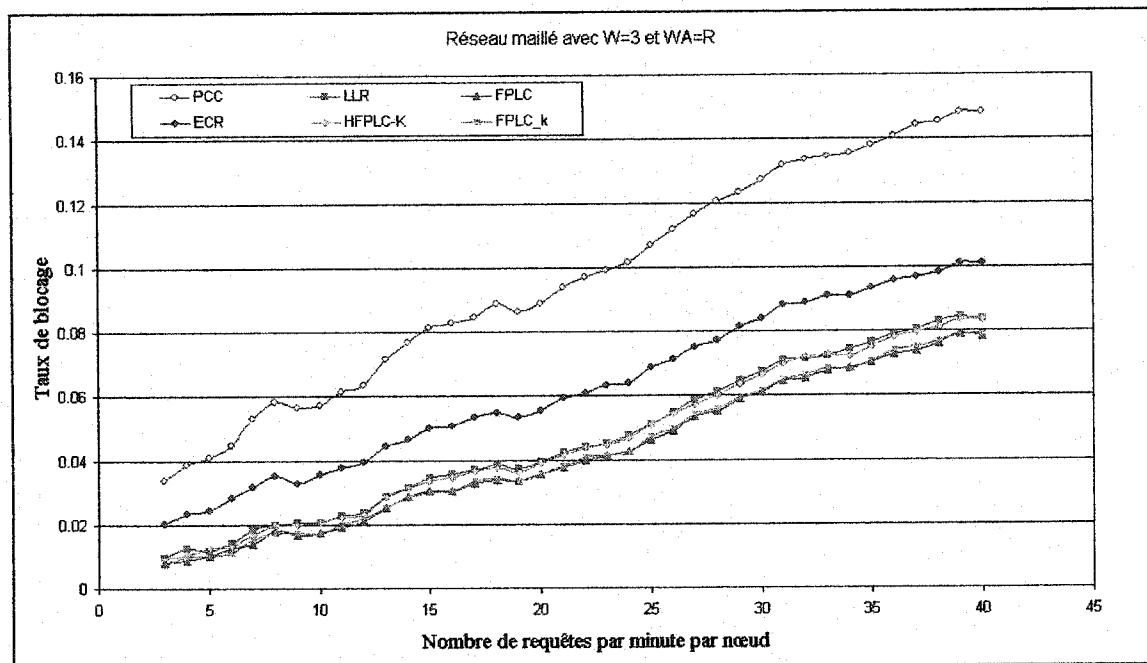


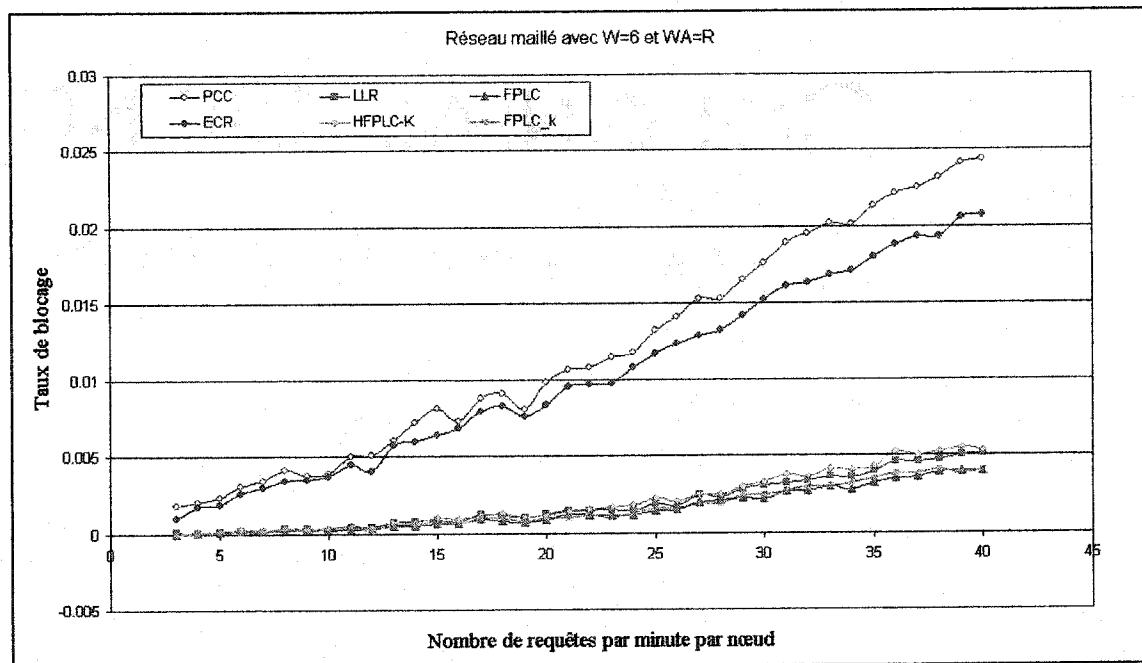
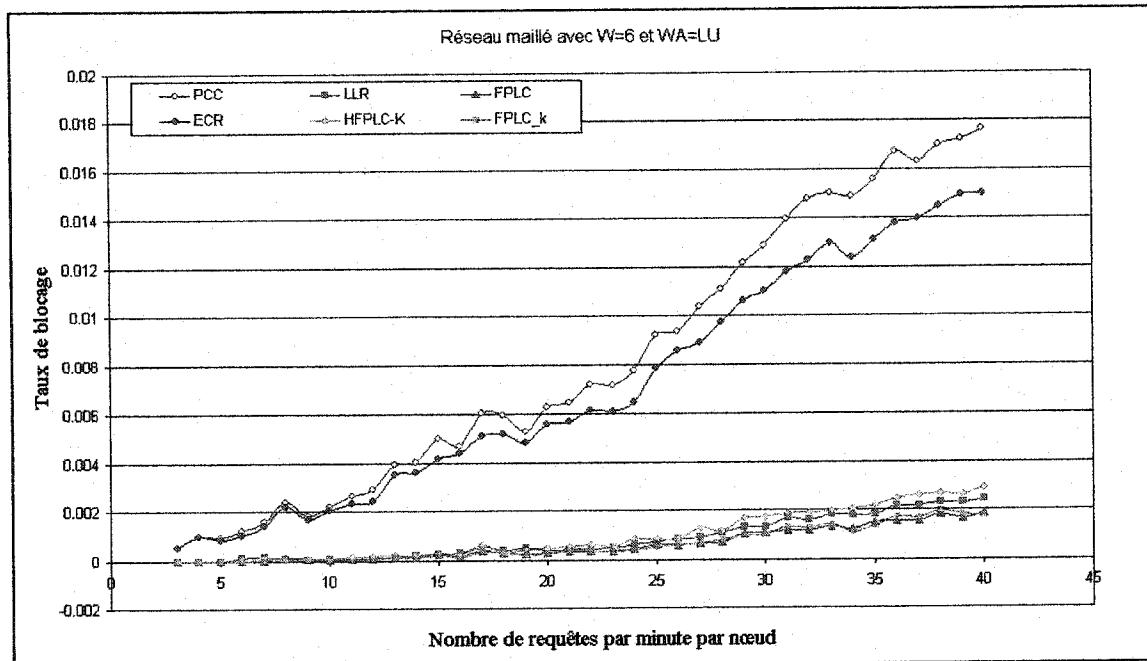


## Résultats additionnels