Thèse

Présentée pour obtenir le grade de docteur
de l'Ecole Nationale Supérieure
des Télécommunications

Spécialité: **Informatique et Réseaux**

Hazar Aouad

Transport de Flux Temps Réels dans un Réseau IP Mobile

Soutenu le 20 Janvier 2005 devant le jury composé de:

Rapporteurs	Gérard Hébuterne Jean-Pierre Claudé
Examinateurs	Guy Pujolle Jean Michel Fourneau Noémie Simoni Laurent Decreusefond
Directeur de thèse	Samir Tohmé

A tous ceux qui m'aiment
beaucoup et peu

« Faire aisément ce que d'autres trouvent difficile à réaliser,
c'est le talent ;
faire ce qui est impossible au talent,
c'est le génie. »

Henri-Frédéric Amiel

Remerciements

Mes premiers remerciements vont au professeur Samir Tohmé, mon directeur de thèse et actuellement directeur du laboratoire PRISM à l'Université de Versailles Saint-Quentin-en-Yveline. Autre ces remarques et suggestions techniques, académiques et professionnelles, qui m'ont été inestimables, son suivi personnel m'a beaucoup touché. Ses capacités d'écouter et de trouver toujours des solutions constructives sont des qualités rares et très précieuses.

Je tiens à remercier les différents membres de mon jury de thèse présidé par le professeur Guy Pujolle. Messieurs les professeurs Gérard Hébuterne et Jean-Pierre Claudé m'ont fait l'honneur d'accepter de rapporter cette thèse. Mes remerciements vont également aux examinateurs Jean-Michel Fourneau, Noémie Simoni et Laurent Decreusefond.

Je remercie aussi toutes les personnes du département Informatique et Réseaux de l'ENST, le département qui m'a accueilli tout au long de la période nécessaire à l'aboutissement de ma thèse. C'est là où j'ai eu l'occasion et le plaisir de travailler avec différents chercheurs et expert dans différents domaines des réseaux et des télécommunications.

J'ai beaucoup apprécié la chaleur humaine et les gains scientifiques des différentes discussions avec les différents collègues thésards en générale et ceux du groupe Réseau Haut Débit en particulier. M. Rami Langar, actuellement thésard à InfRes, a indirectement participé à des travaux dans cette thèse par le développement d'une partie du simulateur NS au cours de son mémoire de DEA. Sans l'aide des travaux de Dr. Abbas Ibrahim, la partie de l'estimation des flux UMTS ne serait pas réalisée telle qu'elle l'est dans ce manuscrit. Ses études sur la couche MAC radio et son engagement dans le projet Esquimaux, m'ont beaucoup assistaient dans l'aboutissement de cette partie de la thèse.

Mes prochains remerciements ont une portée de 3178 Km. Ils s'adressent à tous les membres de ma famille résidente au Liban. Tous ont contribué, chacun à sa façon et ses moyens, à l'aboutissement de cette thèse. Tout d'abord je remercie mon père, mère et mes frères pour leurs soutiens inconditionnels et illimités. Ensuite, mes grandes mères et mon grand père, mes oncles et mes tantes, mon cousin et mes cousines, m'ont toujours aidé, écouté, distrait ou même remis sur le droit chemin au bon moment.

Dans le meilleur et dans le pire, le sourire angélique de Joumana Yammine m'a toujours fait du bien. J'espère qu'à mon tour, je t'aiderai dans le déroulement de ta thèse et dans beaucoup d'autres domaines aussi.

Parmi mes nombreux amis que je remercie, je suis particulièrement reconnaissant à mon binôme et actuellement colocataire Dany Hanna et à M^{e} Nidal-Georges Hassoun qui n'a cessé de m'encourager tout au long de mes années de recherches.

Je n'oublierai point de citer le révérend père Samer Nassif, en même temps confesseur et ami, qui était toujours à mes cotés dans les moments difficiles. Même si je ne sais pas le nombre de vos prières d'intercession pour moi, je vous assure que je sentais leurs effets. Je remercie aussi monseigneur Saïd Elias Saïd, curé de la paroisse Notre Dame du Liban et triple docteur (Médecine, théologie et droit canonique), qui m'a toujours encouragé durant ma thèse, lui qui s'y connaît bien des pièges rencontrés durant de tel travaux.

Enfin, si j'ai oublié quelqu'un, je le remercie aussi, ainsi que toute autre personne qui m'a aidé de prêt ou de loin au cours de mon cheminement durant mes années de thèses.

Résumé

Dans cette thèse, nous étudions plusieurs méthodes pour la mise en place de la QoS (Quality of Service) dans un réseau IP. Avant d'entamer nos travaux de recherche, nous dévoilons tout d'abord les différents mécanismes de QoS que nous étudierons dans la thèse. MPLS (Multi Protocol Label Switching), DiffServ (Differentiated Services) et les algorithmes d'ordonnancement formeront la base du réseau cœur que nous utiliserons. En accord avec plusieurs travaux, nous définissons trois classes de service à différencier dans le réseau. La première classe comporte les flux voix. Elle nécessite un délai faible et une gigue réduite. Les flux de "données critiques", qui requièrent un taux de perte faible et un délai borné, forme la deuxième classe. La troisième classe, qui regroupe les applications telles que le transfert de fichiers ou l'échange de courriers électroniques, n'exige aucune condition particulière du réseau.

Dans un premier temps, nous modélisons les flux sortant/entrant d'un réseau mobile sans fil. D'abord, nous modélisons les lois d'inter arrivée des paquets d'un flux agrégé au niveau de la couche MAC (Medium Access Control) entrant au réseau UTRAN (UMTS Terrestrial Radio Access Network), le réseau d'accès à l'UMTS (Universal Mobile Telecommunication Service). Le protocole CDMA (Code Division Multiple Access) utilisé dans ce réseau propose un accès différent en fonction de la QoS demandée. Ensuite, nous déterminons la loi caractérisant l'inter arrivée des paquets sortant d'un réseau WiFi (Wireless Fidelity) utilisant la couche MAC 802.11 de base. Pour ces deux réseaux, nous proposons différents modèles d'agrégation de flux voix, Web, transfert de fichiers ou un multiplexage de ces différentes classes. Nous mesurons l'adéquation de deux modèles de loi de distributions aux traces créées. La première distribution est le processus MMPP (Markov Modulated Poisson Process) qui représente un modèle Markovien. Nous expérimentons deux valeurs du nombre d'états: 2 et 4. La seconde loi que nous considérons est la loi Gaussienne. Nos résultats montrent que d'une part, le type des flux agrégés et d'autre part, le réseau utilisé, influent tous les deux sur le modèle produit.

Dans un second temps, nous développons les équations qui déterminent les probabilités stationnaires d'une file d'attente implémentant l'ordonnanceur GPS (Generalised Processor Sharing) avec trois classes de service. En utilisant le mécanisme DiffServ pour différencier les flux, nous mesurons la QoS à la sortie d'une file unique utilisant WRR (Weighted Round Robin), un des algorithmes qui approximent GPS. Nous traçons alors les différentes courbes de délai et de taux de perte observés à la sortie de cette file en fonction de la pondération et de la charge créée par chacune des classes. Nous appliquons les différentes conclusions du choix des paramètres que nous tirons d'un seul serveur à un réseau entier. De plus, nous ajoutons l'ingénierie de trafic de MPLS pour quantifier le gain mesuré par chaque politique. A partir de ce travail, nous avons pu généraliser nos constatations qui deviennent valable aussi bien sur une file que dans un réseau.

Dans un troisième temps, nous développons une méthode d'adaptation dynamique du routage. Nous la proposons afin de palier aux variations de la distribution du délai sur les liens qui forment le chemin de bout en bout. Ce mécanisme se base sur les techniques de tomographie des réseaux afin d'estimer la distribution du délai sur les différents tronçons des chemins observés. Si le délai moyen sur la route utilisée reste supérieur d'un seuil ε pendant un temps τ au délai moyen d'un autre chemin, le mécanisme déclenche alors la procédure de modification du chemin emprunté. C'est l'utilisation du protocole MPLS associé à ce mécanisme qui permet une modification souple et rapide des itinéraires.

Abstract

In this thesis, we study several implementation methods of the QoS (Quality of Service) in an IP network. Before starting our research, we will first reveal the various mechanisms of QoS which we will study in the thesis. MPLS (Multi Protocol Switching Label), DiffServ (Differentiated Services) and scheduling algorithms will form the basic mechanisms of our backbone network which we will study. In accordance with several other works, we define three classes of services to differentiate in the network. The first class includes voice flows. It requires a small delay and a reduced jitter. Flows of "critical data", requiring a low loss probability and a limited delay, form the second class. The third class, gathering the applications such as file transfer or email exchange, do not need any condition from the network.

As a first step, we model outgoing/entering flows of a mobile wireless network. We model the inter arrival law of the aggregation of the flows at the MAC (Medium Access Control) layer, entering the UTRAN (UMTS Terrestrial Radio Access Network) network, the UMTS (Mobile Universal Telecommunication Service) access network. The CDMA (Code Division Multiple Access) protocol used in this network, proposes different access probability according to the QoS required. We determine also the characterizing law of the inter arrival aggregated traffic leaving a WiFi (Wireless Fidelity) network, using the basic MAC 802.11 layer. For these two networks, we propose various models of aggregation of voice, Web, file transfer flow and a multiplexing of these various classes. We measure the precision of two distribution models to those created by the traces. The first distribution used is the MMPP (Markov Modulated Poisson Process) process which represents a Markovian model. We experimented two values of the states number: 2 and 4. The second law that we consider is the Gaussian law. Our results show that both the type of aggregate flows and the network used influence the resulting model.

In a second step, we develop the equations determining the stationary probabilities of a queue implementing a GPS (Generalised Processor Sharing) scheduler with three classes of services. By using the DiffServ mechanism to differentiate flows, we measure the QoS at the output of a single queue using WRR (Weighted Round Robin), one of the algorithms approximating GPS. We then plot the various curves of delays and loss probabilities observed at the output of this queue according to weight and the load created by each class. We apply the various conclusions of the choice of the parameters which we draw from one server to a whole network. Moreover, we add the traffic engineering capability of MPLS to quantify the gain added by each policy. From this work, we can generalize our observations which become valid as well on a queue as in a network.

In a third step, we develop a method of dynamic adaptation of the routing. We propose it in order to solve the problem with the variations of the delay distribution on the links forming the end to end path. This mechanism is based on network tomography techniques in order to estimate the delay distribution on the various sections of the paths observed. If the average delay on the used path remains greater of the ε threshold for a time τ than the average delay of another path, then, the mechanism initiates the procedure of path modification. The use of the MPLS protocol allows this mechanism to be flexible and fast.

Table des matières

Introduction 1
1. Cadre générale 1
2. Objectifs de la thèse 3
3. Contributions de la thèse 3
4. Plan de la thèse 4
CHAPITRE 1 La garantie de la qualité de service dans un réseau IP 6
1.1. Introduction 6
1.2. Le protocole MPLS 6
1.2.1. Entête MPLS 8
1.2.2. Label Switch Path (LSP) 8
1.2.2.1. La création des LSP 9
1.2.2.2. Les méthodes de distribution de label 9
1.2.2.3. Utilisation des LSP 10
1.2.3. Application des LSP: Ingénierie de trafic MPLS 11
1.3. La différenciation de service par DiffServ 12
1.3.1. L'entête DiffServ 12
1.3.2. Le traitement à l'entrée du réseau DiffServ 13
1.4. La combinaison DiffServ et MPLS 14
1.5. Algorithmes d'ordonnancement 15
1.5.1. First In First Out (FIFO) 16
1.5.2. Head Of the Line (HOL) 16
1.5.3. Generalised Processor Sharing (GPS) et algorithmes à poids dérivés 17
1.5.3.1. GPS 17
1.5.3.2. Weighted Fair Queuing (WFQ) 18
1.5.3.3. Worst-case Fair Weighted Fair Queuing (WF^2Q) 18
1.5.4. Round Robin (RR) et ses dérivées 21
1.5.4.1. Weighted Round Robin (WRR) 21

1.5.4.2. Deficit Round Robin (DRR) 21
1.5.5. Les ordonnanceurs à temps 21
1.5.5.1. Earliest Deadline First (EDF) 22
1.5.5.2. Earliest Due Date (EDD) 22
1.5.6. Comparaisons d'ordonnanceurs à poids 23
1.6. Conclusion 32
CHAPITRE 2 Modélisation du trafic agrégé sortant d'un réseau sans fil UTRAN ou 802.11 34
2.1. Le réseau UMTS 35
2.1.1. L'architecture du réseau UMTS 35
2.1.2. La technique CDMA 36
2.1.3. Les services 39
2.1.4. La fonction CAC 39
2.1.5. Le contrôle de trafic 40
2.1.6. Les applications utilisées 41
2.1.6.1. Sous système voix 41
2.1.6.1.1. Application voix (RTC) 41
2.1.6.2. Sous système data 42
2.1.6.2.1. Application web (ITC) 42
2.1.6.2.2. Application transfert de fichiers (STC) 44
2.2. Le réseau WiFi 802.11 45
2.2.1. Introduction 45
2.2.2. Le protocole MAC 802.11 46
2.3. Modèle de flux 48
2.3.1. Sources voix 49
2.3.2. Sources web 49
2.3.3. Sources de transfert des fichiers 49
2.3.4. Multiplexage de service 49
2.4. Agrégation de flux On/Off 49
2.5. Méthodologie d'estimation 51
2.5.1. Estimation MMPP 51
2.5.2. Estimation Gaussienne 53
2.6. Résultats 53
2.6.1. Réseau UMTS 53
2.6.1.1. Utilisateurs voix 53
2.6.1.2. Utilisateurs Web 55
2.6.1.3. Utilisateurs FTP 56
2.6.1.4. Multiplexage d'utilisateurs voix Web et FTP 57
2.6.2. Réseau 802.11 58
2.6.2.1. Utilisateurs voix 58
2.6.2.2. Utilisateurs FTP 61
2.6.2.3. Multiplexage d'utilisateurs voix et FTP 63
2.7. Conclusion 71
CHAPITRE 3 Amélioration des performances par la différenciation des classes 73
3.1. Modèle analytique pour GPS avec trois classes de service 73
3.2. Résultats pour trois classes de service 77
3.2.1. M/M/1 avec trois classes de service 78
3.2.2. M/Pareto/1 avec trois classes de service 83
3.3. Les règles à tirer 88
3.4. Application sur un réseau 88

3.4.1. Le modèle du réseau ... 88
3.4.2. Les Modèles de trafics ... 90
3.4.3. Les paramètres des scénarios de simulations ... 92
3.4.4. Les résultats ... 93
3.4.4.1. Sans QoS ... 94
3.4.4.2. Priorité « Head Of the Line » non préemptive ... 95
3.4.4.3. WRR ... 95
3.4.4.3.1. WRR avec poids EF/AF/BE : 6/3/1 ... 96
3.4.4.3.2. WRR avec poids EF/AF/BE : 4/4/2 ... 97
3.4.4.3.3. WRR avec poids EF/AF/BE : 6/3/1 et ingénierie de trafic ... 97
3.4.4.4. CB-WRR : Class Based WRR ... 99
3.4.4.4.1. Choix des poids du CB-WRR ... 100
3.4.4.4.2. CB-WRR : Class Based WRR pour le cas 2 ... 103
3.4.4.4.3. CB-WRR : Class Based WRR pour le cas 3 ... 104
3.4.4.4.4. Interprétation des résultats ... 104
3.4.5. Interprétation des résultats ... 105
3.5. Conclusion ... 105
CHAPITRE 4 Adaptation dynamique du chemin par la tomographie des réseaux ... 106
4.1. La tomographie dans les réseaux ... 106
4.1.1. Introduction ... 106
4.1.2. L'inférence au niveau lien et au niveau émetteur/récepteur ... 108
4.1.3. L'avantage du multicast ... 110
4.2. Méthodes d'estimation du délai ... 111
4.2.1. L'estimation complète ... 111
4.2.2. Les limitations de l'algorithme ... 113
4.2.3. L'estimation simplifiée ... 115
4.3. Implémentation ... 116
4.4. Validation ... 117
4.4.1. Le réseau expérimental ... 117
4.4.2. Comparaison des mesures et des estimations ... 119
4.4.3. Le gain acquis par la source ... 125
4.5. Ouvertures ... 127
4.6. Conclusion ... 128
Conclusions ... 129
1. Conclusions générales ... 129
2. Ouvertures et perspectives ... 130
ANNEXE 1 : Estimation MMPP ... 132
1. MLE: Maximum Likelihood Estimation ... 132
2. Chaînes de Markov Cachées ... 134
a. Définition ... 134
b. Premier exemple de HMM: Pile ou Face ... 134
c. Deuxième exemple de HMM: le modèle des balles dans l'urne ... 135
d. Les éléments d'une HMM ... 135
3. MLE pour les HMM ... 135
4. L'algorithme EM (Expectation Maximization) ... 136
5. Solution adoptée: utiliser l'algorithme EM pour MLE ... 137
a. L'algorithme proprement dit ... 137
b. Les valeurs initiales ... 138
6. Exemple d'estimation ... 141
ANNEXE 2 : Complexité des algorithmes ... 144

1. La méthode complète 144
2. La méthode simple 144
ANNEXE 3 : Exemple de calcul des probabilités stationnaires pour l'ordonnanceur WRR 146
ANNEXE 4 : Résultats complémentaires pour l'ordonnanceur WRR 148
1. Les chaînes incluses 148
2. L'évolution du délai 161
3. L'évolution du taux de perte 172
ANNEXE 5 : Résultats complémentaires de l'ordonnancement 189
1. Performance de CB-WRR dans le cas 1. 189
2. Performance de CB-WRR dans le cas 2. 191
3. Performance de CB-WRR dans le cas 3. 193
4. Performance des flux en fonction des ordonnanceurs 195
a. Le taux de perte 195
b. Le délai moyen 196
Glossaire 198
Publications 200
Bibliographie 201

Liste des figures

Figure 1: Place de l'entête MPLS 8
Figure 2: Entête MPLS 8
Figure 3: Downstream Unsolicited 9
Figure 4: Downstream on demand 10
Figure 5: La commutation de Label et les classes d'équivalence (FEC) 11
Figure 6: Entête DiffServ 12
Figure 7: Domaine DiffServ 13
Figure 8: Conditionnement aux routeurs de bordure 13
Figure 9: Génération des différentes connexions. 19
Figure 10: Emission suivant l'algorithme GPS. 19
Figure 11: Emission suivant l'algorithme WFQ. 19
Figure 12 : Emission suivant l'algorithme WF²Q. 20
Figure 13: Scénario de transmission des paquets sous EDF. 22
Figure 14: Réseau d'accès UMTS 35
Figure 15: L'architecture de l'UMTS [21]. 36
Figure 16: Graphe représentatif du CDMA 38
Figure 17: Fonctions de permission 41
Figure 18: Modèle de source voix 41
Figure 19: Exemple de génération de paquets. 43
Figure 20: Modèle de source data 44
Figure 21: Le procédé de *Backoff* et de retansmission dans 802.11. 47
Figure 22: Communication entre la source et la destination en 802.11. 48
Figure 23: Les différentes probabilités sur les axes du parallélogramme. 74
Figure 24: La chaîne incluse dans le cas général de trois classes. 75
Figure 25: Schéma d'un ordonnanceur de 3 classes 77
Figure 26: La topologie du réseau considéré 88
Figure 27: La différentiation de service dans les nœuds du réseau 89
Figure 28: La fonction de répartition du temps inter arrivée d'une distribution MMPP-2 91
Figure 29: La fonction de répartition du temps inter arrivée d'une distribution de Pareto et d'une exponentielle de même moyenne. 92
Figure 30: Délai (m sec) sans QoS 94

Figure 31: Taux de perte et gigue sans QoS..... 94
Figure 32: Délai (msec) pour la priorité absolue..... 95
Figure 33: Taux de perte et gigue pour la priorité absolue 95
Figure 34: Délai avec WRR (6/3/1) 96
Figure 35: Taux de perte et gigue pour WRR (6/3/1) 96
Figure 36: Délai pour un WRR 4/4/2. 97
Figure 37: Taux de perte et gigue pour WRR (4/4/2) 97
Figure 38: Exemple d'équilibrage de charge entre les liens pour la destination bas débit. 98
Figure 39: Délai avec les LSP 98
Figure 40: Taux de perte et gigue avec LSP 99
Figure 41: Délai des flux (1/1) et (2/1) 99
Figure 42: Délai des flux (3/1) et (4/1) 100
Figure 43: Délai des flux (5/1) et (6/1) 100
Figure 44: Délai des flux (1/1) et (2/1) 103
Figure 45: Délai des flux (3/1) et (4/1) 103
Figure 46: Délai des flux (5/1) et (6/1) 103
Figure 47: Délai des flux (1/1) et (1/2) 104
Figure 48: Délai des flux (1/3) et (1/4) 104
Figure 49: Délai des flux (1/5) et (1/6) 104
Figure 50: Noeuds, liens et chemins dans un réseau. 108
Figure 51: Exemple de tomographie dans un réseau..... 110
Figure 52: Schéma de l'étude. 112
Figure 53: Deux feuilles, trois branches..... 114
Figure 54: Trois feuilles, cinq branches 114
Figure 55: Quatre feuilles, sept branches 114
Figure 56: Les intervalles d'estimation. 114
Figure 57: En fonction de la QoS estimée, le chemin 1 ou 2 sera utilisé pour acheminer les paquets du flux source. 117
Figure 58: Déroulement temporel de l'experimentation. 119
Figure 59: Les différentes fonctions de répartition du délai à l'instant t= 48 secondes. 120
Figure 60: Les différentes fonctions de répartition du délai à l'instant t= 60 secondes. 122
Figure 61: Les différentes fonctions de répartition du délai à l'instant t=124 secondes. 122
Figure 62: Les différentes fonctions de répartition du délai à l'instant t=148 secondes 123
Figure 63: Les probabilités de dépassement sur chaque tronçon en fonction du temps et de l'algorithme d'estimation. 124
Figure 64: Les différentes fonctions de répartition du délai à l'instant t= 200 secondes. 124
Figure 65: La variation du délai moyen sur les tronçons 1 et 2 en fonction du temps et de la méthode d'estimation. 125
Figure 66: Comparaison du délai moyen du flux util en fonction du chemin suivi. 126
Figure 67: La probabilité du dépassement du seuil 0,1 seconde par le flux utile en fontion du chemin suivi. 127
Figure 68: Premier scénario 134
Figure 69: Second scénario 134
Figure 70: Convergence du cas 1 142
Figure 71: Convergence du cas 2 143
Figure 72: La chaîne incluse dans le plan *i=0, j et k* $\in$*{1..N-1}*. 148
Figure 73: La chaîne incluse dans le plan $i=N_1$, *j et k* $\in$*{1..N-1}*. 149
Figure 74: La chaîne incluse dans le plan j*=0, i et k* $\in$*{1..N-1}*. 149
Figure 75: La chaîne incluse dans le plan j=N_2, *i et k* $\in$*{1..N-1}*. 150
Figure 76: La chaîne incluse dans le plan k*=0, i et j* $\in$*{1..N-1}*. 150

Figure 77: La chaîne incluse dans le plan k=N_3, *i et j* $\in${*1..N-1*}. 151
Figure 78: La chaîne incluse sur la droite *i=0, j=0 et k* $\in${*1..*N_3*-1*}. 151
Figure 79: La chaîne incluse sur la droite *i=0, j=*N_2 *et k* $\in${*1..*N_3*-1*}. 152
Figure 80: La chaîne incluse sur la droite *i=*N_1*, j=0 et k* $\in${*1..*N_3*-1*}. 152
Figure 81: La chaîne incluse sur la droite *i=*N_1*, j=*N_2 *et k* $\in${*1..*N_3*-1*}. 153
Figure 82: La chaîne incluse sur la droite *i=0, j* $\in${*1..*N_2*-1*} *et k=0*. 153
Figure 83: La chaîne incluse sur la droite *i=0, j* $\in${*1..*N_2*-1*} *et k=*N_3. 154
Figure 84: La chaîne incluse sur la droite *i=*N_1*, j* $\in${*1..*N_2*-1*} *et k=0*. 154
Figure 85: La chaîne incluse sur la droite *i=*N_1*, j* $\in${*1..*N_2*-1*} *et k=*N_3. 155
Figure 86: La chaîne incluse sur la droite *i* $\in${*1..*N_1*-1*}*, j=0 et k=0*. 155
Figure 87: La chaîne incluse sur la droite *i* $\in${*1..*N_1*-1*}*, j=*N_2 *et k=0*. 156
Figure 88: La chaîne incluse sur la droite *i* $\in${*1..*N_1*-1*}*, j=0 et k=*N_3. 156
Figure 89: La chaîne incluse sur la droite *i* $\in${*1..*N_1*-1*}*, j=*N_2 *et k=*N_3. 157
Figure 90: La chaîne incluse au point *i=0, j=0 et k=0*. 157
Figure 91: La chaîne incluse au point *i=0, j=0 et k=*N_3. 158
Figure 92: La chaîne incluse au point *i=*N_1*, j=0 et k=0*. 158
Figure 93: La chaîne incluse au point *i=*N_1*, j=0 et k=*N_3. 159
Figure 94: La chaîne incluse au point *i=0, j=*N_2 *et k=0*. 159
Figure 95: La chaîne incluse au point *i=0, j=*N_2 *et k=*N_3. 160
Figure 96: La chaîne incluse au point *i=*N_1*, j=*N_2 *et k=0*. 160
Figure 97: La chaîne incluse au point *i=*N_1*, j=*N_2 *et k=*N_3. 161
Figure 98: Gigue des flux (1/1) et (2/1) 189
Figure 99: Gigue des flux (3/1) et (4/1) 189
Figure 100: Gigue des flux (5/1) et (6/1) 190
Figure 101: Perte des flux (1/1) et (2/1) 190
Figure 102: Perte des flux (3/1) et (4/1) 190
Figure 103: Perte des flux (5/1) et (6/1) 191
Figure 104: Taux de perte des flux (1/1) et (2/1) 191
Figure 105: Taux de perte des flux (3/1) et (4/1) 191
Figure 106: Taux de perte des flux (5/1) et (6/1) 192
Figure 107: La gigue des flux (1/1) et (2/1) 192
Figure 108: La gigue des flux (3/1) et (4/1) 192
Figure 109: La gigue des flux (5/1) et (6/1) 192
Figure 110: Taux de perte des flux (1/1) et (1/2) 193
Figure 111: Taux de perte des flux (1/3) et (1/4) 193
Figure 112: Taux de perte des flux (1/5) et (1/6) 193
Figure 113: La gigue des flux (1/1) et (1/2) 194
Figure 114: La gigue des flux (1/3) et (1/4) 194
Figure 115: La gigue des flux (1/5) et (1/6) 194
Figure 116: Perte EF 195
Figure 117: Taux de perte AF 195
Figure 118: Perte BE 196
Figure 119: Le délai EF 196
Figure 120: Délai AF 197
Figure 121: Délai BE 197

Liste des tableaux

Tableau 1: Exemple d'algorithme EDF. 22
Tableau 2: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WRR avec une pondération 6/3/1..... 24
Tableau 3: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WF^2Q avec une pondération 6/3/1. 25
Tableau 4 : La variation du délai moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur SCFQ avec une pondération 6/3/1..... 26
Tableau 5: La variation du taux de perte moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WRR avec une pondération 6/3/1. 27
Tableau 6: La variation du taux de perte moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WF^2Q avec une pondération 6/3/1. 28
Tableau 7 : La variation du taux de perte moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur SCFQ avec une pondération 6/3/1. 29
Tableau 8: La variation de la variance du délai en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WRR avec une pondération 6/3/1..... 30
Tableau 9: La variation de la variance du délai en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WF^2Q avec une pondération 6/3/1. 31
Tableau 10 : La variation de la variance du délai en fonction de la charge des classes 1 et 2 pour l'ordonnanceur SCFQ avec une pondération 6/3/1. 32
Tableau 11: Valeurs des différents Paramètres. 43
Tableau 12: Les éléments recherchés..... 50
Tableau 13: La variation du débit utile pour des utilisateurs voix. 54
Tableau 14: Les inverses des fonction de répartition et les erreurs quadratiques moyennes des modèles MMPP-2, MMPP-4 et Gaussien pour les utilisateurs voix. 54