obtenus pour le réseau maillé

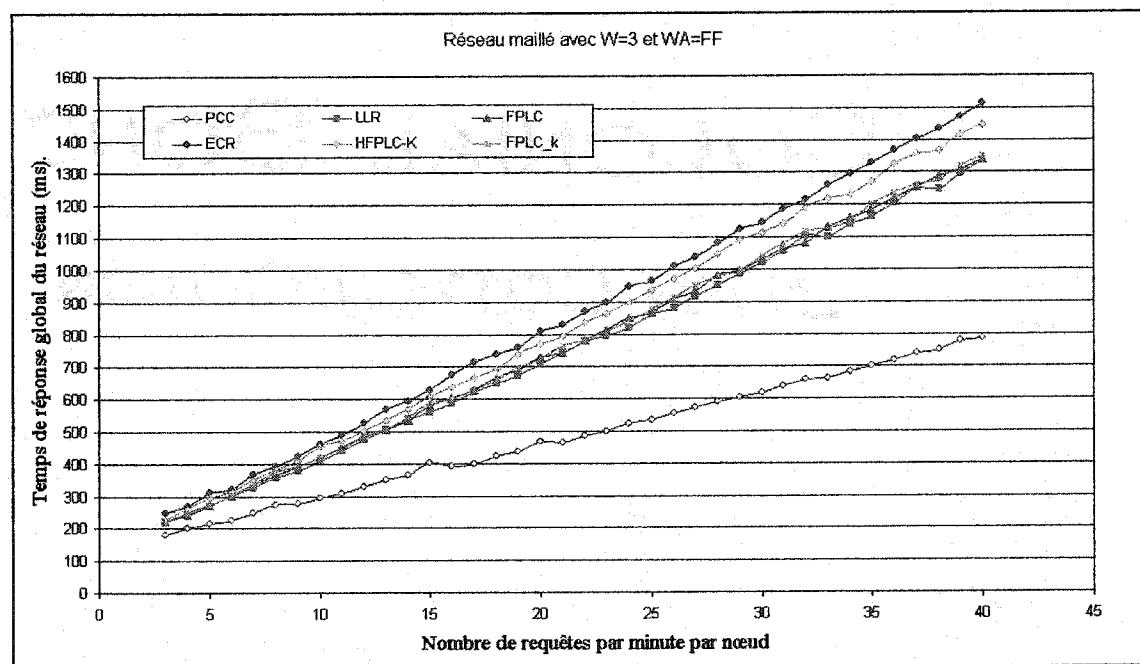
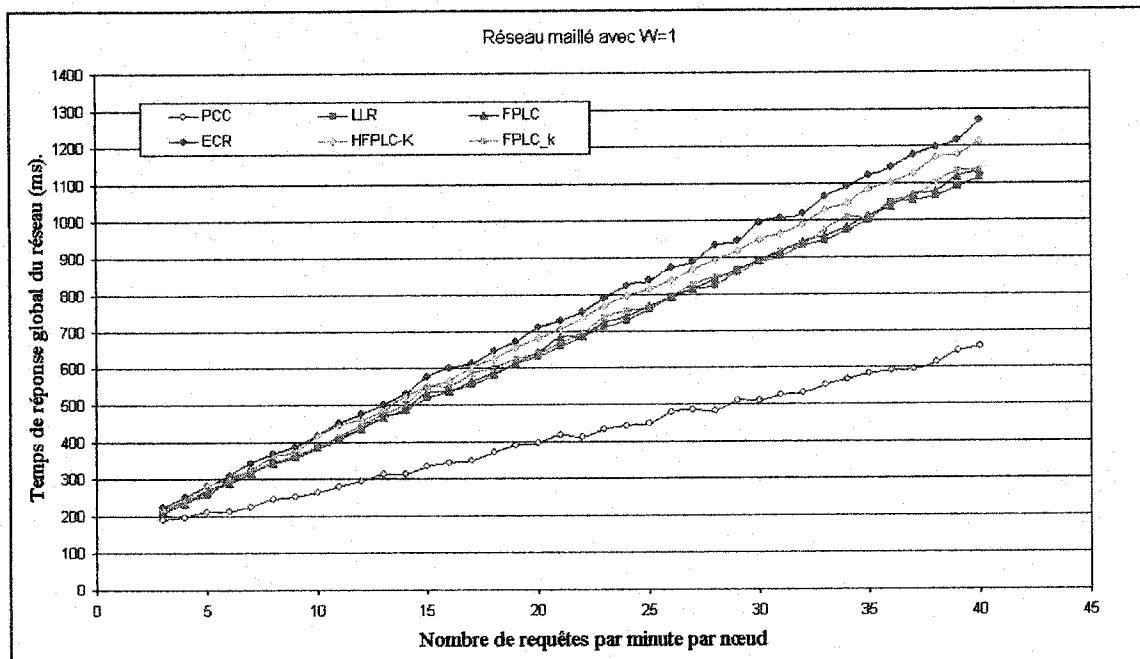
Évaluation du taux de blocage des