Tableau 15: La variation du débit utile pour des utilisateurs Web. 55
Tableau 16: Les inverses des fonction de répartition et les erreurs quadratiques moyennes des modèles MMPP-2, MMPP-4 et Gaussien pour les utilisateurs Web. 55
Tableau 17: La variation du débit utile pour des utilisateurs FTP. 56
Tableau 18: Les inverses des fonction de répartition et les erreurs quadratiques moyennes des modèles MMPP-2, MMPP-4 et Gaussien pour les utilisateurs FTP. 56
Tableau 19: La variation du débit utile pour un multiplexage d'utilisateurs. 57
Tableau 20: Les inverses des fonction de répartition et les erreur quadratique moyenne des modèles MMPP-2, MMPP-4 et Gaussien pour un multiplexage d'utilisateurs. 57
Tableau 21: Caractéristiques du réseau 802.11 utilisé. 58
Tableau 22: La variation des caractéristiques et leurs valeurs moyennes au niveau de la couche MAC d'une base 802.11 pour des connexions voix. 59
Tableau 23: Les inverses des fonctions de répartition et les erreurs quadratiques moyennes des modèles MMPP-2 et Gaussien de l'inter arrivée des paquets au niveau de la couche MAC d'une base 802.11 pour des connexions voix. 60
Tableau 24: La variation des caractéristiques et leurs valeurs moyennes au niveau de la couche MAC d'une base 802.11 pour des connexions FTP. 61
Tableau 25: Les inverses des fonctions de répartition et les erreurs quadratiques moyennes des modèles MMPP-2 et Gaussien de l'inter arrivée des paquets au niveau de la couche MAC d'une base 802.11 pour des connexions FTP. 62
Tableau 26: La variation des caractéristiques au niveau de la couche MAC d'une base 802.11 pour un multiplexage de connexions voix et FTP. 63
Tableau 27: Représentation graphique des valeurs moyennes des caractéristiques au niveau de la couche MAC d'une base 802.11 pour un multiplexage de connexions voix et FTP. 68
Tableau 28: Résultats d'estimations avec un modèle suivant une distribution MMPP-2 et Gaussienne de l'inter arrivée des paquets au niveau de la couche MAC d'une base 802.11 pour un multiplexage de connexions voix et FTP. 69
Tableau 29: Erreur quadratique moyenne pour un multiplexage de connexions voix et FTP. 71
Tableau 30: Les taux de service des classes en présence des autres classes. 74
Tableau 31: Les 26 différentes équations autour des points, des droites et des plans particuliers. 76
Tableau 32: Les différents poids utilisés pour les trois classes. 77
Tableau 33: Les charges totales utilisées. 78
Tableau 34: Les taux relatifs des trois classes 78
Tableau 35: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1 80
Tableau 36: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1 81
Tableau 37: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2 82
Tableau 38: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1 85
Tableau 39: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1 86
Tableau 40: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2 87
Tableau 41: Les caractéristiques des Sauts à Jetons 90
Tableau 42: Caractéristiques des flux UMTS 91
Tableau 43: Les paramètres MMPP-2 des flux UMTS 91
Tableau 44: Les débits des sources par QoS. 92

Tableau 45: Les débits des émissions 93
Tableau 46: Les états simulés..... 93
Tableau 47: Proportion des trafics 93
Tableau 48: Les débits absolus et relatifs des sources par QoS dans les cas simulés. 100
Tableau 49: La matrice de trafic. 101
Tableau 50: Les débits des sources par QoS. 101
Tableau 51: Les états simulés..... 101
Tableau 52: Proportion des trafics 101
Tableau 53: Les débits des sources par QoS. 102
Tableau 54: La matrice de trafic. 102
Tableau 55: Les états simulés..... 102
Tableau 56: Proportion des trafics 102
Tableau 57: Les caractéristiques des liens utilisés dans le schéma de validation. 118
Tableau 58: Les caractéristiques des différents flux. 118
Tableau 59: Délais moyens à partir de l'instant t= 50,6 secondes..... 121
Tableau 60: Estimation des paramètres..... 142
Tableau 61: Les équations de Kolmogorov pour le système avec N_1=1, N_2=1 et N_3=1. 146
Tableau 62: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Poisson. 162
Tableau 63: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Poisson. 163
Tableau 64: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Poisson. 164
Tableau 65: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Poisson. 165
Tableau 66: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Poisson. 166
Tableau 67: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Pareto. 167
Tableau 68: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Pareto. 168
Tableau 69: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Pareto. 169
Tableau 70: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Pareto. 170
Tableau 71: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Pareto. 171
Tableau 72: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Poisson. 173

Tableau 73: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2 avec une distribution de la taille des paquets suivant la loi de Poisson. 174
Tableau 74: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Poisson. 175
Tableau 75: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Poisson. 176
Tableau 76: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Poisson. 177
Tableau 77: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1 avec une distribution de la taille des paquets suivant la loi de Poisson. 178
Tableau 78: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Poisson. 179
Tableau 79: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1 avec une distribution de la taille des paquets suivant la loi de Poisson. 180
Tableau 80: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Pareto. 181
Tableau 81: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2 avec une distribution de la taille des paquets suivant la loi de Pareto. 182
Tableau 82: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Pareto. 183
Tableau 83: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Pareto. 184
Tableau 84: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Pareto. 185
Tableau 85: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1 avec une distribution de la taille des paquets suivant la loi de Pareto. 186
Tableau 86: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Pareto. 187
Tableau 87: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1 avec une distribution de la taille des paquets suivant la loi de Pareto. 188

Introduction

La *convergence* des réseaux en un seul et unique réseau qui puisse servir tous types d'applications favorise l'émergence du protocole IP (Internet Protocol). Elle lui permet aussi de devenir de plus en plus l'élément fédérateur des réseaux de transport des données. En effet, la simplicité de sa gestion, de son déploiement et son caractère d'adaptabilité ainsi que le coût réduit de son infrastructure le rendent très facile à généraliser. De ce fait, les applications de la voix sur IP (VoIP Voice over IP), de la diffusion de télévision ou de la vidéo sur les réseaux en générale et sur Internet en particulier, ne cessent de croître. Par ailleurs, les nouvelles spécifications ETSI du réseau UMTS (Universal Mobile Telecommunication Service), qui devront être basées sur le protocole IP, ne vont qu'accentuer cette tendance.

1. Cadre générale

Bien qu'initialement prévu pour un service sans aucune garantie, l'Internet actuel achemine des paquets de toutes sortes d'applications: des applications d'accès à un serveur (FTP, Mail, P2P, …), des applications interactives entre deux utilisateurs ou plus (Téléphonie, visiophonie, …) ou entre un utilisateur et un serveur (Web, télévision à la demande, accès à une base de donnée centralisée distante, …) … Cependant les marges des valeurs tolérées des délais entre l'émission et la réception des informations échangées, des variations de ce délai, noté gigue, et les taux de pertes des paquets reçus, sont très variables en fonction de l'application. Tout au long de cette thèse, nous engloberons ces différents aspects par l'acronyme QoS (Quality of Service). Les applications que nous désignerons par temps réel stricte (Téléphonie, visiophonie, …) nécessitent des délais réduits, de l'ordre des centaines de millisecondes, ainsi que des gigues faibles de l'ordre des dizaines de millisecondes. Par contre, grâce à la capacité de compréhension et d'interpolation des sens humains, ces applications peuvent tolérer des pertes d'informations allant jusqu'à quelques pourcents. Contrairement aux applications temps réels, les applications interactives (Consultation de pages Web, écoute de la musique en ligne, visualisation de télévision en ligne …) réclament des temps de réponses bornés mais pas réduits. Ils se chiffrent à quelques secondes. De plus, les buffers dans les terminaux d'extrémités permettent de combler aux variations des délais. Cependant, le taux d'erreur n'est pas apprécié: chaque paquet perdu engendrera des mécanismes de signalement d'erreur et de retransmission qui produiront des retards élevés voire même un arrêt de la transmission jusqu'à la réception de l'information manquante. Quant aux applications d'échange de fichiers ou de courriers électroniques, tous les administrateurs de réseau sont d'accord pour leur donner le reste des ressources disponibles et ainsi de ne leur garantir aucune QoS.

Poussée par plusieurs facteurs économiques, sociales, géographiques et politiques, la mobilité s'associe avec la disponibilité des personnes pour créer le nouveau profil de clients. C'est un des principaux éléments qui ont permis la croissance des réseaux mobiles de première et deuxième génération (GSM, DCS, …) à travers le monde. En associant d'autres services que la téléphonie aux terminaux mobiles, nous nous trouvons dans un *réseau de donnée mobile*. Nous pouvons citer par exemple les réseaux en cours de normalisation et de déploiement l'UMTS ou le cdma2000 qui répondent à ces critères. Par la composition en cellules, ces réseaux offrent de larges couvertures: à partir de petits villages isolés jusqu'aux grandes agglomérations métropolitaines. Avec la possibilité de *handover*, qui permet à un

utilisateur de changer de cellule tout en restant connecté, ces réseaux offrent un service aux usagers comme s'ils étaient fixes. Par ailleurs, la limitation du débit maximal qui peut atteindre jusqu'à 6Mb/sec, ainsi que le coût de l'infrastructure pénalisent ces réseaux. Néanmoins, pour les points d'accès très fréquentés –*hot spots*, comme les aéroports, les gares, les centres de congrès, etc. … d'autres possibilités de connexions peuvent exister. Ce sont les réseaux *WiFi* (Wireless Fidelity) basés sur le protocole 802.11 de l'IEEE (Institute of Electrical and Electronics Engineers) qui peuvent jouer le rôle de réseaux haut débit sans fil. Bien qu'actuellement ce réseau ne puisse pas supporter ni le handover ni de garantir la QoS, les travaux des groupes de recherches de l'IEEE étudient les modifications à apporter pour combler ce manque. Les normes en cours d'élaboration 802.11f et 802.11e répondent respectivement à ces questions.

Comme nous venons de le présenter, dans le réseau Internet du futur transitera toute sorte d'informations voix, vidéo, Web, mail, FTP… provenant de plusieurs réseaux d'accès UMTS, GPRS, 802.11x, ISP, ADSL, satellite, LAN… Toutefois, les réseaux d'accès hétérogènes ont une influence primordiale sur les propriétés statistiques des paquets: les distributions du temps inter arrivée, de la taille des paquets… Or comme la QoS sera un composant important à garantir, il faudra assurer des performances strictes et mettre en place une gestion de bout en bout des flux. Dans le cadre des réseaux mobiles (GPRS, UMTS, IEEE 802.11x…), c'est l'accès radio sans fil qui présente le plus de contraintes vu l'aspect partagé et rare des ressources.

Le premier réseau paquet large bande à offrir une QoS à ces utilisateurs et à proposer plusieurs classes de services afin de traiter différemment les flux est l'ATM (Asynchronous Transfer Mode). Ce réseau à cellule commuté permet de créer des chemins –*path*, de bout en bout et ainsi servir les paquets de chaque flux en fonction de sa classe de service d'appartenance. Toutefois ce réseau présente quelques inconvénients dont, les plus fréquemment cités sont d'une part l'introduction d'un surplus d'entête de 5 octets toutes les 48 octets utiles formant la cellule ATM, et d'autre part, l'ajout d'un nouveau plan de contrôle et de gestion dans le réseau.

Une autre technique de gestion de la QoS qui sera introduite plus tard par l'IETF est IntServ (Integrated Services). Cette technique permet effectivement de traiter les flux des paquets en fonction de la demande de la source juste avant de démarrer l'envoi des paquets utiles. IntServ ne nécessite pas d'un nouveau plan de contrôle mais uniquement le développement d'une signalisation adéquate RSVP (Resource ReserVation Protocol). Cependant, ce schéma se heurte à un autre problème, celui du facteur d'échelle. Avec IntServ, chaque routeur dans le réseau doit garder l'état de chaque flux qui y transite jusqu'au moment où la liaison s'achève. En remarquant que plusieurs milliers de flux peuvent passer sur un routeur d'un cœur de réseau, nous réalisons immédiatement pourquoi une telle technique ne peut être utilisée dans le cœur des réseaux haut débit. A partir de ces différentes constatations, et en accord avec les différents partenaires du projet RNRT (Réseau National de Recherche en Télécommunications) Esquimaux (Estimation de la qualité de service d'une architecture Internet audio-visuel sur modélisation des flux) nous avons adopté le protocole MPLS (Multi Protocol Label Switching) et le mécanisme DiffServ (Differentiated Services) dans notre architecture de réseau étudié. MPLS a été conçu pour d'une part créer des chemins commutés dans le réseau IP et d'autre part pour pouvoir implémenter les avantages de ATM liés à l'ingénierie de trafic. DiffServ est un mécanisme de différentiation de classes au niveau de chaque routeur. Il résout le problème du facteur d'échelle de IntServ en définissant un nombre limité de comportement PHB (Per Hop Behaviour) au niveau de chaque nœud. Le Chapitre 1 présente plus en détails les avantages de MPLS et de DiffServ.

La variabilité de la charge des flux de trafics sur chaque lien formant le chemin de la source à la destination influe sur la QoS de la liaison de bout en bout empruntée par les paquets. En plus, l'état de disponibilité des liaisons entre les routeurs peut être un facteur déterminant dans le choix d'un chemin ou d'un autre. Ainsi, même si différentes techniques de différenciation de service sont effectivement implémentées dans tout le réseau, la QoS n'est pas garantie tout au long du chemin. Ces différents constats nous ont amené à s'intéresser à la détermination de la QoS sur les différents tronçons formant un chemin entre une source et une destination. Avec cette cartographie du réseau, chaque source pourra créer, si elle est autorisée, la route qui lui convient pour acheminer les différents flux vers la destination.

2. Objectifs de la thèse

Après avoir exposé précédemment les différentes problématiques qu'un cœur de réseau Internet devra affronter pour être capable de transporter différentes sortes de flux, nous définissons ci-dessous les objectifs que nous avons essayé de cibler tout au long de cette thèse. Afin de faciliter la lecture du manuscrit, nous avons divisé nos objectifs en quatre grands axes qui seront les centres d'intérêts des chapitres respectivements numérotés.

1- A partir des différents besoins sollicités d'un cœur de réseau Internet de prochaine génération, nous avons voulu définir l'architecture et les technologies qu'il faudra mettre en place pour répondre aux exigences requises.
2- Nous cherchons à déterminer les caractéristiques statistiques et temporelles de l'inter arrivée des paquets entrants/sortants d'un réseau mobile sans fil. Nous avons pour cela étudié deux réseaux: le réseau WiFi IEEE 802.11 qui est adapté à une transmission sans fil à très haut débit et le réseau UMTS qui sera, quant à lui, mieux utilisé par des clients à mobilité élevée.
3- A partir de plusieurs paramètres réseaux, nous essayerons d'étudier les techniques de différentiation de service proposées et nous présenterons leurs impacts sur la QoS observée. Nous tenterons de trouver un modèle analytique qui pourra exprimer les règles à tirer afin d'ajuster les paramètres et les politiques qu'il sera recommandé d'implémenter par les ordonnanceurs pour tirer la meilleure rentabilité du réseau.
4- Nous chercherons des solutions innovantes pour remédier aux problèmes de la variation de la QoS sur un chemin donné. Nous essayerons essentiellement de déterminer la route qui minimise le délai de traversée.

3. Contributions de la thèse

Nos contributions se repartissent en trois grands thèmes:

1- Nous avons caractérisé la distribution des lois de l'inter arrivée des paquets formés par l'agrégation de flux homogènes et hétérogènes arrivant ou sortant de deux réseaux mobiles UMTS et 802.11. Ces caractéristiques sont étudiées en fonction du nombre de clients connectés et du type de service requis.
2- Nous avons étudié les effets de l'implémentation des mécanismes de QoS dans un cœur de réseau IP. Nous avons ciblé nos travaux sur le protocole MPLS, le mécanisme DiffServ de différenciation de service et les algorithmes d'ordonnancement. Nous avons ainsi étudié les performances dans un cœur de réseau de l'arrivée de différents modèles d'agrégats de sources Web, voix ou FTP provenant de sources mobiles ou fixes. Nous comparons ensuite plusieurs ordonnanceurs pour tirer des règles pour respecter les QoS recherchées par les divers flux.

3- Nous avons proposé un mécanisme d'adaptation dynamique de routage afin de garder le chemin qui offre la meilleure QoS. Les critères de QoS que nous avons considéré sont le délai moyen et la probabilité de dépassement du délai d'un seuil maximal.

4. Plan de la thèse

Dans le paragraphe suivant, nous décrivons les différentes parties de ce manuscrit.

Nous entamons nos travaux de recherche par le premier chapitre de cette thèse, où nous décrivons les différentes techniques, technologies et protocoles utilisés actuellement dans les réseaux IP pour garantir la QoS des paquets et des flux. Nous illustrons premièrement le protocole MPLS qui présente plusieurs similitudes avec les réseaux ATM. Comme ATM, MPLS est capable de contrôler et de gérer la QoS. Il définit aussi différentes options d'ingénierie de trafic mais sans les contraintes inhérentes à ATM notamment liées d'une part aux 10% de trafic ajouté par les entêtes et d'autre part à la complexité crée par un nouveau plan de contrôle et d'administration. Nous continuons le premier chapitre qui est notre état de l'art, par la description des mécanismes de DiffServ. Cette dernière norme qui est proposée par l'IETF (Internet Engineering Task Force) dans le RFC 2474 [34] permet une différenciation du traitement reçu au niveau de chaque nœud. Nous présentons ensuite le RFC 3270 qui associe MPLS et DiffServ. Comme nous le signalons, cette association est beaucoup plus fructueuse qu'une simple superposition puisqu'elle permet la création de chemin MPLS en fonction de la classe DiffServ. Nous entamons ensuite une description détaillée de divers algorithmes d'ordonnancement. Nous retenons ceux qui répondent le mieux aux besoins de notre réseau.

Nous poursuivons notre travail par le Chapitre 2 avec une modélisation du trafic au niveau de la couche MAC des réseaux UTRAN (UMTS Terrestrial Radio Access Network) et 802.11. Ce chapitre comporte une présentation détaillée de ces deux protocoles ainsi que les algorithmes d'estimation utilisés. Nous présentons aussi la couche CDMA (Code Division Multiple Access) du niveau MAC de l'UMTS proposé par les travaux de [5]. Cette couche offre un accès garantissant une QoS différente en fonction du service demandé par l'application. Notre étude a porté exclusivement sur la loi de distribution de l'inter arrivée des paquets, vu qu'après segmentation et réassemblage, la taille de chaque paquet au niveau de la station de base est la même qu'au niveau de la couche réseau du mobile ou du réseau filaire. Donc la loi qui décrit la variation de la taille des paquets est la même, avec une possibilité de croissance de sa moyenne par l'introduction d'éventuelle entête. Dans le cas d'un multiplexage de flux la distribution de la taille des paquets est représentée par une convolution des lois unitaires. Nous nous sommes alors intéressés à l'agrégation des flux voix, Web, FTP et au multiplexage de ces différents flux existants. En accord avec différents travaux de recherche précédents, nous avons tout d'abord essayé une modélisation MMPP. Avant d'aboutir à un algorithme efficace, fiable et précis pour tirer les différents paramètres de ce processus, nous avons dû développer nos connaissances statistiques dans le domaine de l'estimation, ce qui nous conduit à plonger dans les chaînes de Markov cachées et la maximisation de vraisemblance. Nous avons ensuite essayé d'appliquer le modèle Gaussien pour modéliser cette agrégation des flux.

Nous nous sommes ensuite penchés sur une étude analytique des performances d'une file implémentant l'ordonnanceur GPS (Generalised Processor Sharing) dans le Chapitre 3. Nous avons complété nos travaux par une étude du délai observé à la sortie de cette file en fonction de la pondération de chaque classe de service, de la charge introduite par les différents flux et de la distribution de la taille des paquets. Nous avons alors appliqué les résultats obtenus sur un réseau maillé de prochaine génération implémentant MPLS et DiffServ afin de mesurer la variation des performances en utilisant telle ou telle autre

méthode de QoS. Ces deux dernières études ont été réalisées sur une plateforme de test utilisant le simulateur de réseaux NS (Network Simulator).

Suite à nos constats que la QoS dépend fortement du chemin suivi et de la charge créée par les autres trafics sur les différents liens empruntés par le flux, nous avons étudié dans le Chapitre 4 une méthode d'adaptation dynamique du chemin en fonction de la QoS observée sur les différents tronçons des routes considérées. Nous associons la tomographie des réseaux et les algorithmes d'estimations pour mesurer la variation des délais sur chaque tronçon du réseau. Le chemin qui propose le délai de bout en bout le plus faible est retenu.

Nous concluons notre manuscrit en résumant nos travaux de recherche et en suggérant diverses pistes qui en découlent et qu'il sera intéressant de poursuivre.

CHAPITRE 1
La garantie de la qualité de service dans un réseau IP

1.1. Introduction

Dans nos jours, les opérateurs, les intégrateurs et les ISP, quelque soit leur taille, font constamment face à des défis d'adaptation de leurs réseaux afin de répondre aux besoins de plus en plus accrus de leurs clients. La maîtrise de la QoS revêt alors une importance capitale. L'assurance de cette maîtrise passe par une évolution aussi bien par une garantie au niveau des moyens de transport, notamment en augmentant les débits des liens, qu'au niveau de la mise en place des protocoles de communication adéquats. Un des protocoles de communication les plus promettant pour aider à la maîtrise de la QoS est le Multi Protocol Label Switching (MPLS) qui offre des mécanismes simples pour l'acheminement des flux dans les réseaux orientés paquets. Un autre mécanisme prometteur est DiffServ (Differentiated Services). Après classification des flux en plusieurs classes, il traite les classes présentes différemment. C'est en fonction de ce traitement distinct sur chaque nœud DiffServ, que la QoS des trafics prioritaires sera améliorée et la QoS des trafics moins prioritaires sera obtenue. Le protocole RSVP (Resource ReserVation Protocol) met en place des réservations de ressources pour chaque flux au niveau de chaque routeur capable du réseau afin de garantir la QoS de bout en bout. Or plusieurs études ont montré que le facteur d'échelle de RSVP est très pénalisant. De plus, un nombre réduit de classes de service suffit pour donner la QoS recherchée à l'ensemble des applications existantes. Ces deux observations nous ont contraints à étudier uniquement le mécanisme DiffServ dans cette thèse.

Ce chapitre présente 5 grands axes de recherche. Le paragraphe 2 décrit les aspects les plus importants de MPLS. Comme la normalisation de ce dernier n'est pas encore achevée, nous nous contenterons de présenter les volets complétés et nous indiquerons ensuite les orientations en cours de normalisation. Le troisième paragraphe présente les différentes possibilités d'utilisation de DiffServ. Dans le paragraphe quatre, nous résumerons la normalisation de l'interaction entre DiffServ et MPLS. Le cinquième paragraphe décrit quelques algorithmes d'ordonnancement de file d'attente, avec lesquels sera réalisée la différenciation de service de chaque nœud du réseau.

1.2. Le protocole MPLS

Dans le but de construire les tables de routages, les routeurs présents dans un réseau IP communiquent entre eux à travers des protocoles de routage. Les entrées de ces tables changent selon la technique de routage utilisée. Par exemple, pour la technique « Next Hop Routing », les entrées sont : l'adresse de destination, l'adresse du prochain routeur et le port de sortie. Le premier protocole de routage utilisé sur Internet était le protocole RIP (Routing Information Protocol) (RFC 2453). Celui-ci ne permet la synchronisation que de 16 routeurs à la fois. L'amélioration de RIP conduit au protocole OSPF (Open Shortest Path First) (RFC

2328) qui tient compte de la charge des liaisons. Un protocole concurrent de OSPF est IS-IS (Intermediate System-Intermediate System), une norme de l'OSI (norme de l'ISO 10589 et les RFC: RFC 1195, RFC 3719, RFC 3787 et RFC 3784).

Il est possible de définir et d'administrer plusieurs domaines de routage. Chaque domaine de routage est appelé systèmes autonomes (AS). Le routage inter AS se fait via des routeurs passerelles qui communiquent entre eux par le protocole BGP (Border Gateway Protocol) (RFC 1771), selon des accords conclus entre les opérateurs qui gèrent les AS. Pour faciliter la gestion, plusieurs AS peuvent être gérés par le même opérateur. Le protocole de routage utilisé est alors le BGP intérieur.

A la fin des années 1990, l'infrastructure a été revue suite à l'apparition de l'ATM (Asynchronous Transfer Mode): les routeurs IP sont désormais aux bordures des réseaux et raccordés entre eux au cœur par des commutateurs ATM. Bien que l'ingénierie du trafic s'est beaucoup développée grâce à ATM par l'introduction des circuits virtuels ainsi que par la possibilité de réaliser des statistiques par circuit et d'équilibrer la charge entre différentes liaisons, des limitations subsistent. D'une part, il fallait gérer simultanément deux types de réseaux (IP et ATM) et d'autre part, les entêtes de ses cellules ATM introduisent une perte de 10% de la bande passante.

Lorsque le trafic dépasse un certain seuil, le fonctionnement dans le réseau IP se dégrade compte tenu de la capacité limitée des routeurs. Les files d'attente se remplissent et conduisent les flux à un délai supplémentaire à la transmission. De plus, des paquets se voient rejeter lorsqu'il y a débordement des files d'attente dans les routeurs. Pour pallier à ces problèmes, l'ingénierie du trafic Internet a eu longtemps recours au surdimensionnement avec une utilisation des métriques afin de favoriser le routage sur des liens particuliers. Cependant, cette méthode admet des limitations du fait qu'elle sur-utilise certains liens et sous-utilise d'autres. De même, c'est une approche empirique qui n'est pas garantie à long terme.

A partir des ces différentes constatations, nous pouvons tirer plusieurs objectifs recherchés qu'il faudra satisfaire:

- L'amélioration des performances de routage IP et de son expansion.
- L'ingénierie du trafic IP: faciliter le routage explicite et garantir des contraintes sur les routes empruntées.
- L'introduction des VPN (Virtual Private Network) et des mécanismes de contrôles des tunnels.
- Le contrôle de la QoS (délai de transmission, débit, …) pour les applications du type voix ou vidéo sur IP.
- Développer un algorithme unique de commutation capable de supporter différents algorithmes de routage.

Ces différentes contraintes ont amené l'IETF à la normalisation et le développement de MPLS. C'est un protocole qui répond largement aux différentes contraintes citées plus haut grâce notamment à son facteur d'échelle favorable et ses possibilités d'interagir avec IP.

Deux applications principales ont favorisé le déploiement de MPLS. D'une part ce sont ses différentes capacités d'ingénierie de trafic. C'est un aspect souhaité par les fournisseurs de réseau qui veulent optimiser les coûts de leur investissement. D'autre part, c'est sa facilité de création des VPN. Cet aspect est plus recherché par les entreprises pour pouvoir relier plusieurs sites distants et autonomes.

1.2.1. Entête MPLS

L'entête MPLS, noté parfois entête "shim", se situe entre les entêtes des couches 2 et 3 (c.f. Figure 1).



Figure 1: Place de l'entête MPLS

Schématisée à la Figure 2, l'entête MPLS dispose d'une longueur de 32 bits formée des champs suivants [18]:

- Un *Label* de 20 bits. Cette étiquette courte de longueur fixe marque les paquets à transférer sur le même chemin. Le contenu d'un label n'est pas structuré et n'a de signification que localement, entre les deux nœuds qui l'utilisent.
- Le champ *Exp* (expérimental) de 3 bits, connu sous le nom *CoS*. Il peut affecter les algorithmes d'ordonnancement (Scheduling) et/ou de la gestion de la file (Queue Management). Il peut être exploité aussi lors de l'utilisation combinée de DiffServ et MPLS.
- Un bit *S* d'empilement (stack) pour supporter une hiérarchisation des entêtes.
- Un champ *TTL* (Time To Live) de 8 bits indiquant la durée de vie.

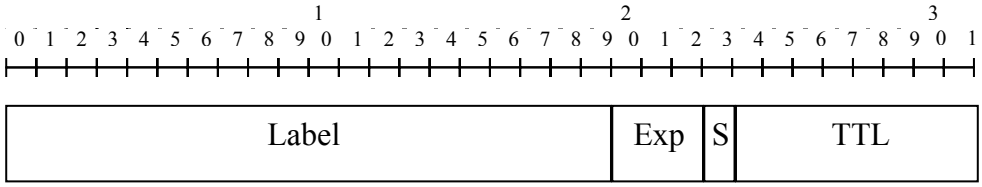


Figure 2: Entête MPLS

Le champ TTL est copié du champ TTL du paquet IP quand celui-ci vient d'être labellisé. Quand le paquet quitte le réseau MPLS, le champ TTL de l'entête MPLS remplace le TTL du paquet IP.

Le label MPLS est ajouté au paquet au niveau du LER (Label Edge Router) d'entrée (Ingress LER) et retiré (désencapsulé) par le LER de sortie (Egress LER). Le LER pourra chercher des informations dans le numéro de port d'application, l'interface d'entrée, etc… pour allouer un Label.

1.2.2. Label Switch Path (LSP)

Sans aucune information supplémentaire, chaque LSP crée, qui est un chemin unidirectionnel entre deux paires de nœud, suit l'itinéraire indiqué par le protocole de routage déployé dans le réseau. Ainsi, si aucune contrainte n'est imposée, les LSP passeront par la même route comme si un routage normal avait été utilisé. Cependant, avec MPLS, les LSP peuvent être créés à des fins précis. Dans ce dernier cas, les LSP devront être bien contrôlés, vu qu'ils doivent répondre à des besoins spécifiques du réseau.

1.2.2.1. La création des LSP

La création des LSP peut être faite de deux manières différentes: soit manuellement soit en utilisant une signalisation. La création manuelle crée des LSP statiques, inadaptable en cas de panne d'une connexion. De plus, les différents Labels utilisés devront être attribués manuellement sur chaque LSR. Par contre, la création par la signalisation ne nécessite pas d'intervention sur chaque nœud et s'adapte en cas d'une chute d'une liaison. Deux protocoles sont utilisés pour créer les LSP: RSVP-TE (une extension de RSVP) (RFC 3209) et LDP (RFC 3036).

Dans le cas de l'utilisation de la signalisation pour créer les LSP, deux types de LSP peuvent exister: d'une part les LSP sans contrainte de passage par des nœuds et d'autre part les LSP qui devront passer par des LSR particuliers. L'arbre des LSP des premiers est construite à partir des tables de routage, via les protocoles IGP (Interior Gateway Protocol) tels SPF, IS-IS, RIP… Le second type de LSP noté ER-LSP (Explicite Routed LSP) suit le chemin que le LSR source a choisi. Il s'agit d'un routage de source où l'acheminement est déterminé par l'émetteur. Ces LSR sont synonymes des *Permanant Virtual Chanel* (PVC) de Frame Relay ou ATM. Dans le cas où **tout** le chemin est spécifié, nous parlerons de routage strict. Néanmoins, MPLS propose plus de flexibilité: uniquement une partie du chemin d'un LSP peut être constitué de LSR prédéfinis. Le reste du chemin empruntera la route qui liera ces LSR pour former le LSP noté *lâche*. Il est à noter que le routage de source lâche nécessite moins de configuration qu'un routage de source stricte. De plus, s'il est nécessaire, la route découverte par le routage intrinsèque au réseau pourrait être figée. Ce dernier schéma est noté par le "*route pinning*". Pour une flexibilité de l'ingénierie de trafic, il est possible de construire des ER-LSP en plus des LSP automatiquement construits.

Un autre type de signalisation porte des informations concernant les caractéristiques de QoS des LSP. Si ces conditions sont définies, chaque LSR du réseau devra s'assurer que les caractéristiques sont maintenues. Ainsi, ce sera possible de garantir les performances de bout en bout. Les caractéristiques définies pour les LSP peuvent englober :

- La bande passante, incluant la BP soutenue, maximale, et la taille maximale du burst.
- Le délai et la gigue.
- La sélection du lien.

Ces caractéristiques sont implémentables grâce à l'extension du protocole LDP en CR-LDP [RFC 3212 et 3214] et à l'ajout de nouvelles extensions à RSVP-TE [RFC 3473].

1.2.2.2. Les méthodes de distribution de label

Il existe deux méthodes de distribution d'association FEC-Label [17]:

- La première est « Downstream unsolicited »

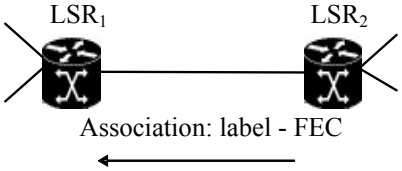


Figure 3: Downstream Unsolicited

LSR_1 et LSR_2 sont deux LSR adjacents. Afin de découvrir le « next hop » pour un FEC donné, LSR_2 crée un label pour cette FEC et communique par LDP l'association à LSR_1.

LSR_1 insère l'association dans sa table de commutation. Si pour une demande de connexion donnée, LSR_2 est le "next hop" pour ce FEC, LSR_1 peut alors utiliser le label.

- La seconde est la distribution "Downstream on demand"

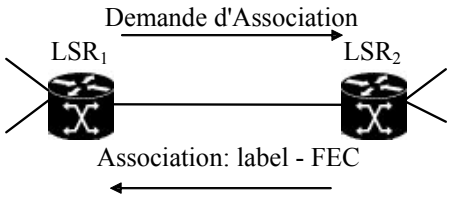


Figure 4: Downstream on demand

LSR_1 reconnaît LSR_2 comme son "next hop" pour un FEC donné. Une demande est alors faite à LSR_2 par LSR_1 pour créer un label pour ce FEC. Si LSR_2 reconnaît le FEC et a lui-même un "next hop", il crée alors un label et le communique à LSR_1.

Ces deux méthodes peuvent être présentes dans un réseau, voir en même temps entre deux LSR. La première méthode utilise beaucoup de signalisations en créant toutes les associations possibles entre LSR adjacents avec toutes les FEC possibles. Mais l'avantage est que si un flux arrive, il sera immédiatement acheminé. Par contre, dans la seconde méthode, la réservation des Labels est effectuée quand un flux demande à être acheminé. Donc, elle consomme moins de ressources pour les réservations mais admet un temps de latence plus grand pour le premier paquet de la connexion.

1.2.2.3. Utilisation des LSP

Une fois les LSP créés, ils pourront acheminer les paquets. Nous suivrons ici un paquet IP à partir de son entrée dans un réseau MPLS jusqu'à sa sortie.

En premier lieu, quand le paquet entre dans un LER, ce dernier déterminera sa QoS et son adresse de sortie. En se basant sur la table de routage et la politique implémentée dans le réseau, le LER sélectionne un label pour l'appliquer au paquet. Une fois marqué par ce label, le LSP que le paquet empruntera sera déterminé. Dans le cœur du réseau, chaque LSR lit le label du paquet. Il le remplacera par un nouveau label comme indiqué dans sa table de commutation et le renverra sur le lien correspondant. Une fois arrivée au LER de destination, ce dernier désencapsule le label, et suivant l'adresse IP, il renvoie le paquet à sa destination finale. Ainsi, les LSP permettent aux paquets IP d'être routés simplement aux extrémités du réseau. Afin de faciliter l'implémentation de la QoS, des classes d'équivalence de transfert (FEC Forwarding Equivalent Class) ont été mises en place. C'est un ensemble de paquets empruntant le même chemin et demandant le même traitement dans le réseau. Sur chaque liaison, les paquets d'une même classe disposent du même label (Figure 5). Actuellement, le mécanisme d'association du FEC au label est basé sur le préfixe de destination IP. Le concept FEC entraîne donc une flexibilité et une rapidité dans le transfert.

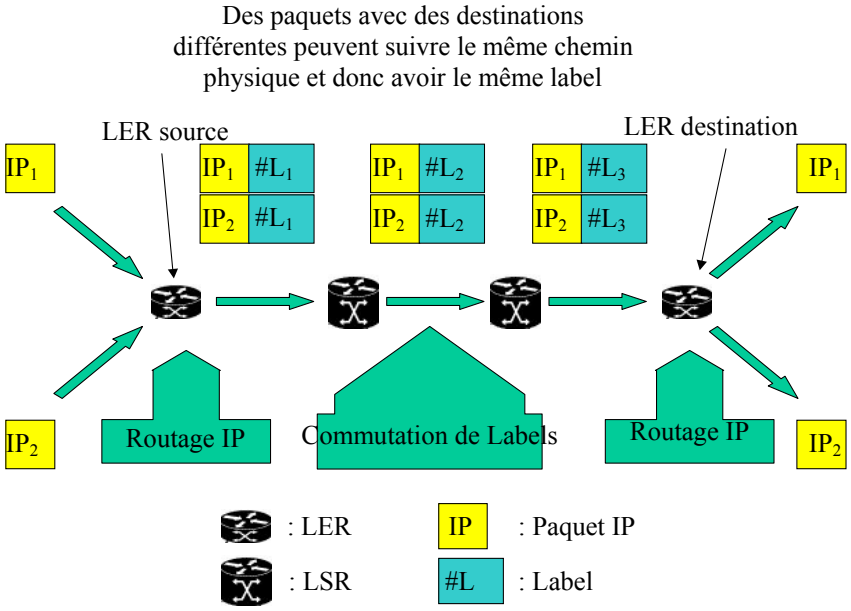


Figure 5: La commutation de Label et les classes d'équivalence (FEC)

1.2.3. Application des LSP: Ingénierie de trafic MPLS

L'apport de l'ingénierie de trafic est de permettre à l'administrateur du réseau la distribution des multiples flux de trafic sur les différentes liaisons existantes afin de maîtriser la QoS dans le réseau. Il devra donc:

- Maximiser l'utilisation des nœuds et des liens du réseau.
- Provisionner les liaisons en fonction des QoS demandées.
- Répartir le trafic dans le réseau pour éviter les surcharges et minimiser l'impact des pannes.
- Prévoir une capacité réseau en cas de panne ou de surcharge ponctuelle.

L'ingénierie de trafic MPLS permet: [RFC 3212 et 3214]

- Le routage explicite (strict et lâche).
- La spécification de la QoS des LSP créés.
- La spécification des paramètres des trafics entrants.
- L'association des trafics aux différents LSP déjà existants.
- La préemption.
- La création et le maintien de routes de secours.

Le routage avec des contraintes facilite à l'opérateur du réseau de réaliser les opérations de choix des routes. Elles sont prises en charge par les LSR eux-mêmes.

Il est possible de protéger les trafics en définissant pour des LSP primaires des LSP secondaires prêts à l'utilisation. Ces derniers sont crées et maintenus pendant toute la durée de vie des LSP primaires associés. Comme cette méthode augmente la signalisation, il est possible aussi de protéger les LSP par le mécanisme de reroutage rapide. Cette technique

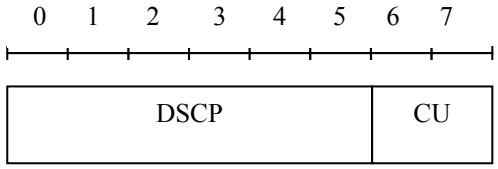
permet au LSP, aussitôt qu'il détecte une rupture au niveau physique d'un de ses liens ou de ses nœuds, de contourner ce lien/LSR défectueux. Cette technique permet une solution plus robuste à long terme en avertissant le LSR Ingress du problème rencontré. Par conséquent, ce dernier pourra redemander la création d'un nouveau LSP répondant aux contraintes demandées.

La préemption avec MLPS définit l'importance relative entre les LSP au niveau d'un même LSR. Deux types de préemption sont définis: la préemption à la création pour créer les LSP d'une façon croissante en fonction de leur priorité, et la préemption de maintien qui forcera un LSP de priorité supérieure à rejeter un LSP de priorité inférieure si cela permet l'établissement du LSP de priorité supérieure. Dans le cas contraire, la préemption n'est pas faisable. Dans chacun des deux types de préemption, 8 classes sont définies. Il est utile de signaler que par défaut, la préemption est inactive.

1.3. La différenciation de service par DiffServ

1.3.1. L'entête DiffServ

L'entête *"DS field"* est définie [34] afin de remplacer le champ TOS (Ipv4) [RFC791] et Traffic Class (Ipv6) [RFC2460].



DSCP : Differentiated Services CodePoint
CU : Currently Unused

Figure 6: Entête DiffServ

Six bits du champ DS sont utilisés comme "*CodePoint*" pour choisir le comportement au prochain nœud (PHB *Per Hop Behavior*). Les bits 6 et 7, actuellement non utilisés, sont ignorés par les nœuds. Afin de faciliter l'évolution future du champ, les bits du DSCP n'ont pas été structurés. La seule contrainte sur les valeurs des DSCP est que si deux connexions admettent deux valeurs de DSCP, et si la valeur de $DSCP_1$ est supérieure à la valeur $DSCP_2$, alors, le service du premier devra être meilleur que le service du second.

Dans [34] et [50], trois classes ont été définies dans le cadre de DiffServ :

- Expedited Forwarding (EF) [8]

Les flux appartenant à cette classe devront être garantie un délai faible, une gigue réduite et une perte faible.

- Assured Forwarding (AF) [13] et [31]

Quatre classes AF sont définies: AF_1, AF_2, AF_3 et AF_4. Chacune de ces classes indépendantes est divisée en trois sous-classes (i.e. AF_1 devient AF_{11}, AF_{12} et AF_{13}).

- Default Behavior (DE) ou Best Effort

Les flux appartenant à cette classe n'ont aucune garantie de délai.

1.3.2. Le traitement à l'entrée du réseau DiffServ

Les nœuds de frontières doivent vérifier la conformité du trafic sortant et entrant dans le réseau DiffServ au TCA (Traffic Conditionning Agreement) entre les deux réseaux.

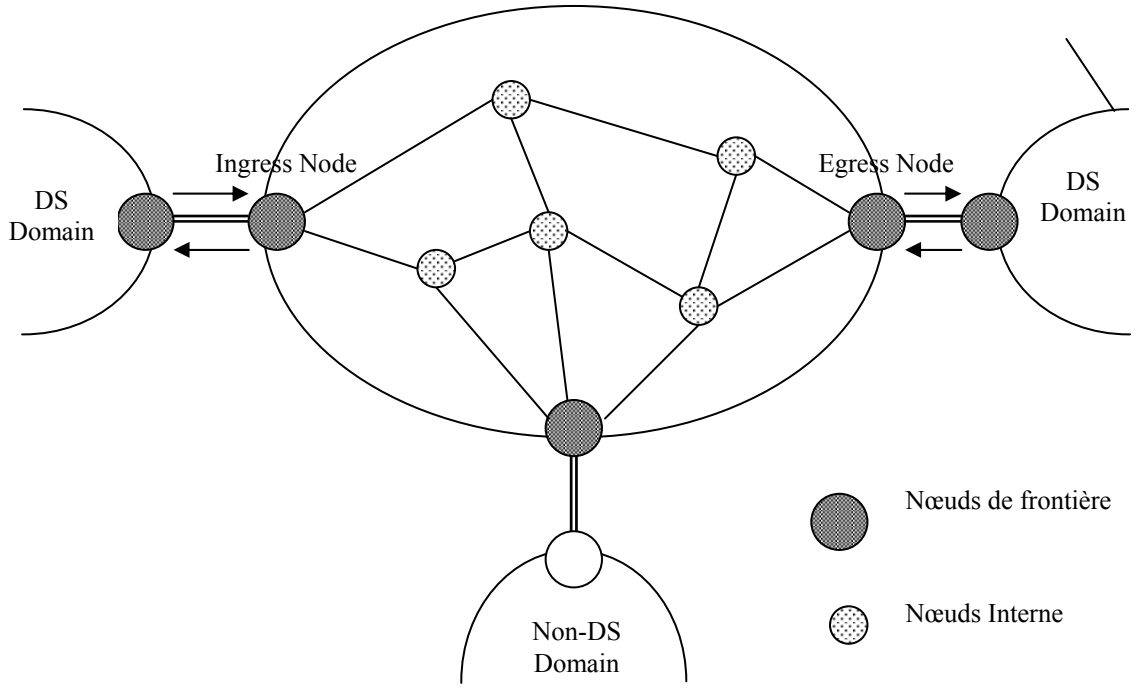


Figure 7: Domaine DiffServ

Deux sortes de nœuds de frontière existent (Figure 7). Les nœuds Ingress qui contrôlent les flux entrants et les nœuds Egress qui vérifient les flux sortants. A priori, dans chaque routeur de frontière DS, le conditionnement illustré sur la Figure 8 est effectué sur chaque flux. Cependant, afin de minimiser les délais d'attente dans les routeurs, il se peut que ce conditionnement soit effectué uniquement dans les routeurs Ingress.

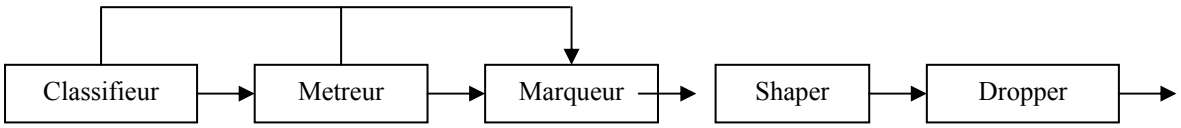


Figure 8: Conditionnement aux routeurs de bordure

Classifieur:

Les données doivent être classées aux limites du domaine DiffServ. Deux types de classifieurs sont définis :

- BA (Behavior Aggragate). La classification des paquets se fait en fonction du champ DSCP uniquement.
- MF (Multi-Field). La classification des paquets se fait en fonction de plusieurs paramètres (adresse source, adresse destination, DSCP, protocole ID, n° de port source, n° de port destination…).

Métreur:

Après le passage dans le classifieur, les données sont présentées au métreur où elles passent par un procédé de mesure qui calcule les caractéristiques du flux. Elles sont comparées au SLA/contrat de trafic entre les deux réseaux. Le métreur contrôle les paramètres du trafic et peut prendre des dispositions si l'un d'eux est dépassé. A noter que le SLA peut contenir plusieurs conditions pour différentes applications, de telle sorte que le métreur puisse examiner des champs dans l'entête IP.

Marqueur:

Lorsque les données seront classifiées et leurs débits caractérisés, le PHB sera marqué dans l'octet DSCP. La raison pour laquelle le PHB n'est pas directement marqué avec la classification, est la possibilité de marquage de paquets de la même classe différemment en fonction du débit entrant au nœud. Pour un trafic, le marqueur peut décider de marquer un paquet en classe 1 si la longueur de la file est inférieure à un seuil s_1; en classe 2 si la longueur de la file est entre deux seuils s_1 et s_2; en classe 3 si la longueur de la file est supérieure au seuil s_2. Ce marquage peut se faire à l'entrée du réseau fournisseur, comme à la sortie d'un réseau client.

Shaper et Dropper:

Si le trafic correspond au profil de sa classe, les paquets seront acheminés vers le réseau. Par contre, s'ils excèdent leur contrat, les paquets supplémentaires seront soit retardés dans le shaper de sorte que le trafic de sortie soit celui requis, soit rejetés par le dropper. La sélection semble catégorique mais il est parfois nécessaire, surtout quand la mémoire de la file est pleine ou quand le délai des paquets est faible, d'éliminer les paquets excédentaires. Ce dernier cas est plausible pour les paquets de voix qui n'acceptent pas de grands délais.

Un routeur intermédiaire peut modifier le DSCP si :

- Les DSCP portent des informations sur le réseau.
- Des conditions existent pour déclasser ou surclasser un paquet pour des raisons bien définies.
- Une limite entre deux zones n'est pas couverte par un SLA. Dans ce cas, la politique est décrite par le routeur du domaine en amont.

De plus, les nœuds internes peuvent effectuer un re-marquage, voire même toutes les fonctionnalités des nœuds de frontière. Ainsi, le destinataire d'un paquet DiffServ ne doit pas supposer que la valeur du champ DSCP du paquet qu'il a reçu correspond à la même valeur émise par la source.

1.4. La combinaison DiffServ et MPLS

Avant de relater les possibilités de création des LSP en prenant compte du champ DiffServ, nous formulons quelques définitions relatives au traitement des flux [23]:

- FEC: Dans un domaine MPLS, c'est un groupe de paquets IP qui sont commutés de la même manière. (e.g. sur le même chemin, avec le même traitement).
- Behaviour Aggregates (BA): c'est l'ensemble des paquets IP passant dans un domaine DiffServ et demandant le même traitement. En d'autres termes, un BA est un ensemble de paquets qui possède le même champ DSCP. Chaque nœud DiffServ détermine, à partir de la valeur du DSCP dans l'entête du paquet, le PHB (Per Hop Behavior) à appliquer (i.e. l'ordonnancement et la probabilité de perte).

- Ordered Agregate (OA): c'est un ensemble de Behaviour Aggregates liés par un certain partage d'ordre des contraintes.
- PHB Scheduling class (PSC): c'est l'ensemble des PHB appliqués aux différentes Behaviour Aggregates du même Ordered Agregate.

Par exemple, AF_{1x} est un PSC comprenant les PHB de AF_{11}, AF_{12} et AF_{13}. De même, EF est un PSC comprenant exclusivement le PHB de EF.

A present, nous pouvons présenter les deux méthodes de création de LSP prenant en compte les informations relatives à DiffServ:

- E-LSP (EXP-Inferred-PSC LSP): dans ce type de LSP, le champ EXP contient les informations d'ordonnancement et de rejet (dropping). Ainsi, chaque E-LSP peut contenir jusqu'à 8 PSC.
- L-LSP (Label-Only-Inferred-PSC LSP): dans ce type de LSP, les informations d'ordonnancement sont transportées implicitement dans le Label. Le champ EXP contient les informations de perte uniquement. Dans chaque LSP de cette famille, un seul PSC –donc un seul OA– est acheminé par LSP.

Les E-LSP et L-LSP, qui sont établis soit par LDP soit par RSVP, peuvent coexister dans le même réseau DiffServ/MPLS. La réservation des ressources pour un L-LSP est effectuée pour tout le chemin. Donc, le contrôle d'admission est déterminé par les ressources de toute la PSC. Dans le cas d'un E-LSP, puisque la réservation est aussi faite pour tout le LSP, celle-ci correspond alors à l'ensemble des PSC transportés.

Dans le but de favoriser le facteur d'échelle et d'obtenir une meilleure gestion de la bande passante, la notion de "Type de Classe" est définie. Il s'agit d'un ensemble de classes ayant une bande passante agrégée commune minimale et maximale qui garantit le niveau de performances requis. Il n'existe pas de bande passante maximale ou minimale qui soit garantie à une seule classe dans un *type de classe*.

Dans les E-LSP, la correspondance entre la valeur du champ EXP et le DSCP peut se faire par une des trois méthodes suivantes:

- Préconfiguration ou management.
- Signalisation.
- Dans le cas où aucune des deux méthodes précédentes n'est faite, la valeur du EXP devra correspondre au PHB par défaut, qui est celui du mode Best Effort.

Dans les différents paragraphes qui précèdent, nous avons présenté un protocole (MPLS) et un mécanisme (DiffServ) de mise en place de QoS dans les réseaux IP. Néanmoins, ces deux méthodes utilisent les algorithmes d'ordonnancement pour implémenter la QoS et la différenciation des classes sur les nœuds. En conséquence, dans le prochain paragraphe nous décrirons plusieurs algorithmes d'ordonnancement. Nous insisterons en particuliers sur les algorithmes que nous utiliserons dans la suite de la thèse, en l'occurrence dans le Chapitre 3.

1.5. Algorithmes d'ordonnancement

La différenciation des classes mise en place par DiffServ se fait principalement par l'ordonnancement. La question qui vient se poser est la suivante: en présence de plusieurs files, laquelle le routeur devra-t-il servir en premier? Dans ce paragraphe, nous présentons les principaux algorithmes d'ordonnancement que les constructeurs de routeurs et de

commutateurs de paquets ont implémenté. Excepté l'ordonnanceur FIFO qui ne sera utilisé ici que pour des aspects de comparaison, nous avons décomposé les algorithmes en 4 familles:

- L'ordonnanceur à priorité absolue
- Les ordonnanceurs dérivés de GPS (Generalised Processor Sharing)
- Les ordonnanceurs dérivés du RR (Round Robin)
- Les ordonnanceurs à temps

Bien que nous ayons décomposé les algorithmes en 4 familles, toute combinaison entre ces familles est permise. D'ailleurs, à la fin du Chapitre 3, nous adoptons un ordonnanceur "mixte" pour différencier les flux.

Mais, tout d'abord, nous définissons quelques objectifs des ordonnanceurs:

- **Isolation des flux:** quelque soit le débit des flux sources, la portion de service allouée au différents flux doit être la plus proche de celle requise. En présence de flux à débit élevé, les flux à faible débit ne doivent pas être pénalisés. Cet aspect est important surtout dans les réseaux non protégés par un contrôle d'admission.
- **Délai:** le retard introduit par le traitement de l'algorithme doit être acceptable pour la QoS du trafic.
- **L'utilisation**: la bande passante du réseau doit être utilisée efficacement.
- **L'équité:** la bande passante doit être partagée équitablement entre les différents usagers actifs de la même classe.
- **Simplicité:** l'algorithme d'ordonnancement doit être facile à implémenter et simple à gérer.
- **Facteur d'échelle:** l'algorithme devra gérer un grand nombre de connexion et devra supporter une diversité de la vitesse des liens.

1.5.1. First In First Out (FIFO)

C'est le plus simple algorithme. L'ordre de service des paquets est l'ordre chronologique de leur arrivée dans la file. L'avantage d'un tel algorithme est sa simplicité absolue (pas de puissance du processeur utilisé ni de mémoire supplémentaire). Bien sur, ce schéma ne permet de différencier aucune classe.

Dans le réseau IP standard, sans mécanisme de QoS, tous les flux sont traités identiquement au niveau du cœur de réseau, c'est-à-dire sans aucune différenciation entre les divers protocoles de transport ni entre les applications utilisées pour générer les paquets. De ce fait, tous les flux subiront le même délai de traversée du réseau. Et dans le cas d'une congestion, tous les paquets auront la même probabilité de perte.

1.5.2. Head Of the Line (HOL)

C'est un algorithme assez simple: à l'entrée de l'ordonnanceur, les paquets sont classifiés et mis dans des files. La priorité de chaque paquet correspondant à la priorité de sa file. Les files sont servies de la file la plus prioritaire à la file la moins prioritaire.

Cet algorithme assez simple ne garantit la performance que pour les paquets de plus haute priorité. Nous savons aussi d'après la théorie des files d'attentes que le temps d'attente moyen de la file la plus prioritaire dépend aussi du taux d'arrivée et du temps de service moyen de toutes les files. De plus, dans un grand réseau, les priorités changent d'un serveur à un autre, ce qui complique le calcul du temps de traversée.

1.5.3. Generalised Processor Sharing (GPS) et algorithmes à poids dérivés

1.5.3.1. GPS

Cet algorithme [6] de base, garantit une borne au délai de traversée du réseau. Pour chaque connexion *i*, un poids ϕ_i est associé. En conséquence, la capacité du lien sera partagée entre les différentes connexions en fonction du poids associé à chacune d'elles. Si *C* désigne la capacité du lien, alors le débit minimum garanti pour la connexion *i* en présence de *N-1* autre classes de poids $\phi_j, j \in \{1..N-1\}$ sera :

$$m_{d_i} = \frac{\phi_i}{\sum_{j=1}^{N} \phi_j} C$$

Lorsqu'une ou plusieurs connexions n'émettent pas de paquets, leur(s) part(s) seront reparties sur les autres connexions actives en fonction du poids respectif de chacunes d'elles.

Donc nous pouvons écrire :

$$D_i = \begin{cases} \dfrac{\phi_i}{\sum_{j \in B(t)} \phi_j} C, & i \in B(t) \\ 0 & autrement \end{cases}$$

où : $D_i(t)$: débit alloué à la connexion *i* à l'instant *t*.
C : débit total du lien.
B(t) : l'ensemble des connexions actives à l'instant t.

En notant $W_i(t_1,t_2)$ la quantité de données servie de la connexion *i* entre les instants t_1 et t_2, nous pouvons écrire:

$$\forall i, j \in [1,...,N] \quad \frac{W_i(t_1,t_2)}{\phi_i} = \frac{W_j(t_1,t_2)}{\phi_j}$$

Ce qui veut dire qu'un serveur GPS sert toutes les connexions simultanément et en fonction de leurs parts respectives. En se basant sur l'indice d'équité $\left| \frac{W_i(t_1,t_2)}{\phi_i} - \frac{W_j(t_1,t_2)}{\phi_j} \right|$, nous constatons que cet indice est nul pour l'algorithme GPS. Il s'agit du cas idéal où toutes les connexions sont servies en fonction de leur poids.

L'inconvénient de GPS est que cet un ordonnanceur est irréalisable: un système ne peut pas servir plusieurs paquets en même temps. Cet algorithme suppose que l'arrivée d'un paquet correspond à l'arrivée d'une certaine quantité de fluide dans le système. Un paquet est alors servi quand le volume correspondant à ce paquet est libéré du système. Donc cet algorithme n'est valable que comme une référence pour les comparaisons.

Les temps de départ (les instants de fin de service) dans un système GPS sont calculés en utilisant la notion de temps virtuel. Le temps virtuel n'existe que dans les périodes d'activités d'une classe. Il est initialisé à zéro au début de ces périodes et pendant les périodes d'activités, il évolue suivant $\frac{\delta V}{\delta t} = \frac{C}{\sum_{i \in B(t)} \phi_i}$, avec *C* le débit du lien, *V* le temps virtuel, *t* le temps et *B(t)* correspond à l'ensemble des connexions actives à l'instant *t*.

Soit a_i^k l'instant d'arrivée de la $k^{ème}$ donnée de la connexion *i* et l_i^k sa longueur. Elle entame son service à l'instant virtuel s_i^k et finit à l'instant f_i^k qui peut être déterminé en se basant sur les équations ci-dessous:

1. $f_i^0 = 0$

Pour la notation, nous fixerons l'instant de fin de service du paquet 0 à zéro.

2. $\forall k \geq 1; s_i^k = \max\left\{V(a_i^k), f_i^{k-1}\right\}$

Le paquet commence son service soit dès son arrivée, soit dès que le paquet qui le précède finit son service.

3. $\forall k \geq 1 \; f_i^k = s_i^k + \frac{l_i^k}{D_i}$

Le paquet achève son service après le temps $\frac{l_i^k}{D_i}$ de début de service. Il correspond au temps de traitement par le serveur qui lui alloue une bande passante de D_i.