requêtes pour différents nombres de longueurs d'onde et stratégies d'affectation de longueurs d'onde

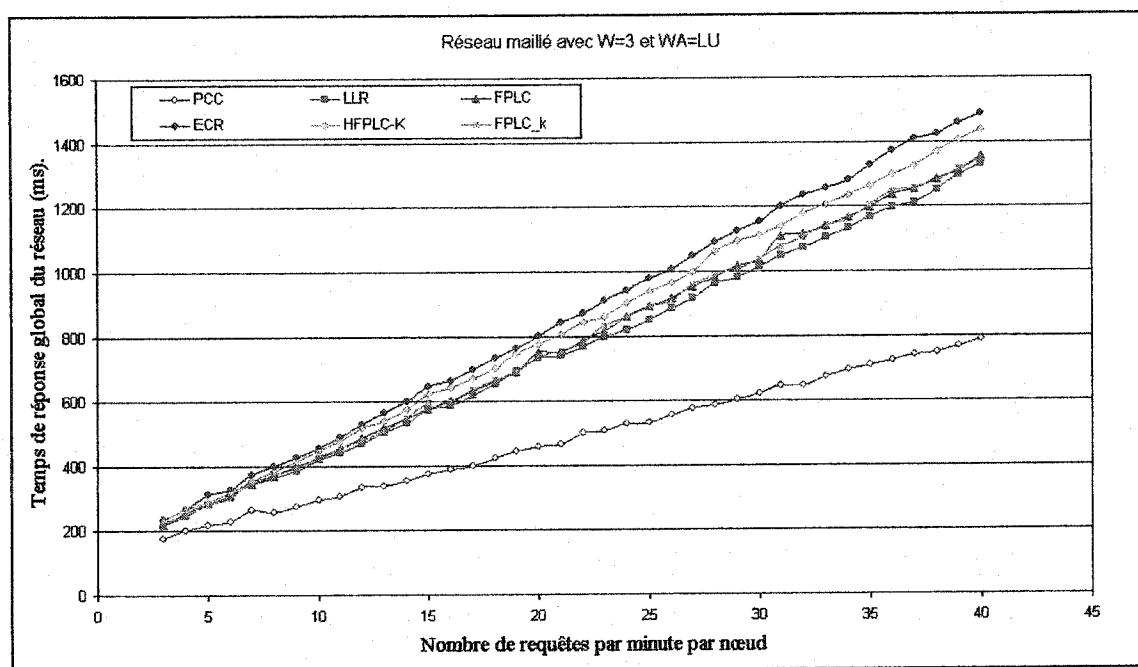