Quelques unes des propriétés de l'algorithme GPS sont les suivantes:

- Il garantit un délai borné à une source contrôlée par un seau percé.
- La bande passante est distribuée équitablement entre toutes les sources actives.
- Les sources peuvent estimer leurs bandes passantes disponibles à chaque instant d'une façon précise.

Cependant, l'algorithme GPS considère que le serveur peut traiter plusieurs files simultanément et que le trafic est divisible indéfiniment. Dans un scénario plus réaliste, une seule connexion peut être servie à la fois et un paquet tout entier doit être traité avant de passer à un autre. De plus, comme le calcul du temps virtuel n'est pas simple, plusieurs algorithmes sont proposés pour résoudre ce problème.

1.5.3.2. Weighted Fair Queuing (WFQ)

Un des algorithmes dérivé de GPS est le PGPS (Packet per packet Generalised Processor Sharing) connu aussi sous le nom de WFQ (Weighted Fair Queuing) [6].

A l'instant τ de l'émission d'un paquet, WFQ choisit le premier paquet, parmi ceux existants à cet instant, qui aurait dû terminer son service si l'ordonnanceur était GPS.

L'ordonnanceur WFQ garantit alors que l'avance de la transmission d'un paquet dans la file WFQ et sa transmission en GPS est toujours inférieure au temps de transmission d'un paquet de taille maximale. De même, en terme de quantité totale de bit servie par chaque session *i*, la file GPS peut au maximum envoyer la taille maximale d'un paquet plus que la file WFQ correspondante.

Le débit garanti pour la connexion *i* s'exprime par $\frac{\phi_i}{\sum_j \phi_j} \times C$.

La complexité d'un serveur WFQ est de *0(N)*, où *N* est le nombre de flux différencié.

1.5.3.3. Worst-case Fair Weighted Fair Queuing (WF^2Q)

Comme son nom l'indique, cet algorithme [33] est équitable même avec le cas le plus pénalisant. A l'instant τ, quand le serveur WF^2Q entreprend le processus de choix du paquet émis, il ne considère que les paquets qui ont déjà entamé leur service -et peut être fini- dans le

système GPS correspondant. Parmi cet ensemble, WF^2Q choisit le paquet qui devra terminer son service le premier.

Dans l'exemple ci-dessous, nous comparons GPS, WFQ et WF^2Q. Supposons une file avec 5 connexions dont l'arrivée des paquets suit la Figure 9 ci-dessous :

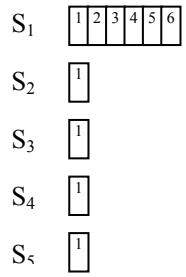


Figure 9: Génération des différentes connexions.

Pour simplifier, tous les paquets ont la même taille *L*. La connexion S_1 est attribuée à un poids ϕ_1 et chacune des connexions S_2 à S_5 à un poids de $\phi_1/4$. En prenant *C* comme le débit total du lien, le débit de service de la file 1 devient C/2 et celui des files 2 à 5 de C/8. La Figure 10 représente le service de ces paquets suivant le service GPS.

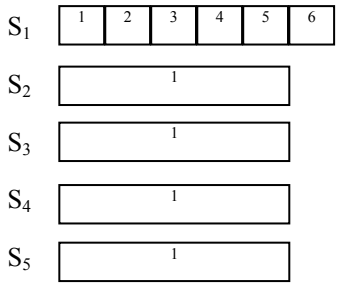


Figure 10: Emission suivant l'algorithme GPS.

Le déroulement suivant l'ordonnanceur WFQ sera comme suit. A l'instant 0, le premier paquet parmi ceux qui existent et qui finit suivant l'ordonnanceur GPS est le paquet 1 de la connexion 1. Il sera donc émis en premier. Le second paquet qui termine son service dans GPS est le paquet 2 de la connexion 1. Ce paquet sera donc traité en second lieu dans la file WFQ. La même logique suivra jusqu'à ce que le paquet 5 soit émis. Juste après, ce sont les paquets des connexions 2, 3, 4 et 5 qui seront servis chacun à la suite de l'autre. Enfin, c'est au tour du paquet 6 de la connexion 1. Le service suivant WFQ est illustré sur la Figure 11 :

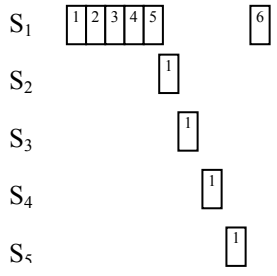


Figure 11: Emission suivant l'algorithme WFQ.

Avec ce scénario, WFQ pénalise énormément le paquet 6 de la connexion S_1. La gigue, la différence de délai, est alors élevée.

Le déroulement suivant WF^{2}Q est différent. Au premier instant, c'est le paquet 1 de la connexion 1 qui est servi. Juste après, à l'instant *L/C*, l'ordonnanceur WF^{2}Q crée l'ensemble des paquets qui ont commencé leur service, soient les paquets 1 des connexions 2, 3, 4 et 5. Le paquet 2 de la connexion 1 n'est pas inclus vu qu'il n'a pas encore entamé son service dans la file GPS. Il commencera son service à l'instant $L/C.\phi_1$. Après l'émission du paquet 1 de la connexion 2, qui correspond à l'instant *2.L/C*, l'ensemble des paquets qui ont commencés leurs services devient, en plus des paquets 1 des connexions 3, 4 et 5, le paquet 2 de la connexion 1. Ce dernier entame son service avec GPS à l'instant *2.L/C*. La Figure 12 donne l'ordre de service des paquets avec l'ordonnanceur WF²Q.

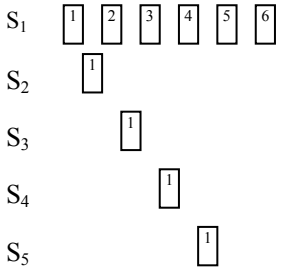


Figure 12 : Emission suivant l'algorithme WF²Q.

Avec l'ordonnanceur WF²Q, le délai total pour la transmission des paquets de la connexion S_1 reste le même, mais la gigue est réduite considérablement. De plus, les connexions S_2 à S_5 sont servies beaucoup plus rapidement : les délais sont répartis sur toutes les connexions.

Soit $d_{i,s}^k$ l'instant de départ du $k^{ème}$ paquet de la file *i* avec l'ordonnanceur *s*; $W_{i,s}(0,\tau)$ la quantité de données servies pour la classe *i* entre les instants zéro et τ par l'ordonnanceur *s*; L_{max} la taille maximale des paquets et $L_{i,max}$ la taille maximale des paquets de la connexion *i*, les propriétés suivantes sont vérifiées par le WF^{2}Q :

1. $d_{i,WF^2Q}^k - d_{i,GPS}^k \le \frac{L_{\max}}{C}$

   Comme WFQ, un paquet dans un ordonnanceur WF^{2}Q n'attend pas plus que le temps de transmission d'un paquet de longueur maximale que dans la file GPS correspondante.

2. $W_{i,GPS}(0,\tau) - W_{i,WF^2Q}(0,\tau) \le L_{\max}$

   La quantité de donnée servie avec GPS pour la classe *i* ne peut être plus grande que celle servie par WF^{2}Q que par la longueur maximale d'un paquet.

3. $W_{i,WF^2Q}(0,\tau) - W_{i,GPS}(0,\tau) \le \left(1 - \frac{\phi_i}{\sum_{j=1}^{N}\phi_j}\right) L_{i,\max}$

   La quantité de bits servis par WF²Q pour la classe *i*, entre les instants zéro et τ ne peut être plus grande que la quantité servie par le GPS correspondant, que par une fraction de la taille maximale d'un paquet de la classe *i* en question.

A partir de ces différentes observations, nous pouvons dire que WF^2Q est une meilleure approximation au GPS que WFQ.

La complexité de WF^2Q est de *0(N)*, avec *N* le nombre des flux à gérer.

Les auteurs de [32] proposent un nouvel algorithme, nommé WF^2Q+, qui a l'avantage de réduire la complexité de WF^2Q tout en gardant ses propriétés de limitation du temps d'attente et de quantité de données transmises. Ainsi, la complexité de WF^2Q+ devient de *0(log N)*, *N* étant le nombre de connexions.

Plusieurs autres approximations de GPS que nous n'avons pas présentées ici existent encore. Citons notamment SCFQ (self-clocked fair queueing) [54] et Virtual Clock [35].

1.5.4. Round Robin (RR) et ses dérivées

Le Round Robin consiste à scruter les files les unes à la suite des autres. Si la file contient un paquet, ce dernier sera servi. Par contre, si aucun paquet ne se trouve à ce moment, l'ordonnanceur passe directement à la file suivante.

Plusieurs méthodes de services existent:

- Le service reste dans une file tant qu'il y a des paquets à servir.
- Un nombre limité de paquets peut être servi pour chaque queue.

Cet algorithme est simple. Il est équitable si les paquets des différentes files sont de mêmes longueurs. Dans le cas contraire, les files avec les paquets les plus longs seront les mieux servies.

1.5.4.1. Weighted Round Robin (WRR)

L'ordonnanceur parcourt les différentes files et les sert en fonction du poids associé à chacune d'elles. Cet ordonnanceur reste simple. L'équité est vérifiée si les paquets des différentes files ont la même longueur. Dans le cas contraire, ce sont les files avec les paquets de plus grandes tailles qui seront favorisées.

1.5.4.2. Deficit Round Robin (DRR)

Les concepteurs de DRR [37] voulaient résoudre le problème de l'équité de WRR. Ils ont alors introduit un compteur de déficit lié à chaque file. A chaque tour, ce compteur est incrémenté d'un quantum q_i pour la classe *i*. Si la valeur du compteur devient plus grande que la taille du premier paquet dans la file, alors ce paquet sera envoyé au réseau et le compteur de la file sera réduit de la valeur de la taille du paquet. Cette valeur sera sauvegardée pour le prochain tour. Dans le cas où la somme du déficit et du quantum est inférieure à la taille du paquet, le déficit est incrémenté du quantum et sauvegardé, mais le paquet ne sera pas émis. L'ordonnanceur passera ensuite à la prochaine file. Quand l'ordonnanceur se présente devant une file vide, le déficit de cette dernière est initialisé à zéro afin de garder l'équité.

Cet ordonnanceur devient donc indépendant de la taille des paquets de chaque classe. Par le choix du quantum de chaque file, la bande passante allouée à chaque classe sera fixée.

1.5.5. Les ordonnanceurs à temps

Les ordonnanceurs de cette famille servent les paquets en fonction de leur ordre temporelle d'émission. Chaque paquet admet un certain temps limite d'envoi en fonction de sa classe.

1.5.5.1. Earliest Deadline First (EDF)

Cet algorithme associe à chaque paquet un délai maximum d'attente i.e. deadline. A l'entrée de l'ordonnanceur, ce dernier fixe pour chaque paquet un temps au bout duquel il doit être servi. Ainsi, les paquets sont triés et donc émis en fonction de la valeur croissante du délai permis.

Un problème existe avec cette technique: en cas de surcharge du nœud, les performances en terme de garantie de délai d'attente ne seront plus respectés (voir Tableau 1 et Figure 13).

Tableau 1: Exemple d'algorithme EDF.

	Deadline	Temps de traitement	Instant d'émission
T_0=0	6	3	3
T_1=1	7	3	6
T_2=2	8	3	9
T_3=3	9	3	12
T_4=4	10	3	15
T_5=7	13	3	18
T_6=10	16	3	21
T_7=13	19	3	24

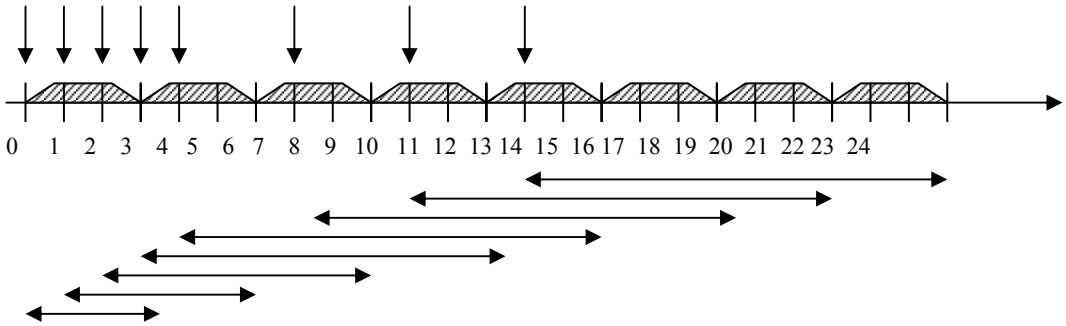


Figure 13: Scénario de transmission des paquets sous EDF.

Pour résoudre ce problème, il faudra un système préemptif qui décide d'arrêter l'émission d'un paquet dans les instants de surcharge.

Plusieurs articles dont [25], ont proposé des méthodes pour améliorer les performances de l'EDF pendant les surcharges. Le principe est soit la préemption: arrêter le traitement d'un paquet, traiter un autre, puis revenir au premier et continuer son traitement, soit l'élimination des paquets dont le délai maximal est dépassé.

1.5.5.2. Earliest Due Date (EDD)

Delay Earliest Due Date (Delay EDD)[12]: ce mécanisme fixe sur chaque nœud le seuil du délai de tel sorte que si toutes les connexions sont à un débit maximal, le délai subit reste inférieur à la limite imposée par le flux. Avant d'accepter un flux, ce mécanisme effectue un contrôle d'admission afin de s'assurer que les performances des flux déjà présents dans le réseau ne seront pas détériorées.

Jitter Earliest Due Date (Jitter EDD) [14]: ce mécanisme fixe le seuil du délai sur chaque ordonnanceur du réseau de façon à ce que même si toutes les connexions sont établies avec un débit maximal, la gigue de chacun des trafics déjà acceptée est vérifiée.

1.5.6. Comparaisons d'ordonnanceurs à poids

Nous comparons ici les trois algorithmes WRR, WF^2Q et SCFQ. Les deux derniers sont deux algorithmes d'approximation de WFQ. Nous présentons dans les figures des tableaux ci-dessous (du Tableau 2 au Tableau 10) les probabilités des délais, des taux de pertes et les valeurs moyennes des variances du délai pour une file unique implémentant ces ordonnanceurs. Nous avons obtenu ces résultats par simulation NS. Les flux des trois classes de trafics existants dans la file sont émis par trois sources différentes. Le temps d'inter arrivée des paquets suit la loi de Poisson et la distribution de la taille des paquets est exponentielle de moyenne 500 octets. Le débit du lien qui lie les sources à la destination est de 2 Moctests/sec et admet un temps de propagation de 10ms. Nous avons pris trois charges possibles de la file: la première charge est de 0,2, la deuxième de 0,5 et la troisième de 0,8.

Nous constatons à partir de ces différentes courbes que WRR donne une meilleure QoS que WF^2Q ou SCFQ pour les différentes charges possibles et surtout pour la classe 1. Le taux perte de WRR est nettement meilleur. Le délai moyen ainsi que sa variance de WF^2Q et SCFQ présente, à des points, de meilleure valeurs que WRR mais, globalement, WRR reste mieux. Les deux algorithmes WF^2Q et SCFQ ne pénalisent pas beaucoup la classe 3. Néanmoins, nous cherchons avant tout d'avantager la classe 1.

Suite à cette comparaison, et dans la suite de cette thèse, nous utiliserons majoritairement le modèle WRR dans nos différents réseaux simulés. Cette conclusion nous permet aussi, dans notre étude relative à GPS, d'approximer le modèle GPS théorique par l'ordonnanceur WRR.

Tableau 2: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WRR avec une pondération 6/3/1.

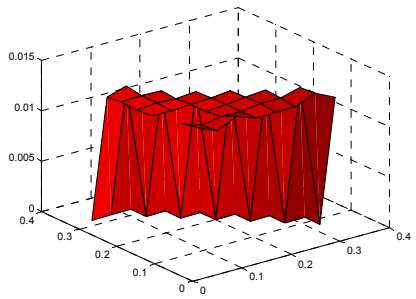
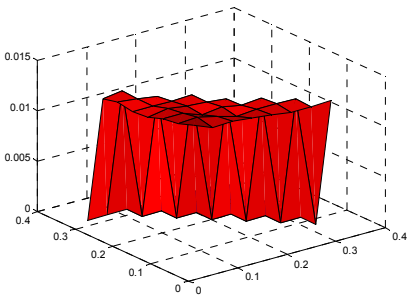
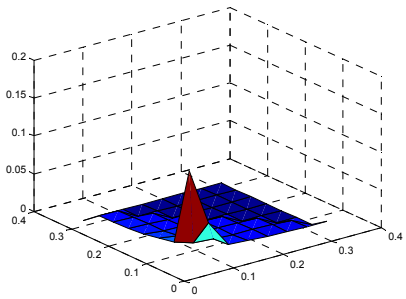
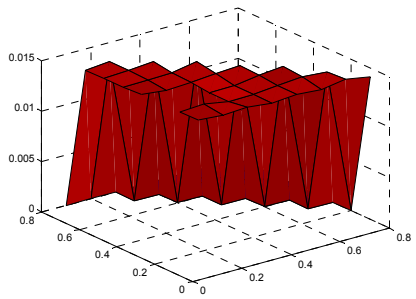
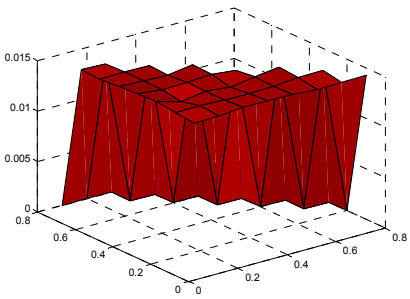
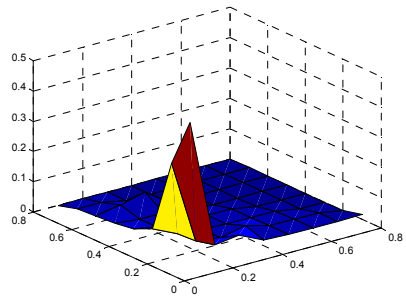
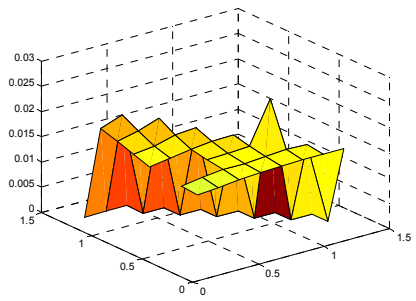
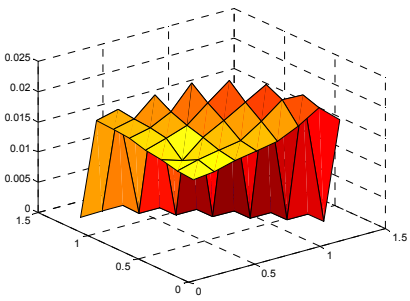
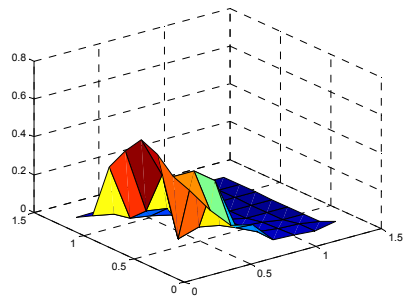
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 3: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WF^2Q avec une pondération 6/3/1.

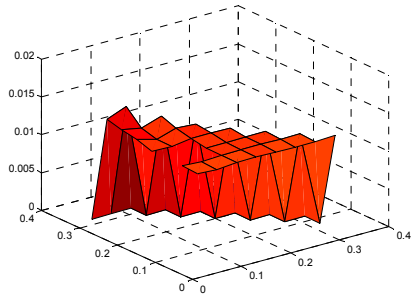
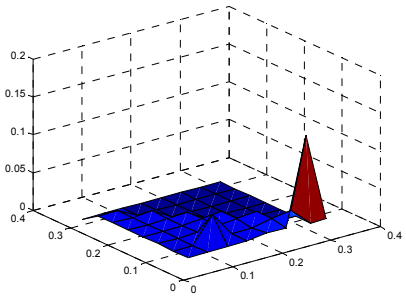
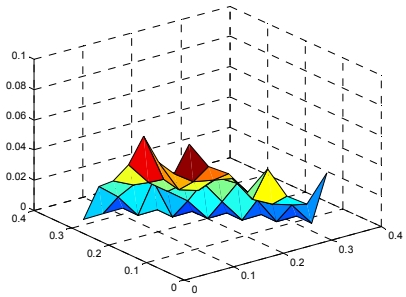
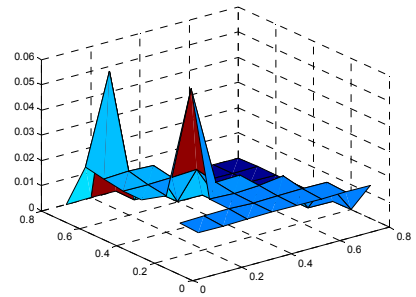
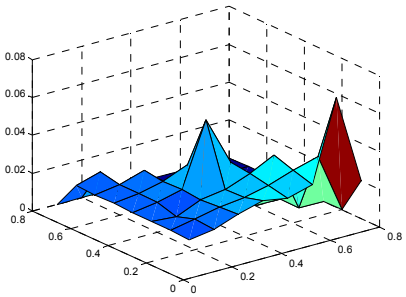
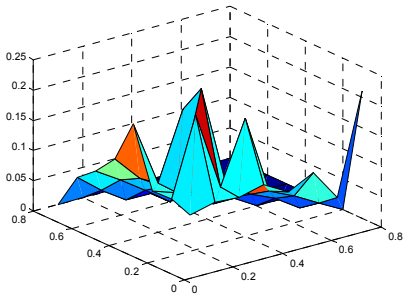
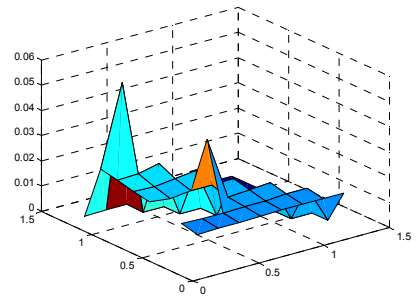
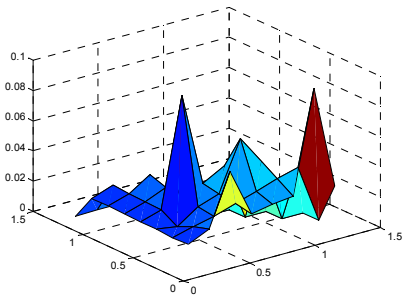
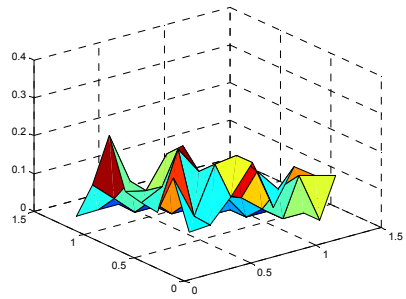
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 4 : La variation du délai moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur SCFQ avec une pondération 6/3/1.

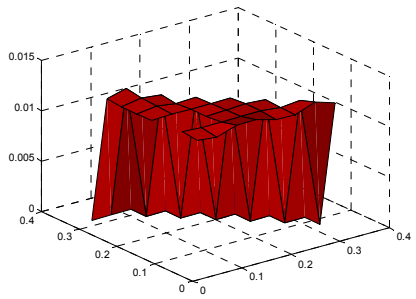
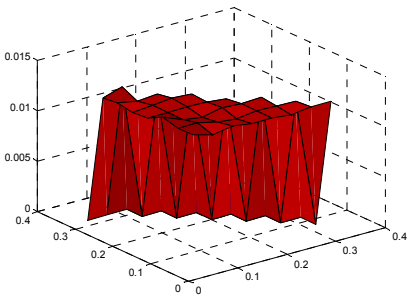
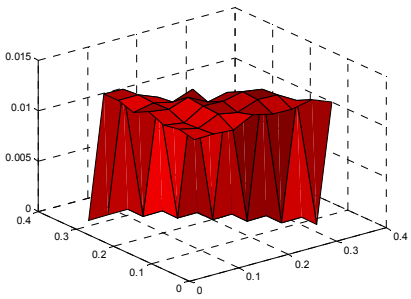
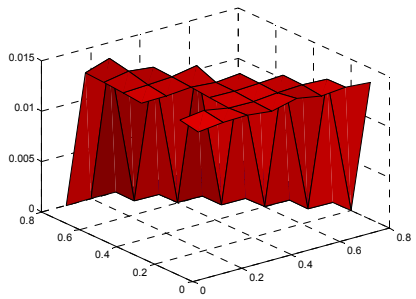
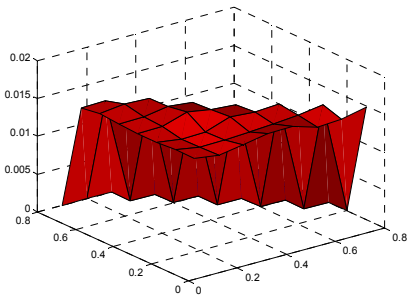
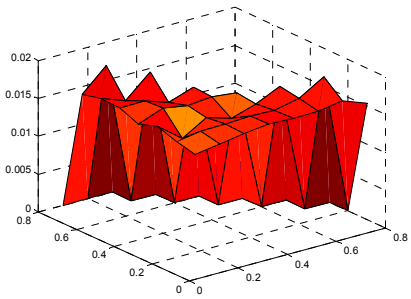
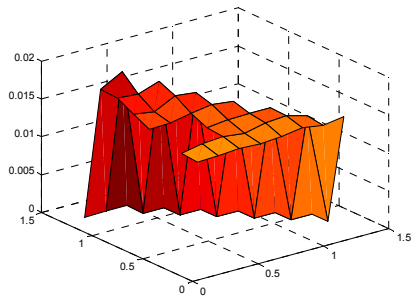
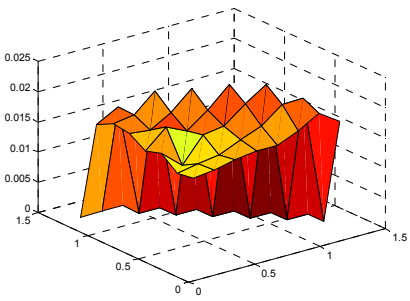
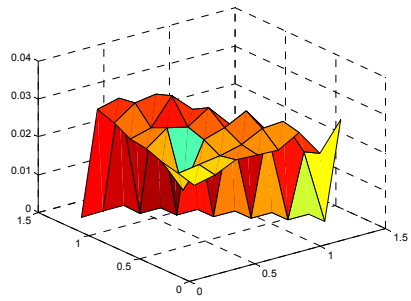
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 5: La variation du taux de perte moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WRR avec une pondération 6/3/1.

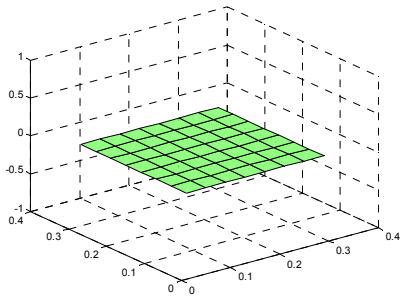
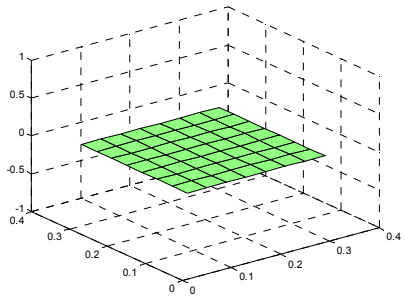
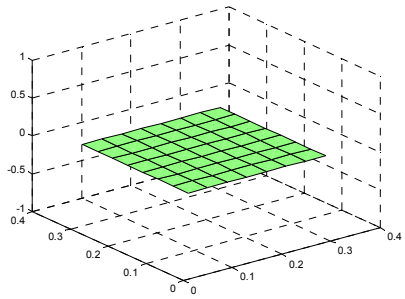
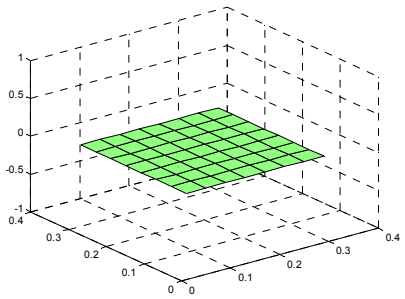
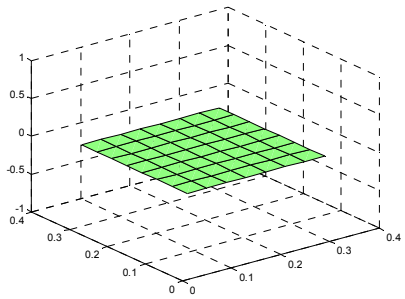
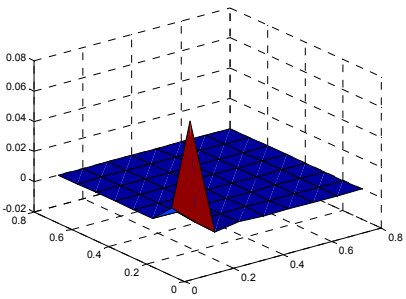
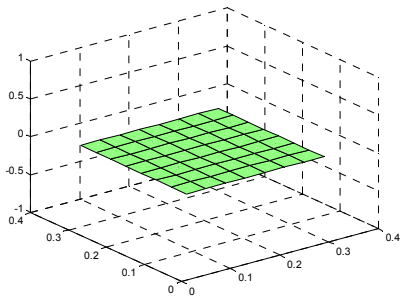
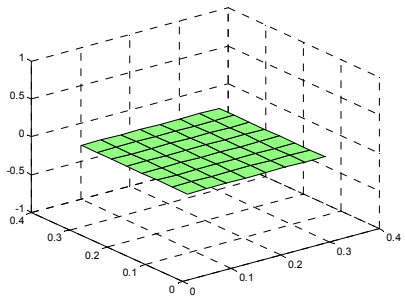
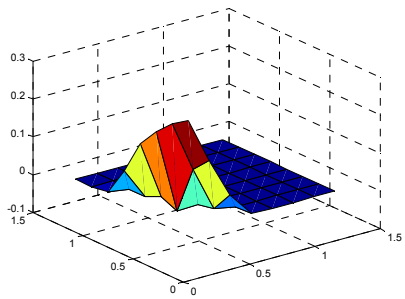
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 6: La variation du taux de perte moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WF^{2}Q avec une pondération 6/3/1.

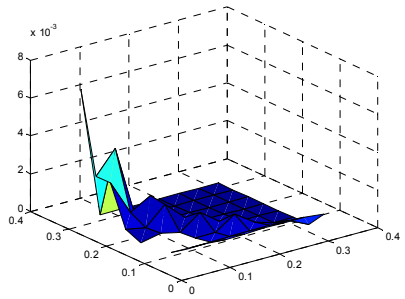
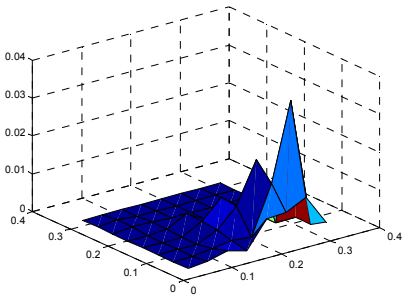
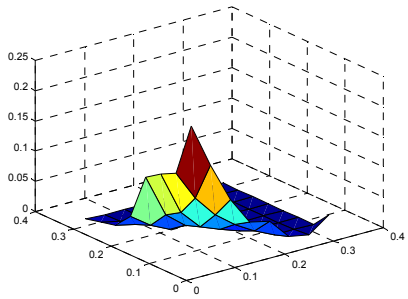
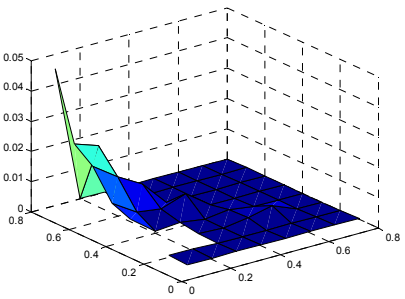
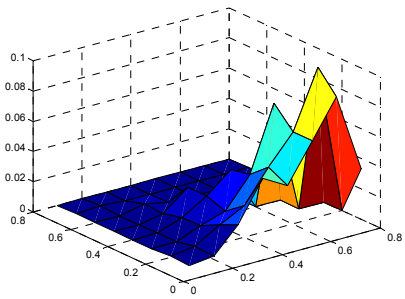
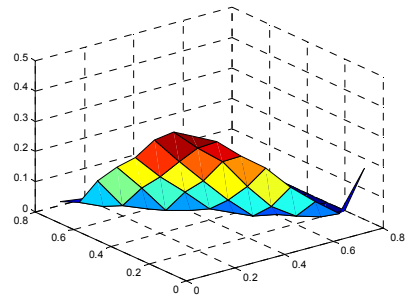
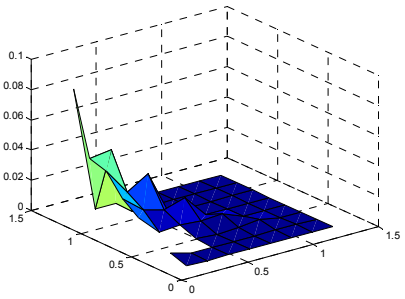
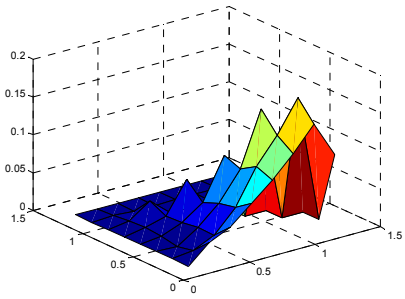
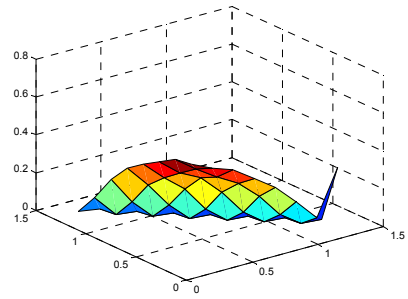
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 7 : La variation du taux de perte moyen en fonction de la charge des classes 1 et 2 pour l'ordonnanceur SCFQ avec une pondération 6/3/1.

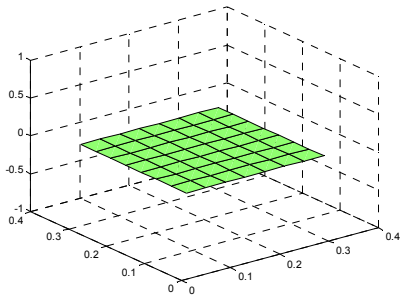
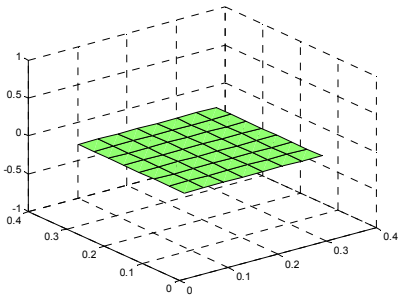
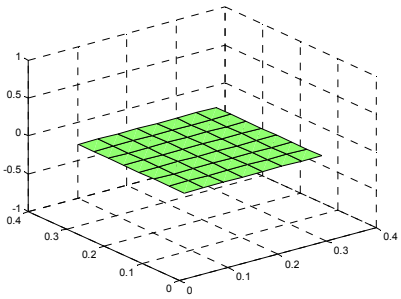
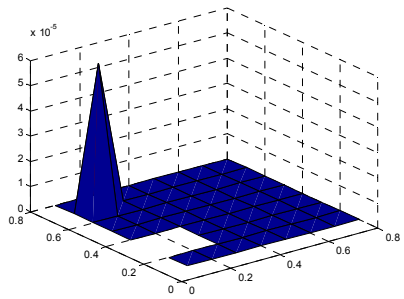
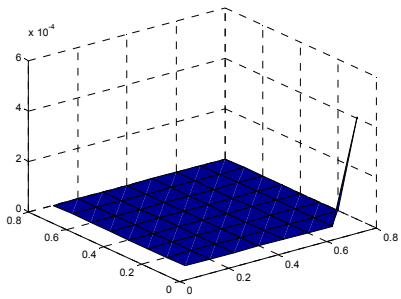
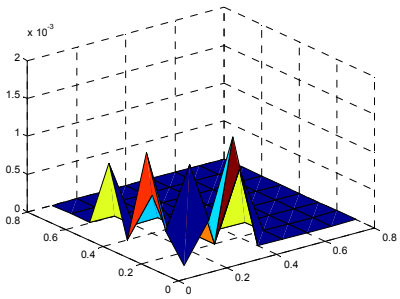
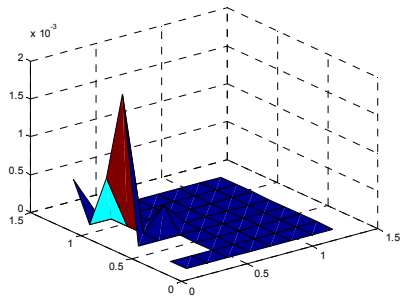
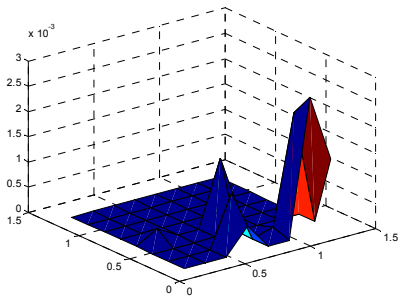
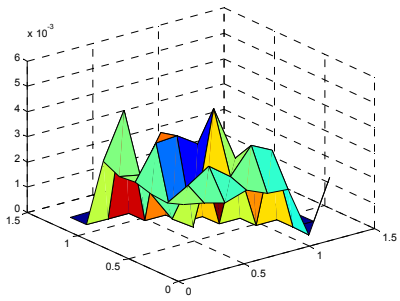
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 8: La variation de la variance du délai en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WRR avec une pondération 6/3/1.

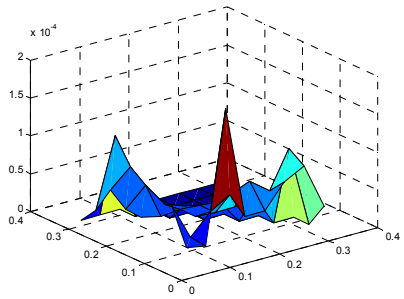
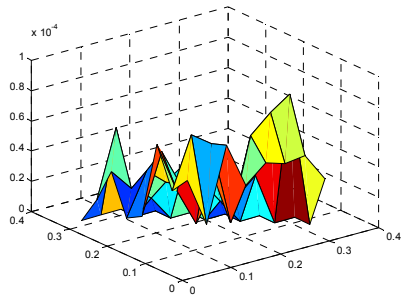
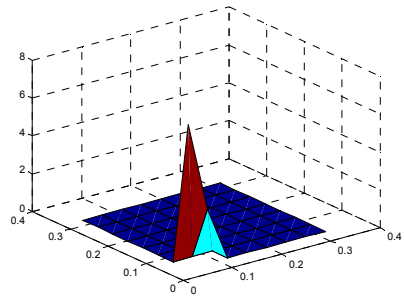
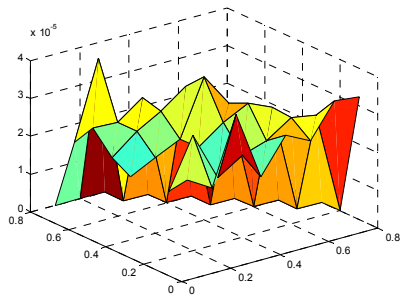
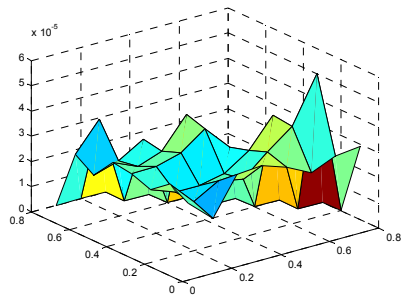
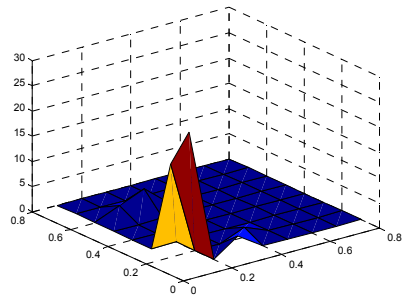
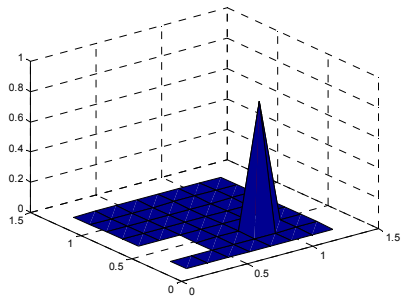
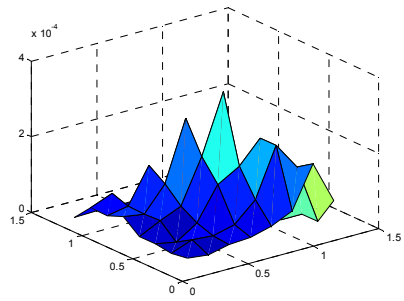
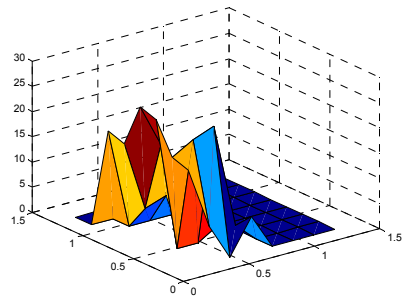
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 9: La variation de la variance du délai en fonction de la charge des classes 1 et 2 pour l'ordonnanceur WF^{2}Q avec une pondération 6/3/1.

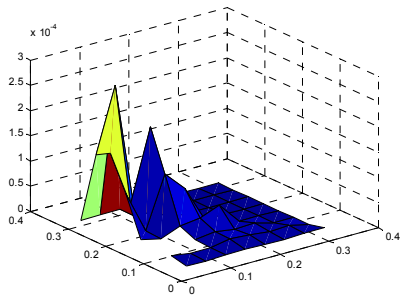
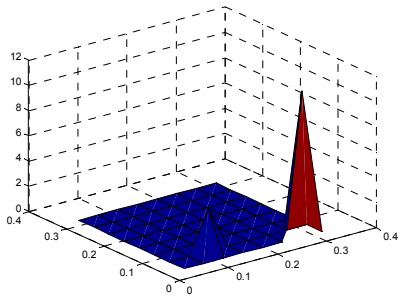
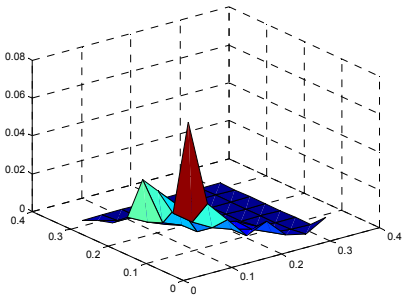
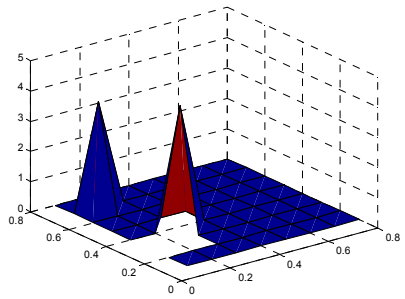
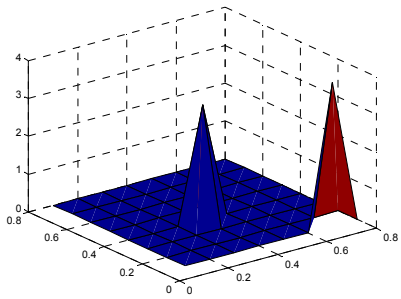
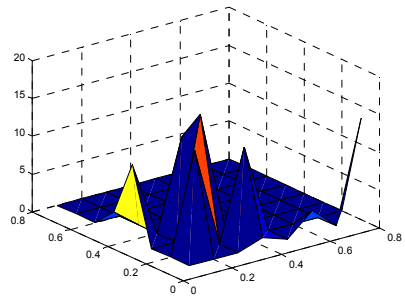
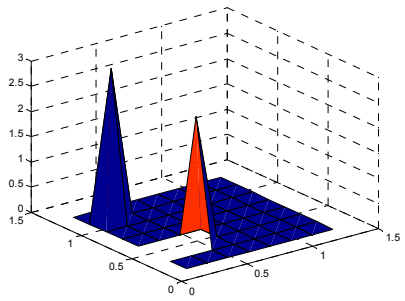
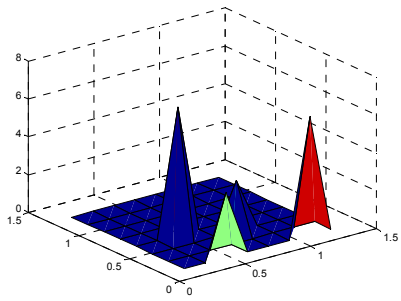
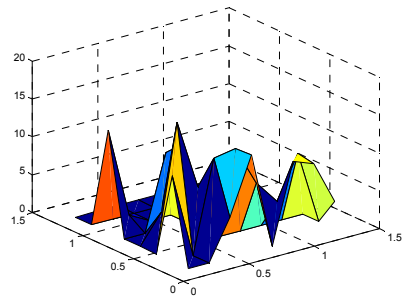
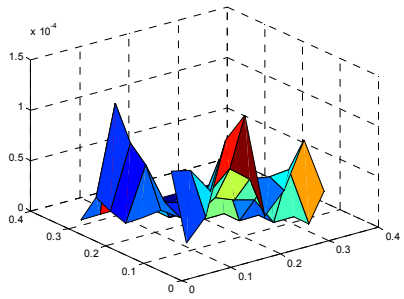
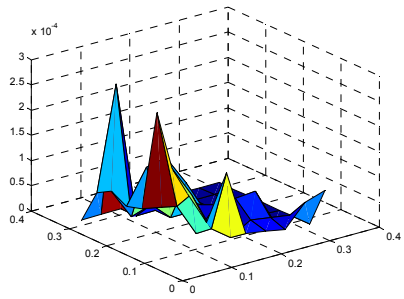
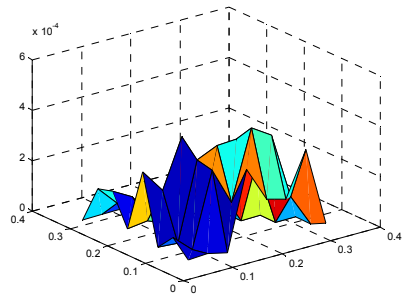
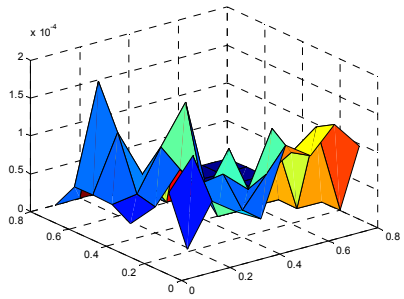
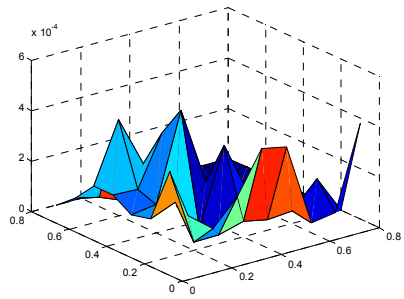
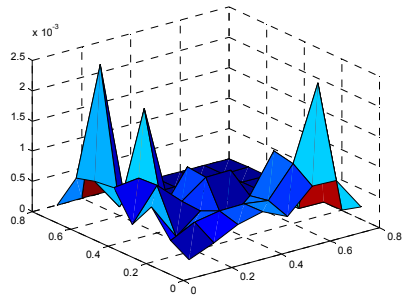
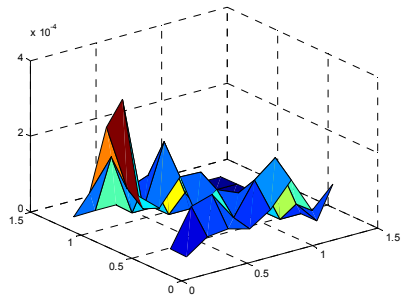
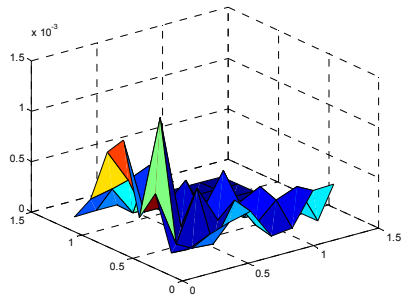
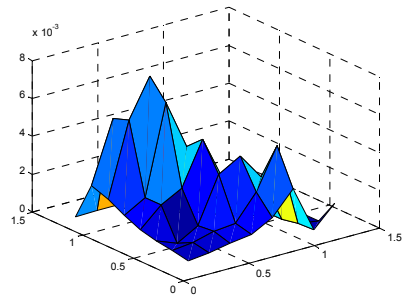
Classe 1	Classe 2	Classe 3
		
		
		

Tableau 10 : La variation de la variance du délai en fonction de la charge des classes 1 et 2 pour l'ordonnanceur SCFQ avec une pondération 6/3/1.

Classe 1	Classe 2	Classe 3
		
		
		

1.6. Conclusion

Dans ce chapitre, nous avons commencé par présenter le protocole MPLS, un des protocoles prometteur pour la mise en place de la QoS dans les réseaux. En effet, il permet plusieurs améliorations dans le choix des chemins et la création de plusieurs tuyaux pour y acheminer tel ou tel autre flux en fonction du souhait du gérant du réseau. De plus, MPLS permet des gains au niveau de la vitesse de commutation, et la protection des routes et des trafics… Quant au mécanisme DiffServ, il permet de différencier les flux et les trafics dans chaque nœud. En fonction de la classe du paquet et des mécanismes associés à cette classe, un comportement spécifique est appliqué au paquet. Avec l'association de DiffServ et de MLPS [23], ces deux méthodes d'implémentations de QoS peuvent coopérer ensemble pour donner de plus en plus de contrôle à l'administrateur du réseau. Cette coopération donne plus d'aspects supplémentaires qu'une simple implémentation indépendante. En dernier lieu, nous avons présenté plusieurs algorithmes d'ordonnancement proposés par les constructeurs de routeurs et tirés de la littérature. Suite à la conclusion du dernier paragraphe, nous avons

utilisé principalement l'algorithme WRR lors de cette thèse, puisqu'il est simple et efficace. De plus, l'inconvénient majeur de WRR est obsolète vu que la taille moyenne de tous les flux que nous avons utilisés a généralement la même valeur, WRR devient alors un algorithme équitable.

CHAPITRE 2
Modélisation du trafic agrégé sortant d'un réseau sans fil UTRAN ou 802.11

Récemment, l'utilisation des mobiles a augmenté énormément et il y a eu beaucoup de travaux afin d'améliorer les divers protocoles sans fil (ad hoc et cellulaire). A partir des réseaux qui fournissent des services de base, le développement actuel est de fournir un ensemble plus riche d'applications. Les performances de divers mécanismes et politiques proposés peuvent dépendre directement des caractéristiques du trafic dans le réseau. Ainsi, les modèles précis pour le trafic et une compréhension de l'impact de divers facteurs sur les caractéristiques du trafic sont nécessaires afin d'améliorer les possibilités des réseaux sans fil en général et développer efficacement les différentes politiques d'admission, les ordonnanceurs et le dimensionnement…

Ce chapitre propose de caractériser les flux sortants d'un réseau sans fil d'accès à l'UMTS (Universal Mobile Telecommunication Service) ou IEEE 802.11. Nous avons utilisé des fichiers traces de simulations de réseaux afin de modéliser le trafic. Nous développons ici un modèle pour le trafic circulant dans les réseaux sans fil IEEE 802.11 et UTRAN (UMTS Terrestrial Radio Access Network) au niveau de la couche MAC. Les résultats peuvent mener à la conception d'outils afin d'améliorer les performances des réseaux mais aussi de mieux modéliser et évaluer ces performances.

Les différents échantillons de trafic sont formés des paquets générés par les applications des utilisateurs: les différents paquets de signalisation, d'enregistrement à la cellule ne sont pas considérés. De même, bien que les différents réseaux étudiés soient mobiles, nous ne traitons ni l'effet du mouvement des terminaux ni le *handover*. Dans chacun des différents scénarios conduits, le nombre d'utilisateurs dans chaque cellule reste le même tout au long de la période de mesure.

Vu les différents travaux effectués dans le domaine de la modélisation, nous avons commencé par comparer les différents flux agrégés avec des processus MMPP (Markov Modulated Poisson Process). Ce dernier est un processus de Poisson doublement stochastique dont le débit est déterminé par l'état d'une chaîne de Markov à temps continu. De plus, la superposition de processus MMPP donne toujours un MMPP avec un nombre de phase plus grand. Un autre avantage des processus MMPP est la facilité d'avoir des résultats analytiques avec la théorie des files d'attente: les performances d'un processus d'arrivée suivant une loi MMPP peuvent être facilement étudiées (MMPP/G/1) contrairement à des processus d'arrivées suivant une loi fractal, auto similaire ou à dépendance longue. Puis, nous comparons le modèle Gaussien aux différents temps d'inter arrivée des paquets que nous disposons.

La première partie de ce chapitre décrit l'architecture du réseau UMTS et de son réseau d'accès UTRAN ainsi que le protocole CDMA (Code Division Multiple Access) utilisé. Le deuxième paragraphe présente les méthodes d'accès radio d'un réseau WiFi 802.11. Le troisième paragraphe expose les différents modèles de flux utilisés dans notre travail. Ensuite, un résumé de plusieurs études d'agrégations de flux On/Off se trouve dans le paragraphe quatre. Ces différents travaux nous ont orienté dans notre travail de modélisation. Le paragraphe cinq présente les algorithmes que nous avons développé afin d'estimer des

paramètres des flux. L'Annexe 1 : présente de plus amples études et comparaisons autour de la modélisation MMPP. Les différents résultats obtenus ainsi que nos contributions sont présentés dans le paragraphe six. Nous achevons le chapitre par la conclusion des résultats de nos études.

2.1. Le réseau UMTS

Plus tard dans ce chapitre, nous étudierons le trafic UMTS sortant du réseau d'accès. Ce trafic est agrégé à la sortie du réseau UTRAN, au niveau du RNC qui gère plusieurs nodes B (Figure 14). Nous avons choisi d'étudier le trafic à ce niveau là pour que l'étude reste valable pour les prochaines versions de l'UMTS. La seule condition est d'avoir la même technologie de l'interface d'accès.

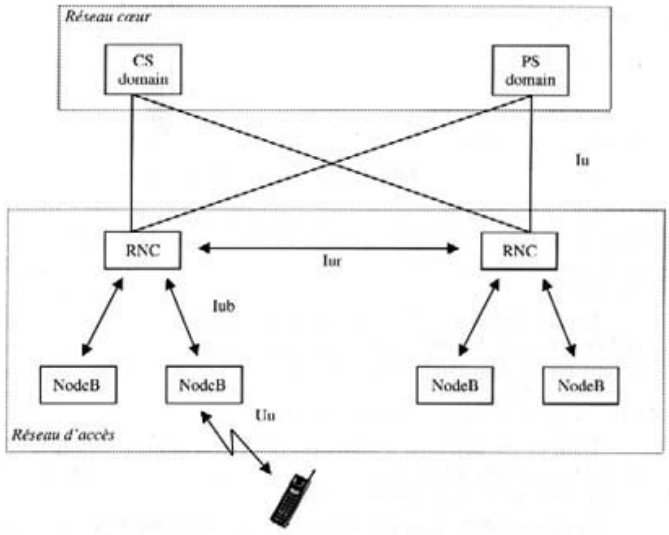


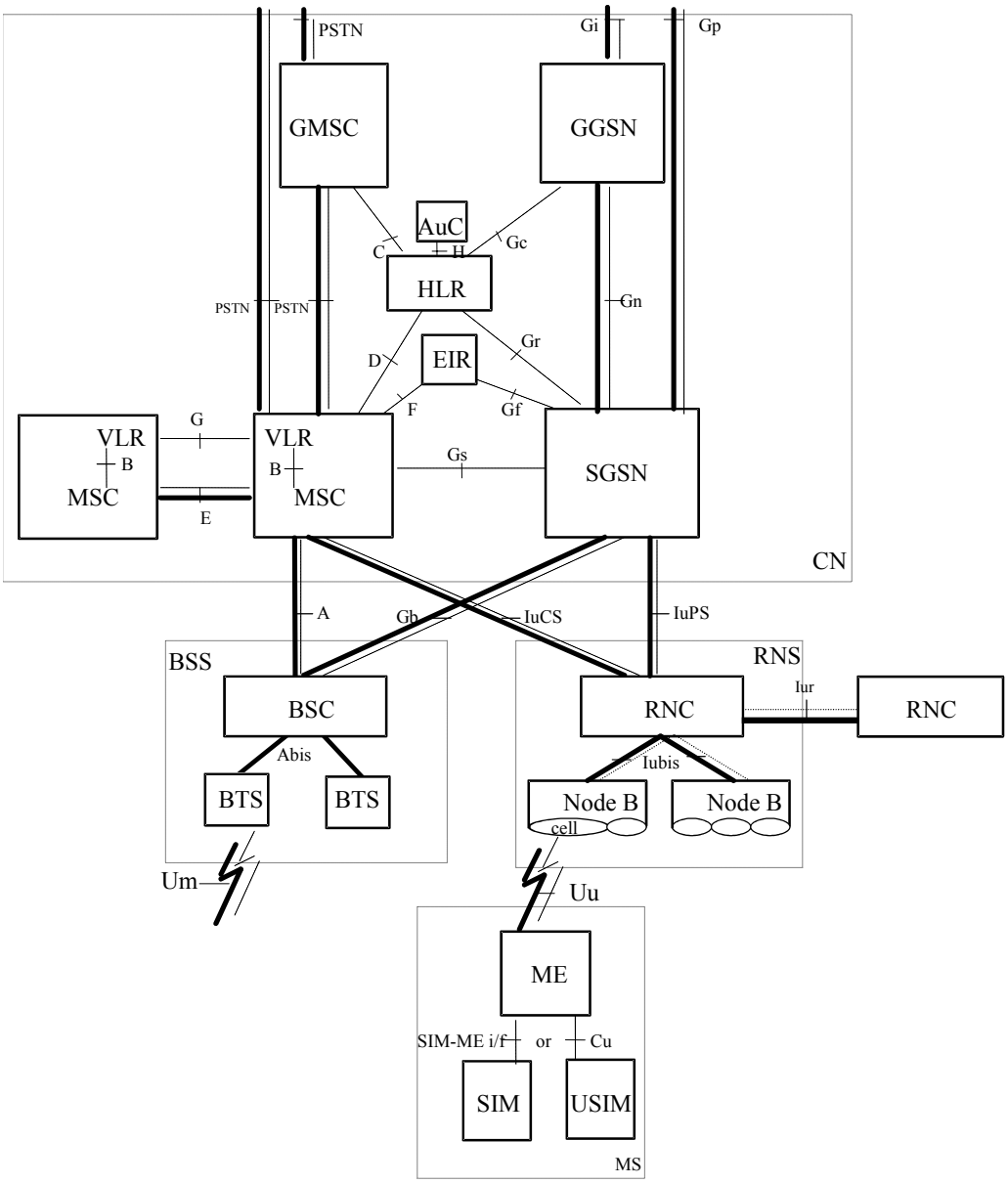
Figure 14: Réseau d'accès UMTS

Cette agrégation de flux peut venir de différentes sources de trafic qui appartiennent à plusieurs classes de service. L'étude est faite au niveau de la couche MAC où une technique d'accès CDMA (Code Division Multiple Access) est utilisée. Le protocole CDMA utilisé ainsi que les services et la fonction de contrôle d'admission sont définis dans [5]. Nous présentons ici brièvement la technique CDMA utilisée puis le protocole d'accès, les services, la fonction CAC et le contrôle de trafic. Tous ces éléments sont essentiels dans la formation du trafic. Les applications utilisées sont représentées par plusieurs sources de trafic qui appartiennent à différentes classes de service allant du temps réel au best effort.