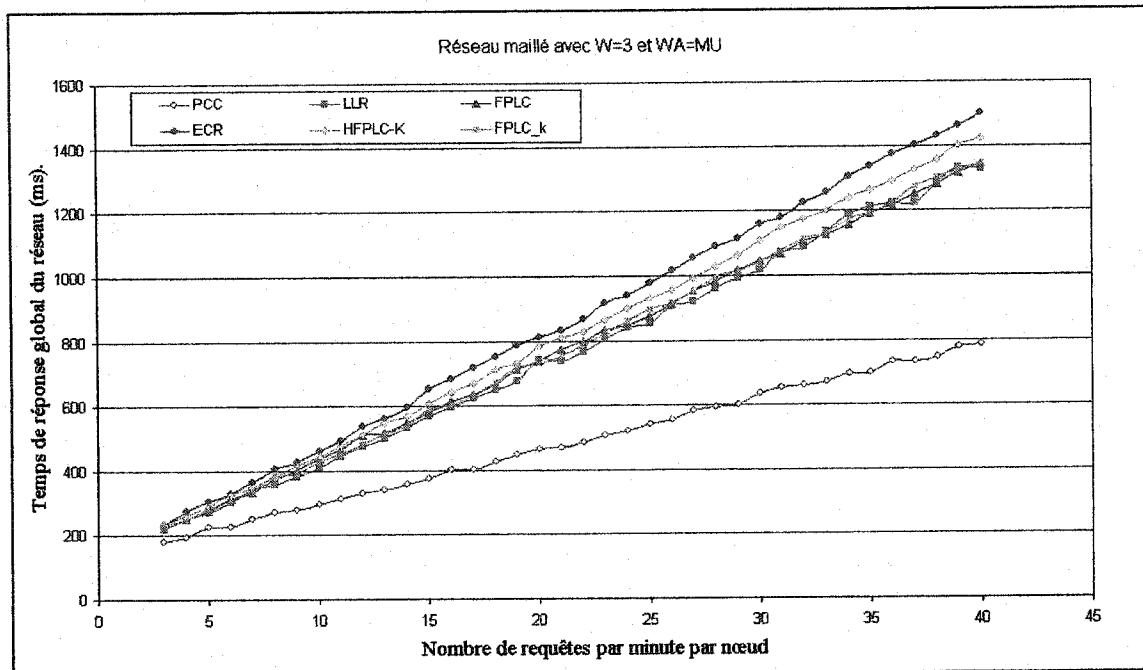


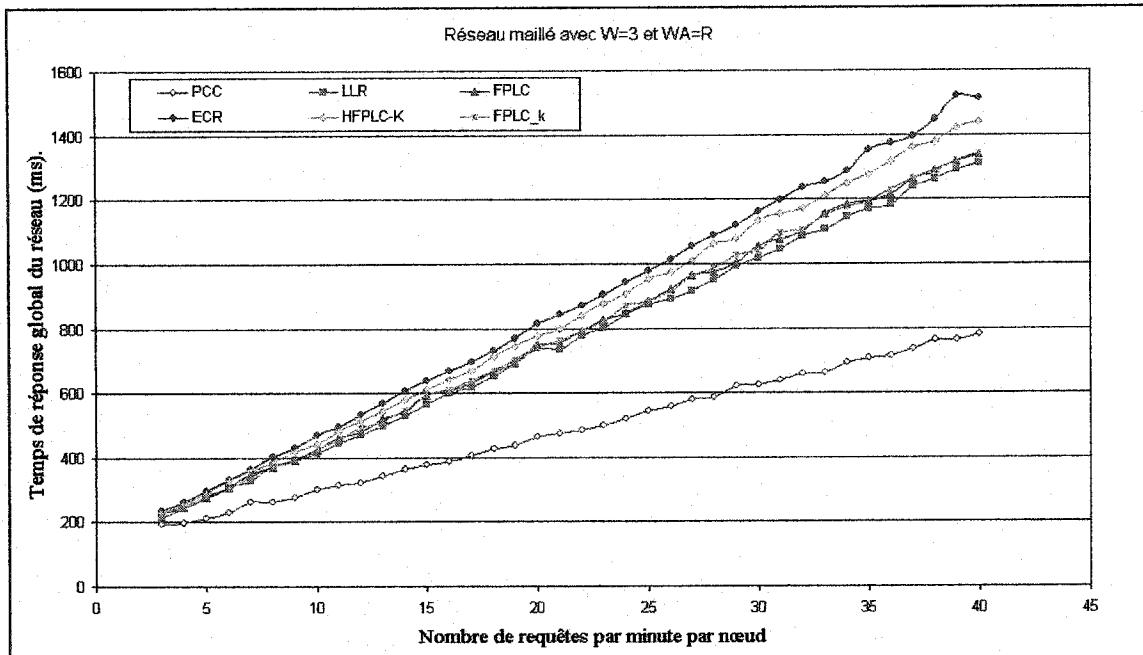




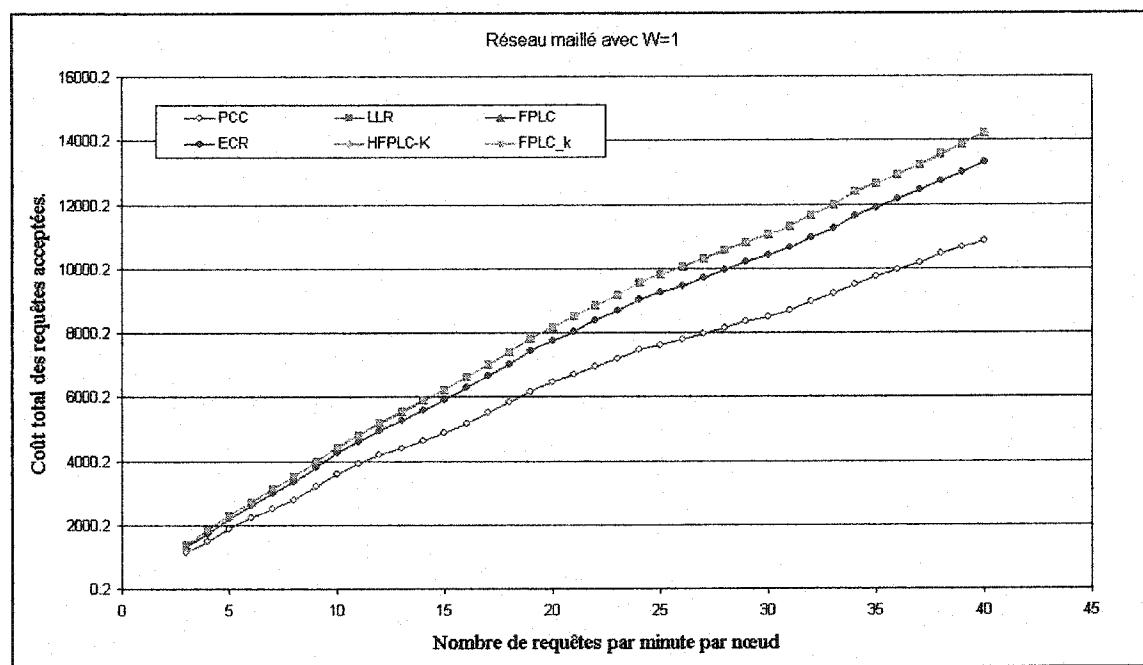
Évaluation du temps de réponse pour différents nombres de longueurs d'onde et stratégies d'affectation de longueurs d'onde

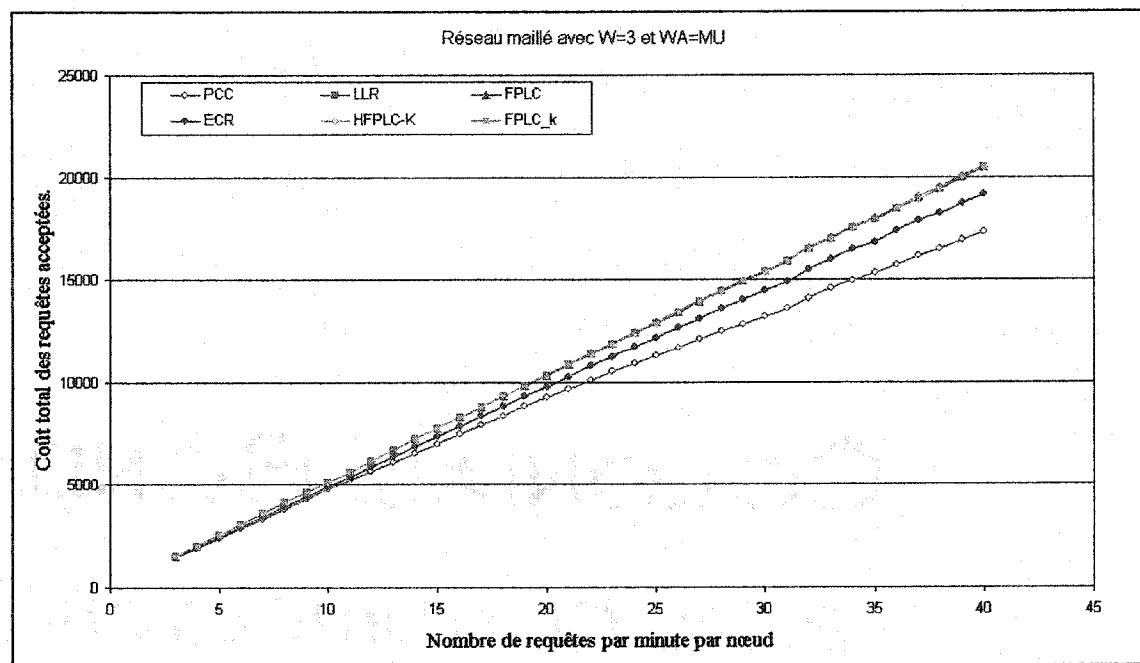