2.1.1. L'architecture du réseau UMTS

Le cœur du réseau (CN Core Network) est la partie fixe du réseau UMTS. Le réseau d'accès, l'UTRAN, a comme fonction principale de gérer la mobilité des utilisateurs. Vu du cœur, les usagers sont fixes. Les interfaces Iu et Uu délimitent le réseau d'accès UTRAN.

Une architecture plus détaillée du réseau et montrant les différents composants réseaux, les trafics utilisateurs et de signalisation, les points d'accès et les interfaces entre les différentes couches de l'UMTS release 99 sont illustrés dans la Figure 15.



Lignes Grasses: Interfaces transportant du trafic usager.
Lignes pointillées: Interfaces transportant du trafic de signalisation.
NOTE 1: La figure montre les inter connections directes entre les entités. Il se peut que cela soit fait physiquement par un réseau (SS7 ou IP).
NOTE 2: Quand le MSC et le SGSN sont dans une même entité physique, nous parlons de UMTS MSC (UMSC).

Figure 15: L'architecture de l'UMTS [21].

2.1.2. La technique CDMA

L'accès en CDMA, utilisé entre autre en UMTS, permet à un grand nombre d'usagers d'utiliser la même onde porteuse mais sans s'interférer les uns avec les autres. Elle consiste à répartir l'information radioélectrique émise sur une bande de fréquences plus large que celle réellement nécessaire à la transmission du signal utile. Ce dernier apparaît alors comme un bruit et sa densité spectrale est constante sur l'intégralité de la bande occupée. De plus, le temps est divisé en trames où plusieurs utilisateurs peuvent envoyer en même temps mais

avec divers codes. Un utilisateur qui veut émettre des paquets, doit choisir d'abord un code puis envoyer ses paquets. Deux choix du code sont possibles: soit le code est choisi aléatoirement, soit le système associe un code à l'utilisateur dès que ce dernier se connecte.

Il s'agit de multiplier au sens mathématique du terme (OU exclusif) chaque bit à transmettre par un code pseudo aléatoire PN (Pseudo random Noise code) propre à chaque utilisateur. La séquence du code (constituée de sf éléments appelés "chips") est unique pour un utilisateur donné et constitue la clef de codage. Elle est conservée si le symbole de donnée vaut 1, inversée sinon. Nous appelons *facteur d'étalement* SF (Spreading Factor) la longueur *sf* du code. Si chaque symbole a une durée Tb (Temps bit), nous avons 1 chip toutes les Tb/sf secondes. Le nouveau signal modulé a un débit sf fois plus grand que le signal initialement envoyé par l'usager. Ainsi, il utilisera une bande de fréquences sf fois plus étendue. La relation entre le débit initial et final est donc :

Débit Chip = Débit Bit * SF

Ainsi plus SF est grand, plus le débit chip, de l'ordre de 3,84 Mcp/s pour le WCDMA (Wideband Code Division Multiple Access) utilisé en UMTS, est grand et plus le débit de données du canal sera élevé. Cela permet de dégager des canaux à débits variables selon les besoins des utilisateurs (bandwidth on demand).

Comme dans plusieurs travaux, [5] suppose que la performance du CDMA est fonction de la probabilité d'erreur d'un bit BER (Bit Error Ratio). Une des approximations pour calculer l'erreur en CDMA est l'approximation standard Gaussienne (SGA Standard Gaussian Approximation). En supposant que l'interférence due à l'accès multiple (MAI Multiple Access Interference) est Gaussienne et en utilisant des récepteurs à simple corrélation, le BER est donnée par $P_e = Q(\overline{SNR})$ sachant que $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-u^2/2} du$. SNR est le rapport signal sur bruit. Si nous considérons la technique de séquence directe en CDMA (CDMA/DS Direct Sequence) ($\Pr\{x_j = 1\} = \Pr\{x_j = -1\} = 0{,}5$) où x_j est un chip dans la séquence avec une longueur de code sf, la valeur moyenne du rapport signal sur bruit pour le paquet i dans le cas des puissances reçues inégales est donnée par $\overline{SNR} = \sqrt{\frac{P_i}{(3sf)^{-1} \sum_{\substack{k=1 \\ k \neq i}}^{K} P_k + \frac{N_0}{2T}}}$ pour un système avec K utilisateurs simultanés ayant chacun une puissance reçue P_j (j=1..K). La durée d'un bit est T et le bruit blanc additionnel est noté $N_0/2$.

Dans un système cellulaire de R+1 cellules de K utilisateurs actifs dans chacune et en supposant un contrôle de puissance parfait (dans une cellule, les puissances reçues par la station de base sont identiques de valeur P_0), le rapport SNR s'écrit en négligeant le bruit blanc et en supposant que le signal de l'utilisateur k dans la cellule i est reçu par la station de base 0 avec une puissance de $P_{(k,i)0}$:

$$\overline{SNR} = \sqrt{\frac{3sfP_0}{(K-1)P_0 + \sum_{\substack{k=1 \\ k \neq i}}^{K} \sum_{i=1}^{R} P_{(k,i)0}}}$$

Dans le cas où l'interférence reçue par les autres cellules est proportionnelle à l'interférence totale dans la cellule considérée avec un facteur de f, le SNR devient :

$$\overline{SNR} = \sqrt{\frac{3sfP_0}{(K-1)P_0 + (K-1)fP_0}} = \sqrt{\frac{3sf}{(K-1)(1+f)}}$$

Nous supposons d'une part que l'atténuation du signal est d'ordre 4 et donc proportionnel à la 4ème puissance de la distance parcourue et d'autre part que le rapport f varie entre 0,44 et 0,75.

Si la longueur d'un paquet est de L et en utilisant un code correcteur d'erreur qui peut corriger jusqu'à t erreurs et si $Q_E = 1 - P_e$ est la probabilité qu'un bit soit bien reçu, la probabilité de succès d'un tel paquet est :

$$Q_E(K) = \sum_{i=0}^{t} C_L^i (1-Q_e)^i (Q_e)^{L-i}$$

La probabilité d'erreur d'un paquet due à l'interférence est: $P_{error}(K) = 1 - Q_E(K)$

Cette probabilité détermine un facteur essentiel dans le calcul de la QoS pour une telle application. Pour expliquer le comportement de cette probabilité, nous illustrons le cas de sf =16, f = 0.5, L = 160 bits, t = 10 sur la Figure 16 qui détermine la probabilité qu'un paquet soit correctement reçu par la station de base en fonction de la charge du système.

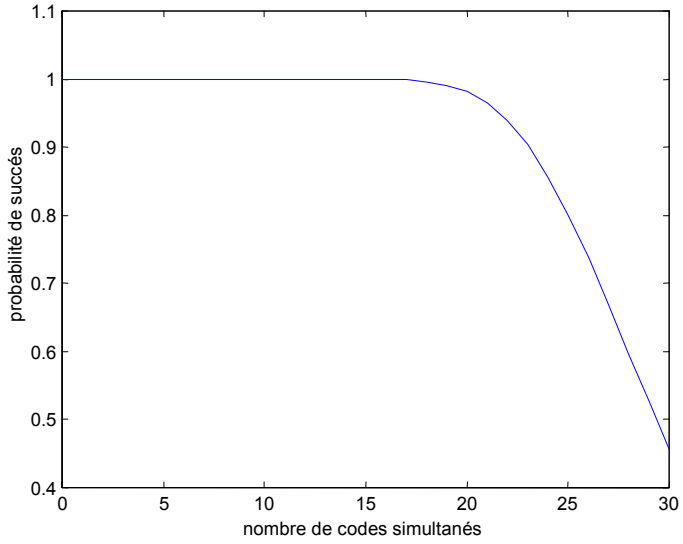


Figure 16: Graphe représentatif du CDMA

L'utilisation de cette technique par les différents services doit garantir une qualité de service bien précise pour chaque utilisateur. Ceci impose différents paramètres pour chaque type de service. Le système peut être divisé en sous-systèmes de voix et de données.

Si le nombre de codes utilisés simultanément est très élevé, la probabilité d'erreur augmente considérablement. Alors, il faudra fixer un nombre maximal de codes utilisables sur une même trame. C'est pourquoi, au début de chaque trame, une probabilité de permission est diffusée aux différents utilisateurs pour limiter l'accès des utilisateurs aux codes. Cette probabilité est une fonction de nombre de codes utilisés dans la trame précédente. La fonction de permission dépend de la qualité de service demandée par l'utilisateur et donc différente pour le sous-système voix et le sous-système data.

Par exemple, d'après la Figure 16 nous remarquons que l'erreur augmente considérablement après une limite donnée. Nous pouvons alors déterminer le seuil pour une QoS acceptable pour les utilisateurs voix et data. Une erreur inférieure à 1% est nécessaire pour les utilisateurs voix et 0.1% pour les utilisateurs data. Ceci donne des seuils de 20 et 18 utilisateurs pour les services voix et data respectivement.

Notons qu'en CDMA, la collision n'est pas définie comme en TDMA vu que plusieurs utilisateurs peuvent émettre des paquets en même temps mais sur plusieurs codes sans risque de perte. Nous observons plutôt une augmentation de l'interférence. Cette dernière,

proportionnelle au nombre d'utilisateurs simultanés, augmente rapidement après un seuil donné. A partir de ce seuil, la probabilité de perte de paquet devient considérable. C'est ce que nous définissons par collision CDMA. La *collision* se produit donc quand la probabilité de perte dépasse la limite de la QoS pour un type de service.

2.1.3. Les services

Vu que plusieurs types d'applications existent et que chacune d'elles demande une QoS différente, et sachant qu'un nombre aléatoire de clients peut accéder au canal radio, plusieurs services sont fournis aux utilisateurs. Du point de vue du temps d'attente, nous pouvons regrouper les utilisateurs en trois types:

- Les utilisateurs qui nécessitent un temps de réponse très court comme ceux utilisant une application de voix.
- Les utilisateurs interactifs qui réclament des délais réduits comme ceux surfant sur internet en utilisant un navigateur Internet
- Les utilisateurs qui ne demandent pas de contraintes temporelles. Les applications de transfert de fichiers ou d'échange de Email sont les meilleures illustrations de ce type d'utilisations.

Ces différents types de services sont supportés au niveau MAC par trois MTCs (MAC Transfer Capabilities) afin de garantir la QoS requise. Ces MTCs sont le RTC (Rapid Transfer Capability), le ITC (Intermediary Transfer Capability) et le STC (Slow Transfer Capability). Ces services sont définis au niveau MAC comme suit:

- **RTC** doit garantir une transmission rapide des paquets générés par la source. Ceci est indispensable pour les **services temps réels**.
- **ITC** doit garantir un délai de transmission modéré aux paquets. Ceci est souhaitable par des **services interactifs** comme le **transactionnelle**. La probabilité de perte de ce genre de paquet doit être très faible.
- **STC** peut supporter un **long délai** avant de transmettre un paquet mais doit garantir une probabilité de perte réduite.

La réalisation de ces contraintes se fait par deux niveaux: l'un au niveau appel et l'autre au niveau paquet. Par le premier niveau, la fonction CAC doit refuser une demande si elle estime que la couche MAC est incapable de garantir le délai requis. Le second niveau de réalisation est effectué aux instants d'émission de chaque paquet par le biais de la couche MAC qui contrôle le délai par l'intermédiaire des probabilités de permission. Le fait que les utilisateurs ITC et STC acceptent un certain délai, l'intégration de tampon dans leurs terminaux est possible, ce qui n'est pas le cas pour les utilisateurs RTC.

La différence entre ces MTCs est réalisée en associant différentes priorités aux différents MTCs. Ainsi, la classe RTC a la priorité la plus haute, vient ensuite la classe ITC et enfin la classe STC. La priorité est déterminée par la fonction de permission associée à chaque classe. Cette méthode octroie des garanties statistiques aux différents utilisateurs en diminuant l'influence d'une classe par rapport aux autres. Une garantie stricte nécessite la séparation des différents utilisateurs soit temporellement soit fréquentiellement.

2.1.4. La fonction CAC

Les indices de la couche MAC, K_v et K_d, sont les valeurs maximales respectives du nombre d'utilisateurs voix et data servis simultanément avec des probabilités d'erreurs acceptables.

Donc:

$P_{error}(k) \leq P_{ev}$ $\forall\ k \leq K_v$ pour les utilisateurs voix.

$P_{error}(k) \leq P_{ed}$ $\forall\ k \leq K_d$ pour les utilisateurs data.

P_{ev} et P_{ed} sont les valeurs maximales respectives des probabilités d'erreurs acceptées par les utilisateurs voix et data. $P_{error}(k)$ est la probabilité d'erreur avec k désignant les utilisateurs voix et data. En pratique $P_{ev} > P_{ed}$ et donc $K_d < K_v$ .Si le nombre total d'utilisateurs simultanés est $k \leq K_d$, tous les utilisateurs voix et data seront reçus avec une qualité acceptable. Si $K_d \leq k \leq K_v$, parmi les k paquets, ceux des connexions voix auront une qualité acceptable à l'opposé des paquets data. Enfin, pour $K_d < K_v \leq k$ tous les utilisateurs auront une qualité de service inacceptable. Par exemple, si P_{ev} = 1% et P_{ed} =0.1%, les indices de la couche MAC sont $K_v = 20$ et $K_d = 18$.

Ces indices ne sont pas les seuls facteurs qui déterminent les limites pour la fonction CAC mais ils sont essentiels pour déterminer les seuils de contrôle de trafic de la fonction de permission. Pour la fonction CAC, l'activité des différents utilisateurs, leurs qualités de service ainsi que la probabilité de rejet jouent des rôles fondamentaux. Or, en CDMA, les différents utilisateurs s'interfèrent; c'est pour quoi il faudra minimiser cette influence afin de respecter la qualité de service de chaque utilisateur.

La fonction CAC utilisée dans [5] est orientée QoS. Elle accepte un utilisateur si elle estime qu'il ne va pas dégrader la QoS des autres utilisateurs et si elle peut respecter sa qualité demandée. Le service voix est représenté au niveau MAC par RTC. ITC et STC représentent les services data.

Un utilisateur RTC, ITC ou STC qui demande une connexion est admis si la partie de ressource qu'il consomme est disponible. De plus, la fonction CAC retranche de l'ensemble des ressources du système la valeur relative à cette réservation. Dans le cas contraire, l'utilisateur n'aura pas accès au système.

2.1.5. Le contrôle de trafic

Le contrôle de trafic est réalisé au niveau MAC par la fonction de permission. Différentes fonctions sont associées aux différents services de façon à réaliser la qualité de service de chaque MTC. Ces fonctions sont définies par plusieurs paramètres et ont le rôle de distribuer les utilisateurs de la façon la plus homogène possible ainsi que de minimiser l'influence mutuelle entre les différentes classes. Pour cette raison, trois fonctions différentes sont définies pour les différents MTCs comme dans la Figure 17.

Dans cette figure, nous fixons les valeurs telles que la probabilité de permission soit la plus grande pour le service RTC (voix). Elle décroît pour atteindre les utilisateurs ITC (web). Elle diminue encore pour atteindre la fonction de permission des utilisateurs STC (data qui comprend le transfert de fichiers et l'Email). Ceci donne une priorité maximale aux utilisateurs RTC puis ITC et enfin STC. Les fonctions possèdent plusieurs paramètres qui sont le point de départ, la pente, le point de déviation et le point de zéro.

Le point de déviation est toujours choisi égal au spreading factor. Les points de départs sont a_1, a_2 et a_3, les angles d'inclinaisons α_1, α_2 et α_3. Les points de zéro sont égaux à S_d pour les utilisateurs web et data puisqu'ils ont la même limite d'erreur tandis qu'il est égal à S_v pour les utilisateurs vocaux. Ce nombre correspond au nombre maximal de codes actifs acceptés avec une QoS convenable.

Les différents paramètres vont être choisis par la suite afin de respecter la qualité de service demandée par chaque utilisateur tout en maximisant la capacité totale du système.

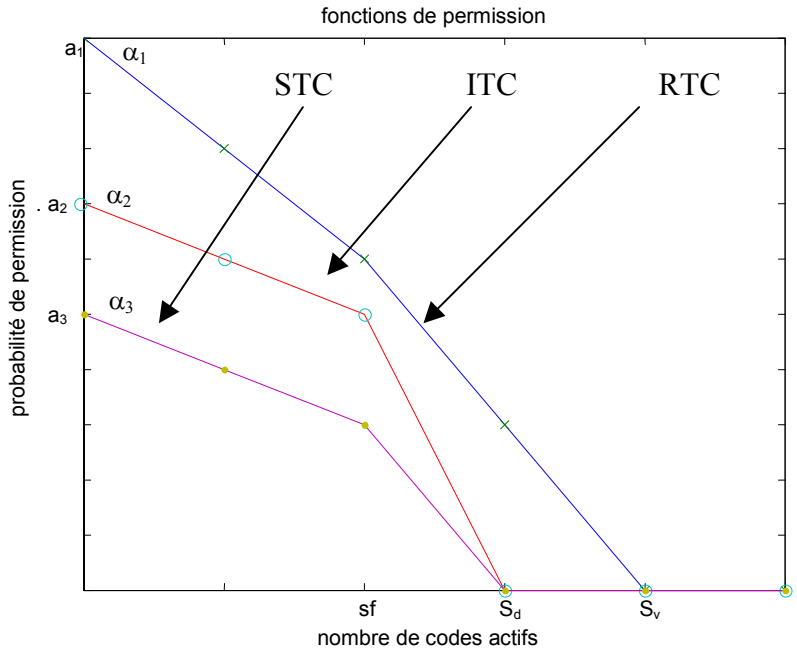


Figure 17: Fonctions de permission

2.1.6. Les applications utilisées

2.1.6.1. Sous système voix

2.1.6.1.1. Application voix (RTC)

Un paquet voix est formé par des bits produits par une source voix pendant T_f secondes et ayant une entête de H_v bits. En utilisant un détecteur d'activité lent (S-VAD Slow Voice Activity Detector) étudié dans [20] par l'ETSI, les sources voix sont considérées comme des sources On/Off. Une source voix crée alors des paquets pendant les durées de parole et n'émet pas de paquets pendant que l'utilisateur est silencieux. Ainsi la source est considérée comme un processus de Markov à deux états Figure 18.

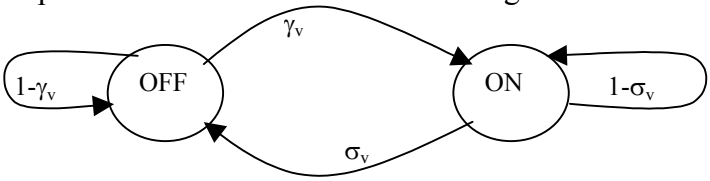


Figure 18: Modèle de source voix

Nous noterons t_{on} et t_{off} les valeurs moyennes des états respectifs On et Off. La probabilité que la source reste en phase On durant un temps τ s'écrit:

$$\sigma_v = 1-exp(-\tau/t_{on})$$

C'est aussi la probabilité de passage de l'état On à l'état Off après un temps τ. De même, la probabilité de passage de l'état Off à l'état On s'écrit:

$$\gamma_v = 1-exp(-\tau/t_{off})$$

La probabilité que le système soit en phase On, c'est à dire en train d'émettre des paquets est:

$$\pi_{on} = t_{on} /(t_{on} + t_{off})$$

Il s'agit du facteur d'activité de la source. La probabilité d'être dans l'état Off s'écrit:

$$\pi_{off} = t_{off} /(t_{on} + t_{off})$$

Quand la source de trafic passe de l'état Off à l'état On, l'utilisateur emploie alors son code afin de transmettre les paquets. C'est à ce moment là que la fonction de permission $p_v(x)$, avec x le nombre de terminaux actifs, est utilisée. L'utilisateur doit réaliser un tirage de Bernoulli de paramètre $p_v(x)$ (il transmet avec une probabilité $p_v(x)$ et ne transmet pas avec une probabilité 1- $p_v(x)$). Si le tirage avec la fonction de permission lui permet d'émettre, le terminal transmettra son paquet en utilisant son propre code et attendra la réponse de la station de base. Si la réponse est positive, il commence à envoyer ses paquets. Dans le cas contraire, il attend la durée d'un slot et refait le tirage de Bernoulli.

Or, un paquet voix ne peut pas attendre plus qu'un temps D_{max} secondes avant d'être envoyé. Si le terminal n'arrive pas à accéder au canal au bout du temps D_{max}, le paquet sera rejeté. C'est la première raison de perte. Elle sera noté par la probabilité de rejet P_{drop}. La deuxième méthode de perte est l'erreur due à l'interférence des autres utilisateurs CDMA. Elle sera notée P_{error}. Ces deux raisons contribuent alors à former la probabilité globale de perte d'un paquet qui s'exprime par :

$$P_{loss} = P_{error} + P_{drop}$$

2.1.6.2. Sous système data

Le sous-système data englobe les transferts de fichiers (FTP File Transfer Protocol) et du trafic Web.

2.1.6.2.1. Application web (ITC)

Le web représente une bonne partie du trafic de data dans l'UMTS. Il est modélisé par des sources décrites dans [22]. La couche MAC est transparente au trafic de données, mais la couche RLC insère une entête de 2 octets à chaque TB (Transport Block). La simulation n'envisage que le trafic de type *web browsing* Downlink (UTRAN→ UE). C'est un cas réaliste vu que le trafic Uplink est faible; il représente les requêtes de demande des pages web à afficher.

Chaque **session** consiste en un nombre de ***packet calls***. Le *packet call* modélise une réception d'une page *web*. Cette page est transmise en plusieurs **paquets** ou datagrammes. Donc chaque *packet call* génère une rafale de paquets. C'est là une des caractéristiques des trafics de données. Quand ces paquets arrivent à son terminal, l'usager doit lire la page *web* qu'il a demandée: c'est le temps de lecture d'un *packet call*.

Donc les paramètres qui décrivent le trafic web sont:

- Le processus d'arrivée des sessions.
- Le nombre de *packet calls* par session: D_{pc}
- Le temps de lecture entre *packet calls*: N_d.
- Le temps inter arrivé des datagrammes dans un même *packet call*: D_d.
- La taille d'un datagramme: S_d.

La durée d'une session peut être calculée de ces diverses données.

La Figure 19 donne un exemple de génération de paquets de données.

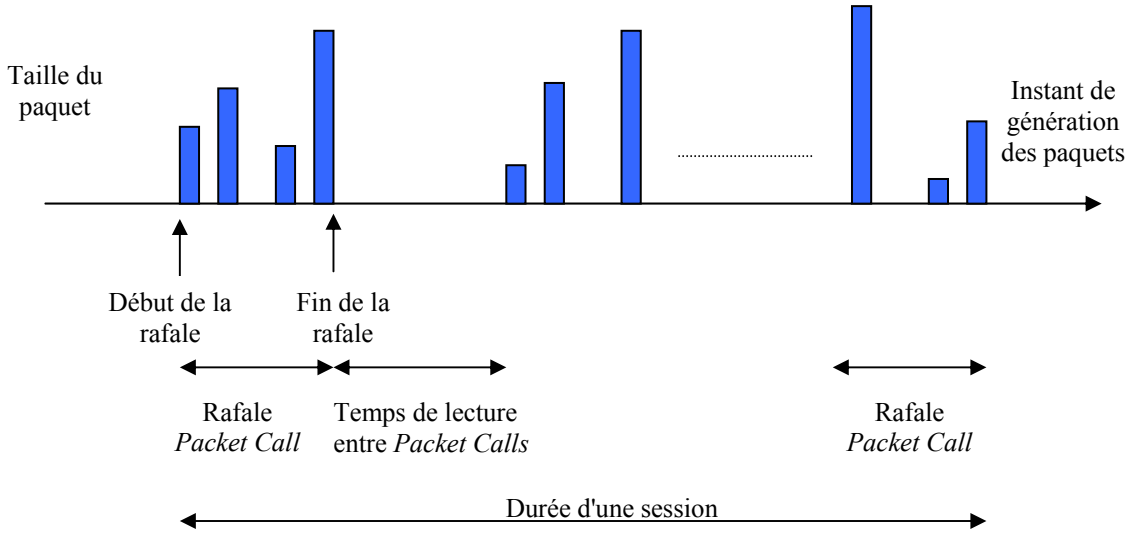


Figure 19: Exemple de génération de paquets.

Les différentes valeurs utilisées dans les simulations sont données dans le Tableau 11.

Tableau 11: Valeurs des différents Paramètres.

		Arrivée des sessions	Nombre de *Packet Calls* par session	Temps de lecture entre deux *Packet Call*	Le nombre de paquet dans un *Packet Call*	Temps entre deux paquets consécutifs dans le même *Packet Call*	Taille d'un paquet
		Processus de Poisson	Distribution Géométrique	Distribution Géométrique	Distribution Géométrique	Distribution Géométrique	Distribution de Pareto tronqué
Paramètres	UDD 8 kbps		Moyenne $\mu_{N_{PC}} = 5$	Moyenne $\mu_{D_{PC}} = 412\,s$	Moyenne $\mu_{N_D} = 5$	Moyenne $\mu_{D_d} = 0.5$	Paramètre α=1.1 Taille minimale k =81.5octets Taille maximale *m*
	UDD 16 kbps					Moyenne $\mu_{D_d} = 0.125$	
	UDD 64 kbps					Moyenne $\mu_{D_d} = 0.0625$	
	UDD 144 kbps					Moyenne $\mu_{D_d} = 0.0277$	
	UDD 384 kbps					Moyenne $\mu_{D_d} = 0.0104$	
	UDD 2048 kbps					Moyenne $\mu_{D_d} = 0.00195$	

La distribution géométrique est la représentation discrète de la distribution exponentielle. La probabilité d'une variable exponentielle s'écrit:

$P(y=n) = p\ (1-p)^{n-1}$ p étant le paramètre de la distribution.

La moyenne d'une distribution géométrique est l'inverse de son paramètre.

La fonction de distribution d'une variable aléatoire qui suit la loi de Pareto tronqué s'écrit:

$$f(x) = \begin{cases} \dfrac{\alpha k^{\alpha}}{x^{\alpha+1}}, & \text{pour } k \le x \le m \\ \beta, & \text{pour } x = m \end{cases}$$

où β est la probabilité pour x soit supérieure à m. Donc β s'écrit:

$$\beta = \int_{m}^{+\infty} f(x)dx = \left(\frac{k}{m}\right)^{\alpha}, \alpha > 1$$

Le modèle proposé suppose que la taille minimale d'un paquet k soit de 81,5 octets. Sa taille maximale est de 1502 octets, qui est la taille maximale d'un datagrame IP (1502octets-taille maximale pour un paquet IP sur un réseau Ethernet). Le facteur α est égal à 1,1.