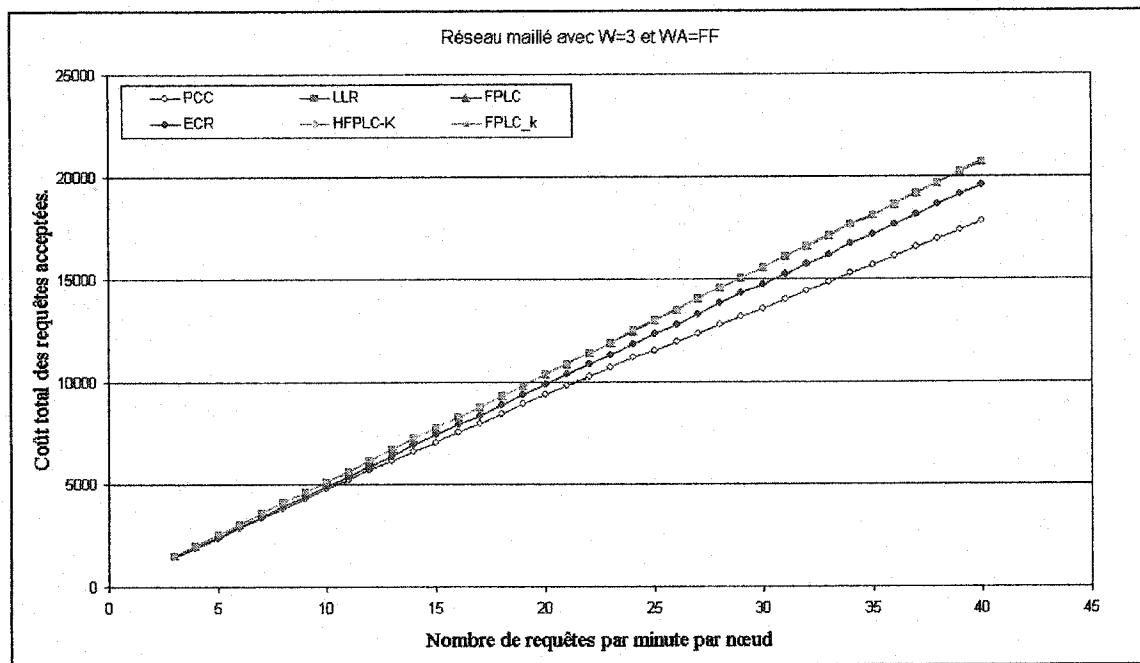


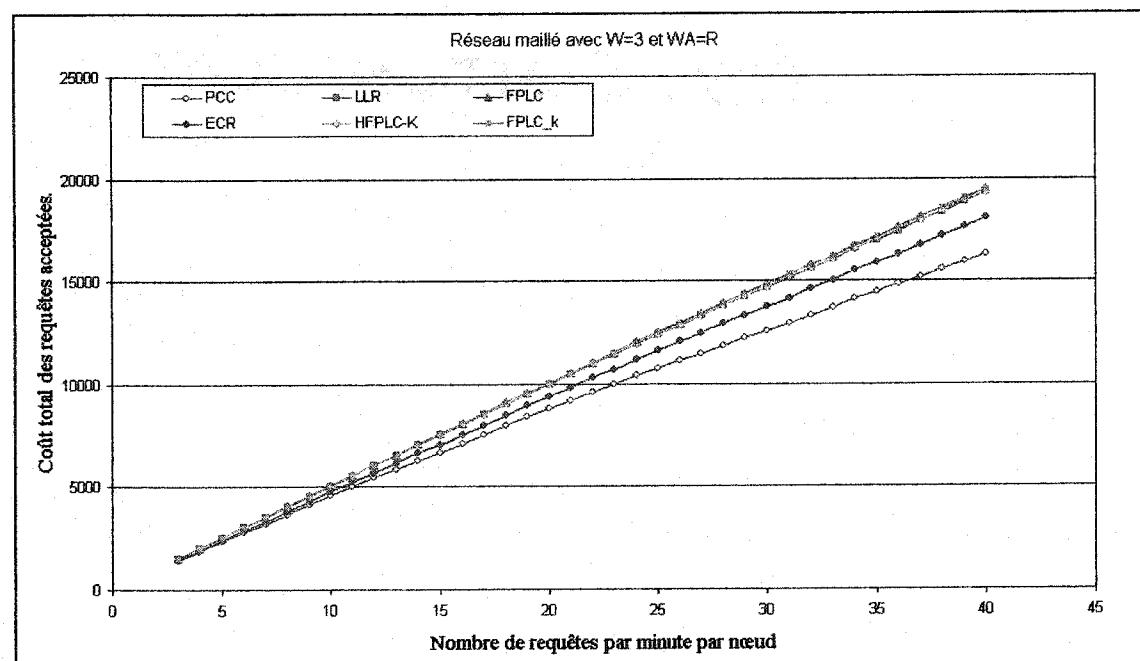
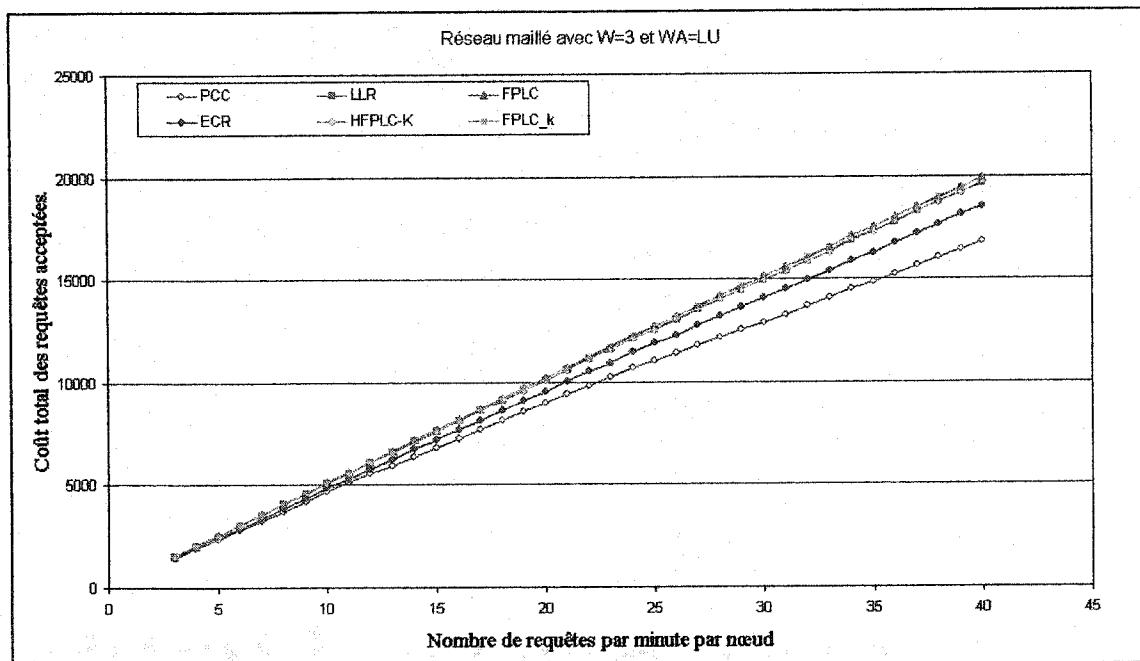




## Évaluation du coût total des requêtes pour différents nombres de longueurs d'onde et stratégies d'affectation de longueurs d'onde







## Résultats additionnels obtenus pour le réseau ARPANET

### Évaluation du taux de blocage des requêtes pour différents nombres de longueurs d'onde et stratégies d'affectation de longueurs d'onde