Ce trafic, qui est interactif, est considéré dans le sous-système data mais présente des exigences plus strictes que le transfert des fichiers et l'Email en terme de délai d'attente. En effet, un utilisateur web préfère ne pas attendre très longtemps pour une requête d'informations sur le web, ce qui exige un accès rapide de la source web sur le canal afin de délivrer les informations demandées. Pour cette raison, la probabilité d'accès au canal du web est différente du transfert des fichiers considérés comme service STC.

2.1.6.2.2. Application transfert de fichiers (STC)

Les sources de transfert de fichiers sont constituées d'un nombre d'utilisateurs qui génèrent des rafales suivant un processus de Poisson indépendant, de moyenne λ. Chaque rafale suit une distribution géométrique de longueur moyenne L_b bits.

Une source de donnée peut être modélisée comme un processus de Markov à deux états. Le premier état est "Attente A" qui représente l'attente d'un fichier, et le deuxième état "Fichier F", qui représente une source qui admet un fichier à transmettre. Le passage de l'état "Attente A" à l'état "Fichier F" est Poissonnien de paramètre λ. Cette probabilité de passage après une durée τ est donc $\sigma_d = 1 - exp(-\lambda\tau)$. La probabilité de passage de l'état F à l'état A après un séjour de durée τ est $\gamma_d = \tau / (L_b / R_s)$ avec R_s le débit de la source de trafic. C'est la durée τ divisée par le temps total moyen pour émettre un message qui est égal à L_b / R_s.

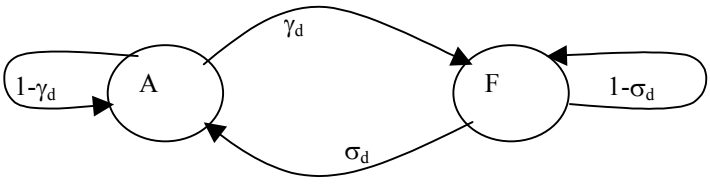


Figure 20: Modèle de source data

Un utilisateur data qui désire émettre des paquets essaie tout d'abord d'accéder à la trame en faisant un tirage de Bernoulli de paramètre $p_f(x)$ ou $p_w(x)$ (transfert de fichiers ou web), où x est le nombre d'utilisateurs dans la trame précédente. Si l'utilisateur échoue son tirage, il essaie sur la trame suivante de la même façon. Dans le cas où l'utilisateur

parviendrait à y accéder, il envoie son paquet sur la trame et attend la réponse de la station de base concernant le nombre total de codes actifs. Si ce nombre est inférieur à un seuil S_d, l'utilisateur passe à l'état de réservation et utilise son code choisi tout au long de la rafale. Dans le cas contraire, il revient à l'état de contention. Un paquet data peut attendre longtemps avant d'être envoyé; pour cette raison le paquet est stocké dans un tampon en attendant l'accès au canal. Par contre, un paquet envoyé et perdu à cause de l'interférence CDMA ne peut pas être retransmis au niveau MAC. C'est au niveau de la couche RLC que la retransmission sera accomplie. Ceci représente la seule raison de perdre un paquet qui est la probabilité d'erreur :

$$P_{loss} = P_{error}$$

De plus, le temps d'attente d'un paquet dans le tampon dépend directement de la fonction de permission d'accès au canal qui, quant à elle, est inversement proportionnelle à la probabilité de perte. En effet, quand la fonction de permission est restrictive, le temps nécessaire pour accéder au canal augmente ainsi que le temps de séjour d'un paquet dans le tampon tandis que le nombre total de codes actifs et la probabilité d'erreur diminuent. Par contre, quand la fonction de permission est généreuse, le temps d'accès au canal diminue de même que le temps de séjour d'un paquet dans le tampon. Néanmoins, le nombre total de codes actifs sur un slot et la probabilité d'erreur augmentent. Un compromis entre les deux est réalisé par le choix des paramètres du système comme la fonction de permission et le spreading factor.

Notons enfin que les tampons utilisés sont supposés de taille infinie et qu'aucune perte n'est engendrée par le remplissage des tampons. Mais le temps de séjour moyen est un facteur de QoS à respecter suivant l'utilisateur en question (Email, web…).

2.2. Le réseau WiFi 802.11

2.2.1. Introduction

Le réseau sans fil IEEE 802.11 [29] est un médium de communication partagé entre plusieurs stations. C'est un des réseaux sans fil les plus répandus et les plus utilisés dans le monde industriel et domestique. Il pourra jouer un rôle primordial dans les futurs réseaux multimédias professionnels et personnels. L'avantage principal de ce type de réseau est sa simplicité, son facteur de mise à l'échelle favorable et sa robustesse face aux pannes. Le groupe de travail 802.11 de l'IEEE normalise, à partir de la fin des années 90, l'accès à la bande de fréquence industrielle, scientifique et médicale (ISM) de 2.4 GHz et la bande de fréquence non licencée autour de 5 GHz. Ici, nous nous concentrons sur les implications de ce protocole relatif aux caractéristiques du trafic.

Le standard IEEE 802.11 [29] définit simultanément la couche physique (PHY) et la couche MAC des réseaux sans fil.
La couche PHY peut utiliser un des modes suivants:

- Direct Sequence Spread-Spectrum (opérant dans la bande de fréquence de 2,4 à 2,4835 gigahertz);
- Frequency-Hoping Spread-Spectrum (opérant dans la bande de fréquence de 2,4 à 2,4835 gigahertz);
- Modulation d'impulsion infrarouge (opérant dans la bande de fréquence de 300 à 428 gigahertz).

Dans la première version, le débit de canal peut être 1 Mb/s ou 2 Mb/s. Les versions ultérieures (802.11b, 802.11a, 802.11g…) offrent des débits supérieurs. Mais, dans cette thèse, nous nous limiterons à la première version. D'ailleurs, dans ces différentes versions, seul le débit est modifié. Le fonctionnement principal de la couche MAC reste inchangé (sauf

pour 802.11e, où le standard n'est pas encore achevé). Nous présentons dans le paragraphe suivant une courte description de la couche MAC 802.11.

2.2.2. Le protocole MAC 802.11

Les réseaux 802.11 admettent deux types de configurations différentes: le mode ad hoc et le mode infrastructure. Dans un réseau ad hoc, les ordinateurs sont rassemblés dynamiquement pour former un réseau. Il n'y a aucune structure définie et des ordinateurs quelconques peuvent communiquer directement tant qu'ils sont dans la zone de réception les uns des autres. Par le mode infrastructure, un point d'accès (AP Access Point) est requis pour connecter les terminaux au système distribué (DS Distributed System). Ainsi, chaque terminal peut communiquer avec les autres via ce point d'accès, qui est équivalent à une station de base.

La couche MAC IEEE 802.11 gère l'accès au canal en utilisant une fonction distribuée de coordination (DCF Distributed Coordination Function) basée sur le protocole CSMA/CA (Carrier Sense Multiple Access with Collision avoidnance). Cette méthode est optimale pour un accès asynchrone entre les terminaux. Une alternative au DCF est également disponible sous forme de fonction de coordination de point (PCF Point Coordination Function) qui est un système centralisé qui autorise l'utilisateur à transmettre. Comme indiqué dans le standard, l'implémentation de PCF est optionnelle, alors que celle de DCF est obligatoire. C'est pourquoi cette dernière méthode est la méthode basique pour les deux modes ad hoc et infrastructure.

Avec DCF, la technique d'accès est CSMA/CA qui est conçu pour réduire les collisions dues à la transmission simultanée et multiple de plusieurs sources. Chaque station vérifie si le canal est libre avant d'essayer de transmettre des paquets. Si l'interface radio est inoccupée pendant une durée DIFS (Distributed Interframe Space), la transmission peut commencer immédiatement. Si la bande radio est occupée, la station reportera sa décision jusqu'à la fin de la transmission courante. Une fois le canal libre, la station choisira un intervalle de temps aléatoire, noté *backoff*, et le décrémentera d'une valeur d'un slot tant que le canal est libre. Si le canal redevient occupé, la valeur du *backoff* reste la même jusqu'à la fin de la transmission actuelle. Le nombre aléatoire *backoff* appartient à l'intervalle allant de 0 à la fenêtre de contention (CW Congestion Window). Une fois que la valeur du *backoff* devienne nulle, le terminal pourra alors entamer son émission. Si la transmission échoue, le protocole suppose qu'il y a eu une collision. Dans ce cas, la fenêtre de contention est doublée et un nouveau procédé de *backoff* commence. Le processus recommencera jusqu'à ce que la transmission réussisse ou soit rejetée.

La durée de l'intervalle de *backoff* est aléatoire et comprise entre 0 et CW. Elle est déterminée comme suit:

$$Temps_de_Backoff = Random() * SlotTime$$

où *Random()* est un nombre entier pseudo-aléatoire tiré d'une distribution uniforme de l'intervalle [0, CW].

CW est un nombre entier appartenant à l'intervalle CW_{min} et CW_{max} caractérisé par PHY, la couche radio physique. En d'autres termes, $CW_{min} \leq CW \leq CW_{max}$. De même, la durée du slot est égale à la valeur des caractéristiques correspondant à PHY. Le paramètre CW prend initialement la valeur de CW_{min}. CW prendra ensuite la prochaine valeur de la série après chaque transmission non réussie jusqu'à ce qu'il atteigne la valeur de CW_{max}. Une fois cette dernière valeur atteinte, la valeur de CW reste inchangée jusqu'à ce qu'elle soit initialisée. Ceci améliore la stabilité du protocole d'accès dans des conditions de charge élevées. CW sera remis à CW_{min} après chaque tentative réussite de transmission d'un paquet. L'ensemble des valeurs CW est la suite des puissances du nombre entier 2 moins 1,

commençant par la valeur spécifique de PHY CW_{min} et s'accroissant jusqu'à la valeur de CW_{max}. Le procédé de *backoff* est utilisé afin de réduire la possibilité de collision en choisissant un temps aléatoire différent pour chaque terminal. Ce procédé est illustré dans la Figure 21 [29].

L'effet de l'introduction du *backoff* est que chaque station reporte sa transmission aléatoirement et indépendamment des autres stations. Ceci implique que la probabilité pour que deux stations choisissent la même valeur soit très faible. C'est la station avec le plus petit temps de *backoff* qui émettra avant les autres. Comme CW_{min} et CW_{max} sont fixés pour un PHY donné, DCF ne différencie ni les trafics ni les stations: toutes les stations et toutes les classes de trafic ont la même priorité pour accéder au medium sans fil (WM Wireless Medium). Ainsi, DCF ne supporte pas les applications réclamant différents délais ou différents débits.

Quant à la technique PCF, elle fournit une méthode d'accès sans contention. Un point central coordinateur (PC Point Coordinator) donne le droit d'émission aux différentes stations. Ainsi le PC réside dans le AP. Cette technique d'accès PCF n'est donc disponible que sur la configuration de réseau d'infrastructure. PCF admet un avantage par rapport à DCF vue qu'elle peut initier une transmission après un temps d'attente inférieur à DIFS. L'intervalle de temps d'attente utilisé pour PCF s'appelle l'espace interframe de PCF (PIFS PCF interframe space). Une fois que le AP commande le WM, il scrute les stations suivant une liste d'appel. Cette liste associe une priorité de sollicitation aux différentes stations de données durant la période de non contention.

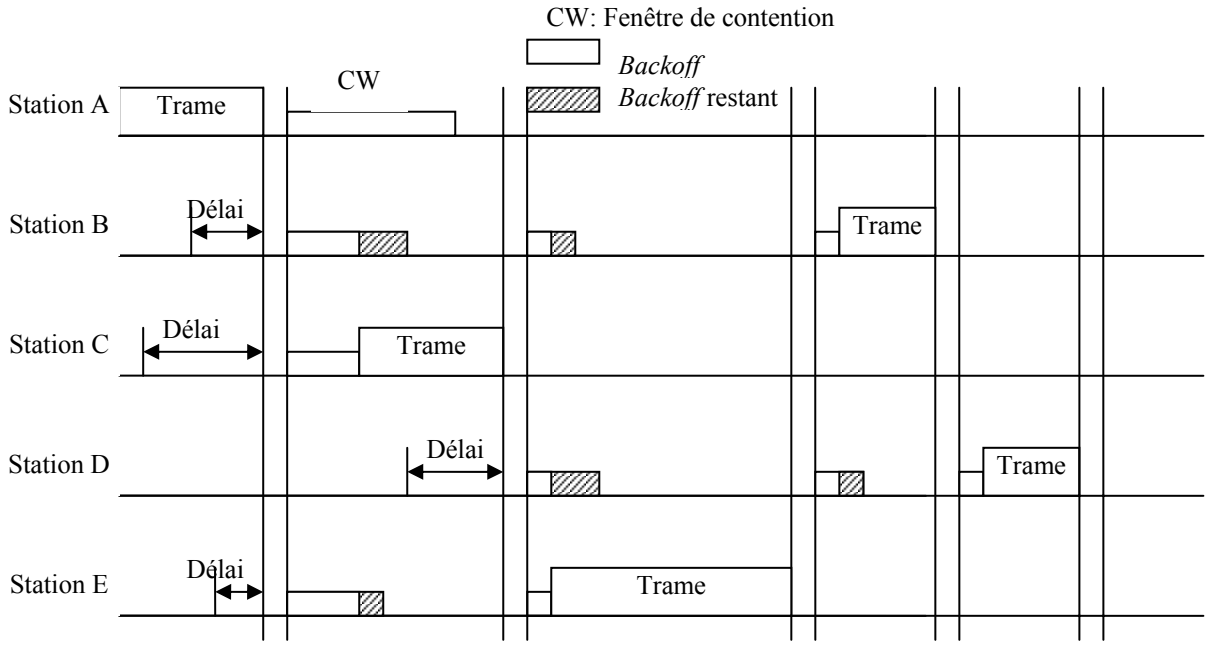


Figure 21: Le procédé de *Backoff* et de retansmission dans 802.11.

DCF et PCF doivent coexister dans la même station de base. Ainsi, pour une station PC opérant en PCF, les périodes de non contention (CFP Contention Free Period) et de contention (CP contention Period) s'alternent au cours du temps. Pendant la période de non contention, une station peut transmettre seulement si elle est autorisée. Pendant le CFP, le PCF est utilisé pour l'accès au medium, alors que le DCF est employé pendant le CP. Le PCF peut supporter quelques applications temps réels.

En évitent la collision, le protocole 802.11 utilise un schéma d'acquittement positif (ACK). Tous les paquets reçus par la couche MAC 802.11 doivent être acquittés par le récepteur: après chaque réception d'un paquet, le récepteur attend un brève laps de temps, appelé l'espace interframe court (SIFS Short Interframe space), avant qu'il ne transmette le ACK de la couche MAC.

Une autre particularité des réseaux locaux sans fil (LANs), connue sous le nom du problème des "nœuds cachés" est traitée dans les spécifications du 802.11. Deux stations qui ne peuvent pas s'écouter mutuellement, peuvent engendrer des collisions au niveau d'un troisième noeud qui reçoit la transmission de ces deux sources. Pour résoudre ce problème, le protocole MAC de l'IEEE 802.11 utilise la réservation. Une station qui voudrait transmettre un paquet, envoie tout d'abord un paquet "prêt à envoyer" (RTS Ready to Send) au récepteur et ce dernier répond avec un paquet CTS (Clear to Send) s'il peut accepter le paquet et n'est donc actuellement pas occupé. Cet échange de RTS/CTS, qui contient également des informations de synchronisation sur la longueur de la transaction suivante, est détecté par tous les noeuds sur la distance d'audition de l'émetteur et/ou du récepteur. Ainsi l'émission est reportée jusqu'à ce que la transmission courante soit complète. L'opération de base du protocole MAC basée sur CSMA/CA de l'IEEE 802.11 est montrée dans la Figure 22 et présente l'échange de divers paquets impliqués par chaque transmission réussie et l'espacement entre ces paquets.

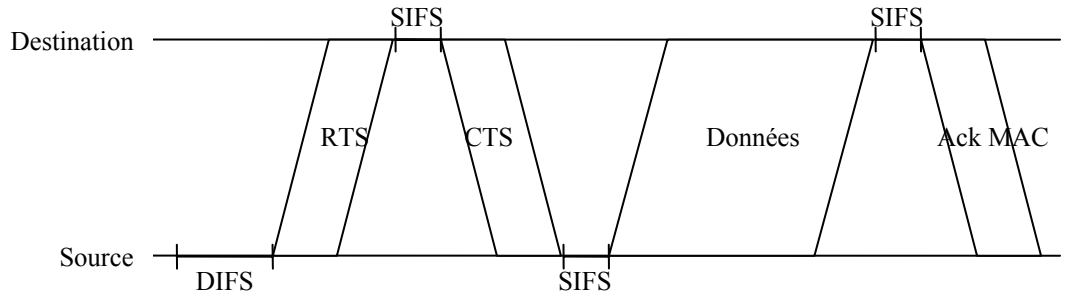


Figure 22: Communication entre la source et la destination en 802.11.

2.3. Modèle de flux

Dans cette section nous présentons les différents modèles de simulation définis afin de caractériser le trafic sortant du réseau UTRAN et WiFi. Ces modèles seront ensuite utilisés par le simulateur événementiel NS (Network Simulator) pour étudier l'agrégation de trafic venant de plusieurs utilisateurs. Dans l'étude du réseau UTRAN, chaque RNC regroupe 10 cellules et donc 10 nodes B. Pour des raisons de simplicité, nous supposons que les cellules sont équitablement chargées. Nous proposons une modélisation pour une agrégation de sources voix seules, des sources web seules, des sources de transfert des fichiers seules et enfin un multiplexage des trois types de sources sur l'interface radio. Pour les scénarios utilisant un réseau WiFi, plusieurs sources de trafics dans une même cellule émettent vers le même point d'accès.

2.3.1. Sources voix

Dans les deux configurations des réseaux UMTS et WiFi, les flux voix sont modélisés par des sources On/Off de paramètres t_{on} = 1 seconde et t_{off} = 1,35 secondes.

2.3.2. Sources web

Comme le débit utilisé est de 8Kb/s, dans chaque cellule, chaque source web admet les paramètres suivants (Tableau 11):

Le nombre d'appels dans une session est géométrique de moyenne N_p= 5.

Le temps de lecture est géométrique de moyenne T_R = 412s.

Le nombre de rafales dans un appel est géométrique de moyenne N_b = 25.

Le temps séparant deux rafales consécutives est géométrique de moyenne T_{sep}= 0.5s.

Enfin les paramètres de distribution de Pareto sont α = 1.1, k = 81.5 et la moyenne d'une rafale est L_w = 24 paquets.

Un contrôle de permission est utilisé pour limiter le nombre maximal des codes sur le canal. Cette fonction pour sf=16 est décrite comme suit:

$$f(x) = \begin{cases} 0.3 - 0.01x & x < sf \\ \max(0, 1.26 - 0.07x) & x \geq sf \end{cases}$$

Ce modèle est uniquement utilisé dans le réseau UTRAN.

2.3.3. Sources de transfert des fichiers

Dans le réseau UTRAN, les paramètres sont: le taux d'arrivée est $\lambda = 0,25$ et la longueur moyenne d'une rafale est L_s = 100 paquets = 16000 bits. Une fonction de contrôle de permission très restrictive est utilisée. Cette fonction pour un sf = 16 est décrite comme suit :

$$f(x) = \begin{cases} 0.2 - 0.01x & x < sf \\ \max(0, 0.36 - 0.02x) & x \geq sf \end{cases}$$

Dans le réseau WiFi, les flux FTP sont modélisés par des sources infinies qui sont contrôlées par TCP.

2.3.4. Multiplexage de service

Dans le réseau UTRAN, la priorité à l'accès aux ressources radio est attribuée par la fonction de probabilité de permission associée à chacune des sources. Les fonctions de permission pour les flux web et data sont les mêmes que celles utilisées dans les paragraphes précédents. Pour les sources de voix dont la limite de QoS est pour K_v = 20 utilisateurs, nous avons utilisé la fonction suivante:

$$f(x) = \begin{cases} 1 & x < K_v \\ 0 & x \geq K_v \end{cases}$$

Comme le réseau Wifi utilisé ne différencie pas les flux, l'accès aux ressources radio n'est pas différencié entre les différents types de connexions demandées. Dans notre cas, comme les flux FTP sont contrôlés par TCP, leur débit change en fonction de la charge du système. Par contre, les sources voix sont indépendantes de la QoS disponible dans le réseau.

2.4. Agrégation de flux On/Off

Comme nous avons présenté plus haut, les sources des flux voix émettent des paquets suivant un processus On/Off. Les intervalles de temps On et Off sont indépendants et suivent chacune une loi exponentiellement distribuée. Au niveau de la couche Mac radio de la station

de base, les flux des différents utilisateurs sont multiplexés suivant le protocole et les mécanismes mis en place.

Plusieurs travaux ont essayé de décrire analytiquement l'agrégation de flux On/Off. Bien que des méthodes différentes sont proposées et analysées, la seule constante inchangée dans ces diverses références est que l'agrégation de flux On/Off est un flux MMPP.

Dans [28], les auteurs proposent un modèle d'agrégation formé d'une somme de deux types de sources On/Off. Chaque type admet des temps de phase On et Off ainsi que des débits indépendants les uns des autres.

Tableau 12: Les éléments recherchés

	Source de type 1	Source de type 2
Matrice de transition	$\begin{pmatrix} -a_1 & a_1 \\ b_1 & -b_1 \end{pmatrix}$	$\begin{pmatrix} -a_2 & a_2 \\ b_2 & -b_2 \end{pmatrix}$
Débit en phase On	r_1	r_2
Nombre de sources	n_1	n_2

Afin de définir le nombre et les caractéristiques des sources, 4 éléments sont utilisés: le débit maximale, le débit moyen, le débit moyen d'ordre 2 et le débit moyen d'ordre 3. En définissant la valeurs moyennes du nombre d'arrivée de paquets dans un intervalle de temps τ par $N(t) = \int_0^t R(\tau) d\tau$, nous pouvons aussi définir $N_1(t)=E[N(t)]$, $N_2(t)=E[(N(t)-N_1(t))^2]$ et $N_3(t)=E[(N(t)-N_1(t))^3]$ les trois moyennes d'ordre respectif 1, 2 et 3.

Les auteurs définissent les deux indices de dispersions I_1 et I_2 tel que $I_1(t) = \frac{N_2(t)}{N_1(t)}$ et $I_2(t) = \frac{N_3(t)}{N_2(t)}$. En minimisant l'erreur entre les valeurs théoriques et de mesure sur ces indices, les différentes valeurs des paramètres recherchés seront obtenues.

Dans [1], les auteurs approximent une agrégation de sources On/Off par un processus MMPP à deux états. Les différentes comparaisons, essais et mesures comparent les taux de pertes obtenus avec le système agrégé et avec la modélisation MMPP-2.

En définissant α comme le temps moyen de transition de l'état Off à l'état On, β le temps moyen de transition de l'état On à l'état Off et λ le taux d'émission durant la phase On, nous définissons:

F_p: le débit maximal

L_l: la taille moyenne des paquets

L_B: la taille moyenne de la rafale

p: le taux d'activité défini comme étant le rapport entre le débit moyen et le débit maximale

Le temps moyen de transition de l'état Off à l'état On $\alpha=p.F_p/(1-p)L_B$,

Le temps moyen de transition de l'état On à l'état Off $\beta=F_p/L_B$,

Le taux d'émission durant la phase On $\lambda= F_p/L_l$,

et M le rapport de la capacité du lien C sur le débit maximal de la source F_p.

Dans le cas où le nombre N de sources à multiplexer est inférieur à M, nous sommes sûr que tous les paquets passeront le lien sans problème et que les seules altérations de la distribution

viennent de l'attente dans les files. Dans ce cas précis, le multiplexage des paquets donne une loi exponentielle simple de paramètre λ.

Dans le cas contraire, (N supérieure à M), un processus MMPP à deux états approxime bien le comportement de l'agrégat. Les paramètres à déterminer sont: $\begin{pmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{pmatrix}$ et $\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix}$. σ_1 (resp. σ_2) est le taux moyen de transition de l'état 1 (resp. état 2), λ_1 (resp. λ_2) le taux d'arrivée moyen dans l'état 1 (resp. 2).

Il est montré dans [1] et dans [38] que le taux moyen de transition σ_2 de l'état 2 à l'état 1 est la valeur propre maximale négative réelle de la matrice Q de transition suivante:

$$Q = \begin{pmatrix} -(\lambda_{M+1} + \mu_{M+1}) & \lambda_{M+1} & 0 & \dots & 0 \\ \mu_{M+2} & -(\lambda_{M+2} + \mu_{M+2}) & \lambda_{M+2} & & 0 \\ 0 & \mu_{M+3} & -(\lambda_{M+3} + \mu_{M+3}) & \lambda_{M+3} & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & 0 & \mu_N & -\mu_N \end{pmatrix}$$

avec $\lambda_{M+i}=(N-(M+i))\beta$ pour i=1,…,N-M
et $\mu_{M+i}=(M+i)\alpha$ pour i=1,…,N-M

$$\lambda_1 = \lambda \sum_{i=0}^{M} i \frac{\pi_i}{\sum_{j=0}^{M} \pi_j} \text{ et } \lambda_2 = \lambda \sum_{i=M+1}^{N} i \frac{\pi_i}{\sum_{j=M+1}^{N} \pi_j} \quad \sigma_1 = \sigma_2 \frac{Np\lambda - \lambda_1}{\lambda_2 - Np\lambda}.$$

avec $\pi_j = \frac{N!}{j!(N-j)!} p^j (1-p)^{N-j}$

Dans [27] la superposition de plusieurs flux de voix et de donnée est supposée MMPP à deux états. Le flux de sortie est choisi de telle sorte que plusieurs de ses caractéristiques statistiques soient préservées. Comme un processus MMPP-2 admet 4 paramètres, quatre éléments sont choisis pour tirer les différents éléments:

- Le taux d'arrivée moyen.
- Le rapport variance sur la moyenne du nombre d'arrivée sur l'intervalle (0,t).
- Le rapport variance sur la moyenne du nombre d'arrivée sur l'intervalle (0,t) quand $t \to \infty$.
- le moment d'ordre 3 du nombre d'arrivée dans l'intervalle (0,t).

Les différentes équations résultantes sont basées sur des approximations et sont fonctions du nombre d'échantillons dans l'intervalle (0,t).

Comme dans [15], nous proposons d'augmenter le nombre de phases dans le processus MMPP pour mieux prendre en compte les différentes caractéristiques statistiques des paquets.

2.5. Méthodologie d'estimation

2.5.1. Estimation MMPP

En utilisant la notation du paragraphe précédent:

$$Q = \begin{pmatrix} \sigma_{1,1} & \sigma_{1,2} & & \cdots & \sigma_{1,N} \\ \sigma_{2,1} & \sigma_{2,2} & \sigma_{2,3} & & \sigma_{2,N} \\ \vdots & & \ddots & & \vdots \\ & \cdots & \sigma_{N-1,N-2} & \sigma_{N-1,N-1} & \sigma_{N-1,N} \\ \sigma_{N,1} & \cdots & & \sigma_{N,N-1} & \sigma_{N,N} \end{pmatrix} \text{ et } \Lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{N-1} \\ \lambda_N \end{pmatrix}$$

Nous définissions les probabilités stationnaires des états par le vecteur ligne $\Pi = (\pi_1, \pi_2, ...\pi_N)$, la solution des équations:

$$\begin{cases} \Pi \cdot Q = 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases}$$

Avec cette notation, nous pouvons exprimer:

- La distribution d'une fonction MMPP à deux états:

$$f(x) = \frac{\pi_1 \lambda_1 e^{-\lambda_1 x} + \pi_2 \lambda_2 e^{-\lambda_2 x}}{\pi_1 \lambda_1 + \pi_2 \lambda_2}$$

- La distribution d'une fonction MMPP à quatre états:

$$f(x) = \frac{\pi_1 \lambda_1 e^{-\lambda_1 x} + \pi_2 \lambda_2 e^{-\lambda_2 x} + \pi_3 \lambda_3 e^{-\lambda_3 x} + \pi_4 \lambda_4 e^{-\lambda_4 x}}{\pi_1 \lambda_1 + \pi_2 \lambda_2 + \pi_3 \lambda_3 + \pi_4 \lambda_4}$$

Une estimation simple des paramètres, basée sur les moyennes d'ordre 2, 3, 4…, ne fournit pas une bonne approximation. Nous présenterons ici l'algorithme que nous avons adopté pour déterminer les paramètres de la fonction d'approximation. Pour d'amples détails, l'Annexe 1 : résume les différentes études que nous avons dû effectuer pour aboutir à ce travail, qui nous a requis des connaissances théoriques poussées dans le domaine de la statistique et notamment les techniques de Maximisation de vraisemblance ou les Chaînes de Markov Caché.

L'algorithme proprement dit:

En s'inspirant de **[55]**, la méthode utilisée est l'algorithme EM pour le MLE. L'algorithme se résume ainsi :

π_0 étant l'estimation initiale de la distribution des états.

n le nombre d'échantillons.

Δt_k l'intervalle de temps entre les paquets formant l'échantillon k.

r=2 ou 4, le nombre d'état du MMPP.

La matrice de densité de transition est donnée par:

$$f(x) = \exp\{(Q - \Lambda)x\}\Lambda$$

La matrice de probabilité de transition de la chaîne $\{X_k\}$ est :

$$P = \int_0^{+\infty} f(y)dy = (\Lambda - Q)^{-1}\Lambda$$

Comme dans **[36]**, la fonction $\overline{F}(t)$ définit une matrice r x r. L'élément *(i,j)* de cette fonction est la probabilité que le processus MMPP démarrant de l'état *i* soit à l'état *j* à l'instant t et qu'aucun évènement ne se produise entre les instants 0 et t.

$$\overline{F}(t) = \exp\{(Q - \Lambda)t\}$$

Chaque itération de l'algorithme est comme suit :

(1) Soit $L(0) = \pi^0$, et pour k=1,..,n, soit $L(k)=L(k-1)f(\Delta t_k)$

(2) Soit R(n+1) = **1**, et pour k= n,..,1, soit $R(k)=f(\Delta t_k)R(k+1)$

(3) Pour *i,j* = 1..r, soit $A_{ij} = 0$, $B_i=0$.

(4) Pour k=1,..,n, soit

$$A_{ij} \to A_{ij} + L(k-1)\int_{t_{k-1}}^{t_k} \overline{F}(t-t_{k-1})1_i 1_j^T f(t_k - t)dtR(k+1)$$

(5) Pour k=1,..,n, soit

$$B_i \to B_i + L_i(k)R_i(k+1)$$

(6) Ce qui donne comme estimation :

$$\hat{q}_{ij} = q_{ij}^0 \frac{A_{ij}}{A_{ii}}, i, j = 1..r, i \neq j \text{ et } \hat{\lambda}_i = \frac{B_i}{A_{ii}}, i = 1..r.$$

Cet algorithme itératif est réitéré jusqu'à la convergence des valeurs recherchées.

2.5.2. Estimation Gaussienne

La distribution Gaussienne, connue aussi sous le nom de distribution Normale, s'écrit:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

avec m la moyenne de la distribution et σ^2 la variance.

Ainsi, en trouvant la moyenne et la variance, la fonction de répartition Gaussienne approximant l'échantillon peut être tirée.

2.6. Résultats

Pour aboutir à des résultats fiables et inclus dans l'intervalle de confiance de la méthode et particulièrement pour pallier aux problèmes qui peuvent découler d'une précision finie des nombres utilisés, il faut prendre un nombre assez élevé d'échantillons. Or, ce nombre élevé cause un temps de calcul extrêmement long, qui peut durer plusieurs semaines pour aboutir à la convergence de l'algorithme.

Afin de comparer les différents modèles, nous avons calculé l'erreur quadratique moyenne ε entre les estimations obtenues et la fonction de répartition de l'échantillon étudié:

$$\varepsilon = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - P(x > x_i))^2$$

Plus ε est réduite, meilleure est l'estimation.

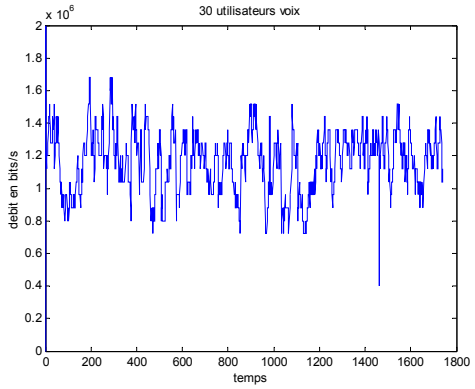
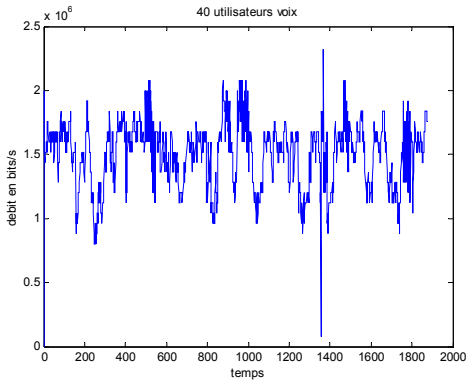
2.6.1. Réseau UMTS

2.6.1.1. Utilisateurs voix

Dans chaque cellule, le nombre de sources voix acceptées par la fonction CAC est fixé. Nous étudions les deux cas: le cas où le canal est moyennement chargé et le cas où le canal est fortement chargé. 30 et 40 utilisateurs voix sont présents respectivement dans chaque cellule.

Aucune fonction de contrôle n'est utilisée dans ce cas et un utilisateur peut envoyer sur un slot chaque fois qu'un talkspurt arrive. C'est le CDMA classique qui donne la meilleure performance [2].

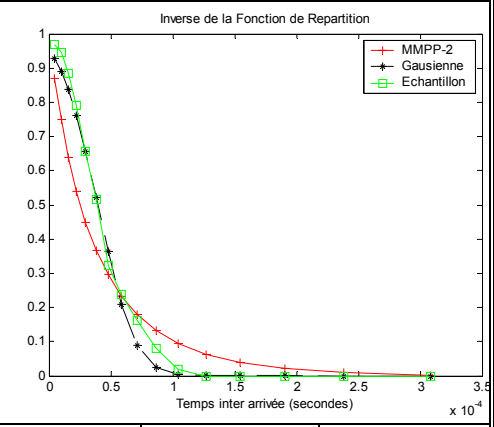
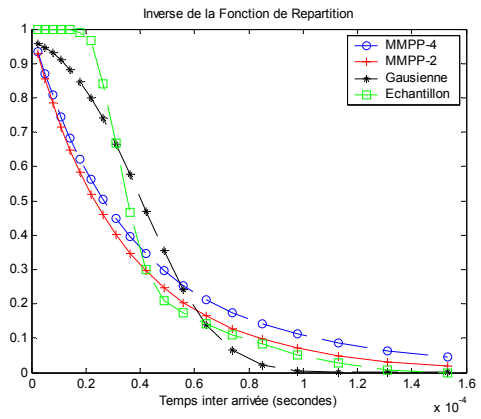
Tableau 13: La variation du débit utile pour des utilisateurs voix.

30 utilisateurs Voix	40 utilisateurs Voix
	
$1{,}167\ 10^6$	$1{,}505\ 10^6$

Dans les deux cas du Tableau 13, le débit varie autour d'une moyenne et sa variabilité ne change pas considérablement entre ces deux configurations.

Les différentes courbes de l'inverse de la fonction de répartition du Tableau 14 arborent une comparaison de l'estimation du trafic des sources décrites précédemment. Au dessous de chaque graphe, l'ensemble des erreurs quadratiques moyennes entre chaque type d'estimation et la distribution des échantillons est noté.

Tableau 14: Les inverses des fonction de répartition et les erreurs quadratiques moyennes des modèles MMPP-2, MMPP-4 et Gaussien pour les utilisateurs voix.

30 utilisateurs Voix			40 utilisateurs Voix		
					
MMPP-4	MMPP-2	Gaussienne	MMPP-4	MMPP-2	Gaussienne
-	0,0159	0,0012	0,0261	0,0313	0,0059

Les estimations basées sur le modèle Gaussien donnent une erreur quadratique inférieure aux deux modèles MMPP utilisés. Cela est plus visible pour le schéma à 30 utilisateurs.

2.6.1.2. Utilisateurs Web

Le système est étudié avec deux charges: une charge considérée moyenne de 500 utilisateurs, et une charge considérée élevée de 700 utilisateurs.

Le débit varie autour de la limite de contrôle. Elle représente la qualité de service qui est de 1440Kb. Nous observons aussi que quand la charge augmente, la variabilité du débit diminue.

Tableau 15: La variation du débit utile pour des utilisateurs Web.

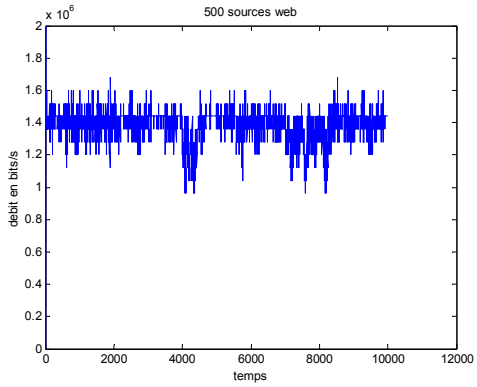
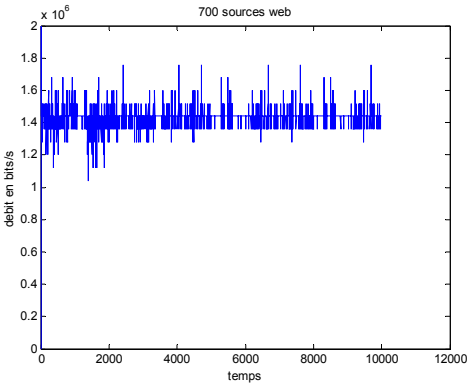
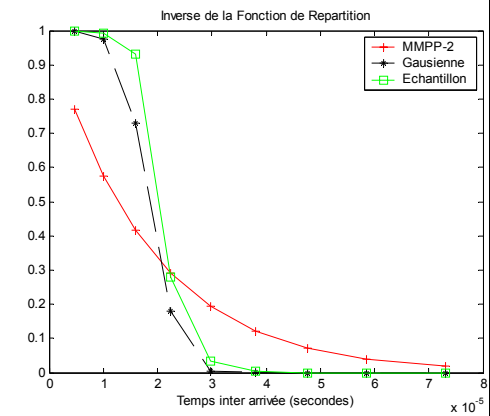
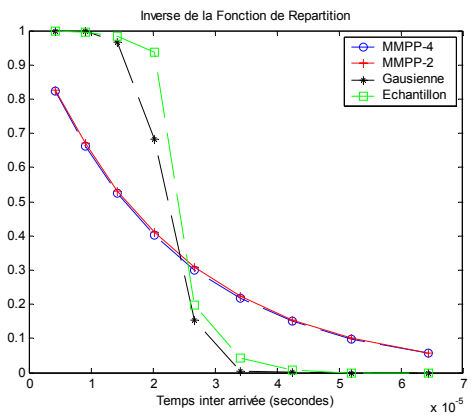
500 utilisateurs Web	700 utilisateurs Web
	
$1{,}388\ 10^6$	$1{,}431\ 10^6$

Tableau 16: Les inverses des fonction de répartition et les erreurs quadratiques moyennes des modèles MMPP-2, MMPP-4 et Gaussien pour les utilisateurs Web.

500 utilisateurs Web			700 utilisateurs Web		
					
MMPP-4	MMPP-2	Gaussienne	MMPP-4	MMPP-2	Gaussienne
-	0,0597	0,0058	0,0795	0,0778	0,0076

Avec les deux cas d'agrégation de flux Web, le modèle Gaussien est mieux adapté pour la modélisation que les modèles MMPP.

2.6.1.3. Utilisateurs FTP

Comme dans les deux cas précédents, deux charges du système sont étudiées qui sont de 50 utilisateurs pour une charge modérée et 70 utilisateurs pour une forte charge.

Avec ce système, la variabilité du débit sortant est faible (c.f. Tableau 17). A forte charge, le débit est presque constant.

Tableau 17: La variation du débit utile pour des utilisateurs FTP.

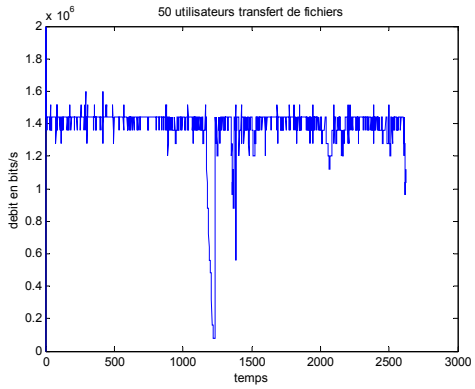
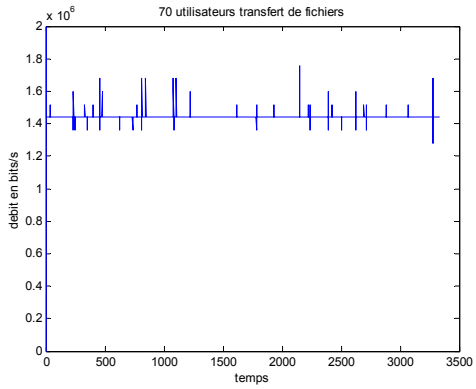
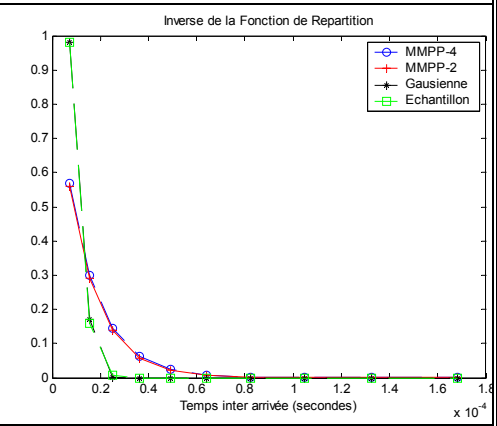
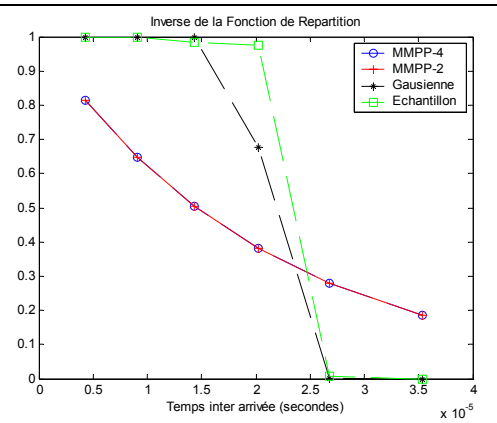
50 utilisateurs FTP	70 utilisateurs FTP
	
$1{,}382\ 10^6$	$1{,}441\ 10^6$

Tableau 18: Les inverses des fonction de répartition et les erreurs quadratiques moyennes des modèles MMPP-2, MMPP-4 et Gaussien pour les utilisateurs FTP.

50 utilisateurs FTP			70 utilisateurs FTP		
					
MMPP-4	MMPP-2	Gaussienne	MMPP-4	MMPP-2	Gaussienne
0,0134	0,0134	$1{,}0963\ 10^{-5}$	0,1421	0,1420	0,0149

D'après le Tableau 18, nous constatons que là aussi, l'approximation Gaussienne est beaucoup mieux adaptée que l'approximation par une MMPP.

2.6.1.4. Multiplexage d'utilisateurs voix Web et FTP

Nous avons étudié deux scénarios: le premier agrégant 15 utilisateurs voix, 150 utilisateurs web et 10 utilisateurs FTP. Le second scénario comporte 15 utilisateurs voix, 150 utilisateurs web et 20 utilisateurs FTP.

Tableau 19: La variation du débit utile pour un multiplexage d'utilisateurs.

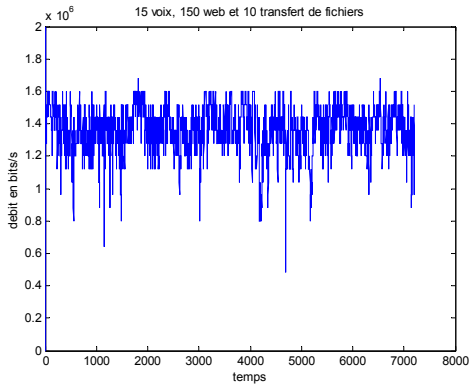
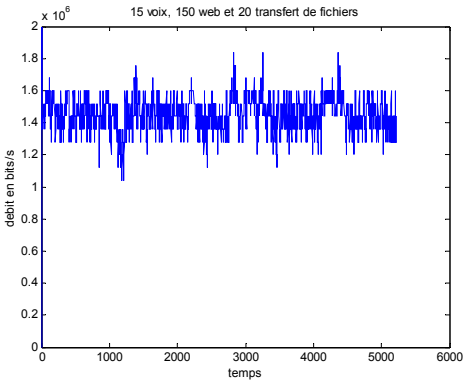
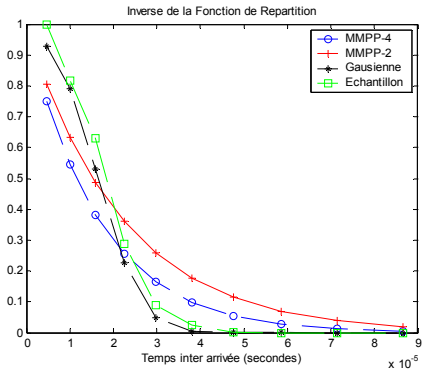
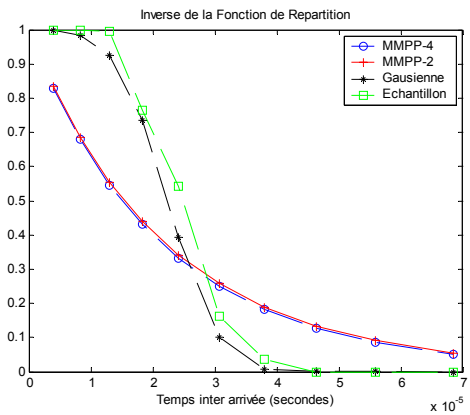
15 voix, 150 web et 10 FTP	15 voix, 150 web et 20 FTP
	
$1{,}37\ 10^6$	$1{,}46\ 10^6$

Tableau 20: Les inverses des fonction de répartition et les erreur quadratique moyenne des modèles MMPP-2, MMPP-4 et Gaussien pour un multiplexage d'utilisateurs.

15 voix, 150 web et 10 FTP			15 voix, 150 web et 20 FTP		
					
MMPP-4	MMPP-2	Gaussienne	MMPP-4	MMPP-2	Gaussienne
0,0143	0,0112	0,0014	0,0547	0,0532	0,0033

Les différentes représentations mais surtout les valeurs des erreurs quadratiques moyennes nous révèlent que le modèle MMPP n'est pas bien adapté à la modélisation. Même en augmentant le nombre d'états du processus, l'estimation n'est pas améliorée. C'est plutôt le modèle Gaussien qui approxime mieux la distribution de l'inter arrivée des paquets. Cela peut

se justifier par le rôle joué par la couche Mac et le protocole CDMA utilisé. Le service RTC utilisé par les connexions voix permettait un taux d'erreur élevé des paquets qui ne sont pas immédiatement acheminés par la couche radio. En plus, le taux d'utilisation de la bande passante disponible est élevé, ce qui suggère que le taux de perte est élevé. Même si cela est vrai, la QoS de chaque appel n'est point détérioré vu que ce type d'application permet des pertes au niveau radio beaucoup plus que les autres types de services. Pour l'agrégation des flux de données, i.e. web et FTP, il suffit de se rappeler que la couche MAC utilisée permettait un séjour des paquets pendant une durée relativement longue, ce qui procure une allure très aplatie de l'agrégat: la probabilité de l'inter arrivée présente un pic à un point donné et diminue très vite de part et d'autre de cette valeur en créant la cloche de Gauss. Ainsi, la représentation par une fonction Gaussienne devient évidente. Cette justification s'applique aussi bien sur les trafics web, FTP qu'au multiplexage des trois services. Dans ce dernier, les flux de données qui peuvent attendre sont majoritaires par rapport aux flux voix.

2.6.2. Réseau 802.11

En utilisant le simulateur de réseaux NS, nous avons procédé à la mise en place de plusieurs scénarios de test. Le Tableau 21 ci-dessous développe les différents paramètres du réseau 802.11 étudié:

Tableau 21: Caractéristiques du réseau 802.11 utilisé.

Taille RTS	44 bits
Taille CTS	38 bits
DIFS	50 µs
SIFS	10 µs
Taille de la trame	20 µs
Bande passante	2 Mb/sec
Fréquence	914 MHz

Les utilisateurs sont répartis sur un cercle autour d'un point d'accès avec lequel ils communiquent tous. Les files d'attente de chaque utilisateur disposent d'un emplacement pour 50 paquets. La politique de service dans toutes les files est FIFO: il n'y a pas de différenciation de traitement ni au niveau de la radio ni au niveau de la couche MAC. Cette stratégie est applicable dans le cas d'un réseau 802.11 de génération actuel. Des terminaux utilisant la norme 802.11e ne sont pas très fréquents à l'heure de la rédaction de ce manuscrit.

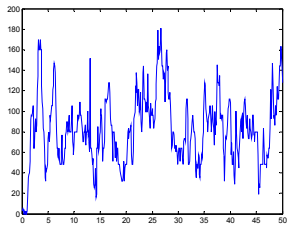
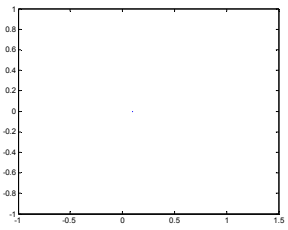
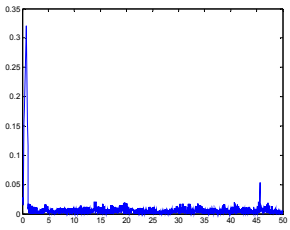
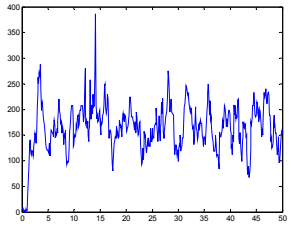
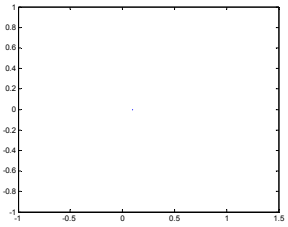
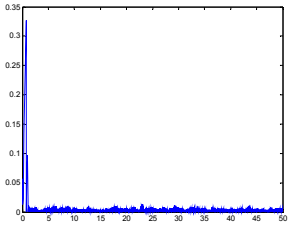
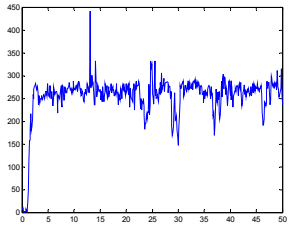
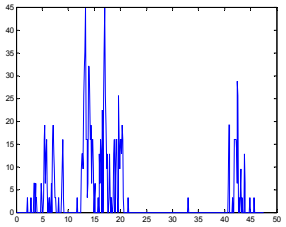
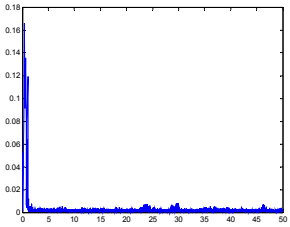
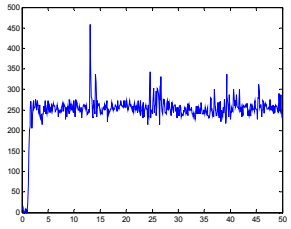
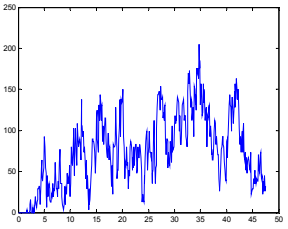
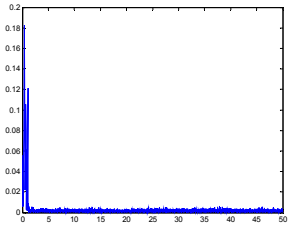
Nous présentons successivement les différents modèles utilisés pour un agrégat d'utilisateurs voix, puis nous passerons à un ensemble de terminaux utilisant une application de type ftp avec TCP comme protocole de transport et enfin, nous étudierons le multiplexage de sources voix et ftp et nous diviserons les flux de sortie en deux agrégations: une qui groupe toutes les connexions voix et donc demandant une certaine QoS dans le cœur de réseau et la deuxième, qui ne requiert pas de traitement spécial au niveau de la QoS, est formée des différents flux FTP présents dans le réseau.

2.6.2.1. Utilisateurs voix

Un ensemble de sources voix émettent leurs paquets vers la source réceptrice. Le Tableau 22 illustre la variation des caractéristiques de la QoS observée sur le canal radio. Au dessous de chaque figure, nous avons introduit la valeur moyenne. Nous observons que plus

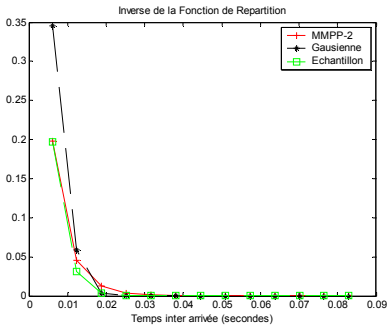
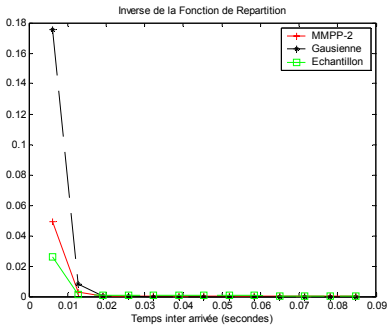
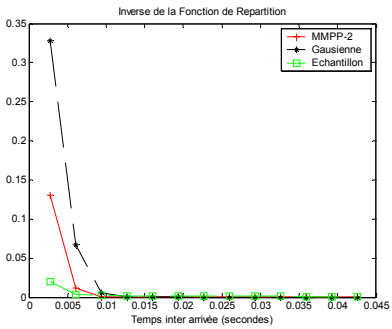
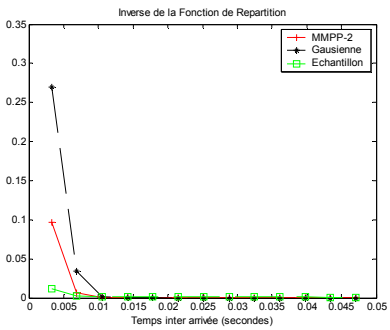
le nombre de sources voix augmente, plus la variabilité du débit de sortie diminue. De plus, arrivé à la valeur de 40 utilisateurs, le système ne garantit plus de QoS aux utilisateurs: le taux de perte moyen qui est de 1,6% pour 40 utilisateurs devient 24% pour 50 utilisateurs. De plus, le débit utile en sortie n'augmente plus, ce qui correspond à dire que les utilisateurs s'interfèrent mutuellement.

Tableau 22: La variation des caractéristiques et leurs valeurs moyennes au niveau de la couche MAC d'une base 802.11 pour des connexions voix.

	La variation du débit utile	La variation de la perte	La variation de l'inter arrivée
15	 83,1507	 0	 0,0039
25	 164,1254	 0	 0,0020
40	 254,7117	 0,0160	 0,0013
50	 249,2794	 0,2428	 0,0013

Les estimations par les modèles MMPP 2 et Gaussien sont dévoilées au Tableau 23. L'erreur quadratique moyenne entre la distribution des échantillons et celle estimée par chaque méthode, est notée sous chaque courbe. Nous constatons que, contrairement à l'UTRAN, la modélisation par un processus MMPP a été respectée comme la prédisait les différents travaux présentés précédemment dans ce chapitre. Même pour le cas de 50 utilisateurs, qui présente un taux de perte élevé, la précision de l'estimation reste très acceptable.

Tableau 23: Les inverses des fonctions de répartition et les erreurs quadratiques moyennes des modèles MMPP-2 et Gaussien de l'inter arrivée des paquets au niveau de la couche MAC d'une base 802.11 pour des connexions voix.

15 sources voix		
	MMPP-2	Gaussienne
	$6{,}58\ 10^{-6}$	$4{,}59\ 10^{-4}$
25 sources voix		
	MMPP-2	Gaussienne
	$1{,}12\ 10^{-5}$	$4{,}56\ 10^{-4}$
40 sources voix		
	MMPP-2	Gaussienne
	$2{,}51\ 10^{-4}$	0,0020
50 sources voix		
	MMPP-2	Gaussienne
	$1{,}49\ 10^{-4}$	0,0014

2.6.2.2. Utilisateurs FTP

Les utilisateurs ftp dans ce cas sont contrôlés uniquement par le protocole TCP: tant que ce dernier leur permet d'émettre des paquets, ils demandent l'autorisation à la couche MAC. Ceci peut être observé sur les différentes figures du Tableau 24 ci-dessous: le débit moyen est toujours élevé quelque soit le nombre de sources ftp existantes dans le réseau. De plus, la valeur moyenne diminue pour un nombre élevé vu qu'une partie de la bande est perdue pour cause d'une probabilité de contention plus élevée qui est directement liée au taux de perte sur l'interface radio. La variation de ce dernier est directement liée avec l'évolution de la fenêtre de congestion de TCP. La valeur de cette fenêtre est contrôlée par les mécanismes de *slow start* et de congestion *avoidnance* de TCP.

Tableau 24: La variation des caractéristiques et leurs valeurs moyennes au niveau de la couche MAC d'une base 802.11 pour des connexions FTP.

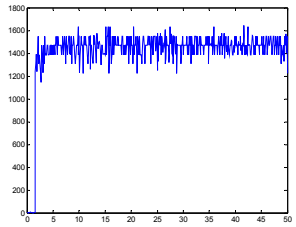
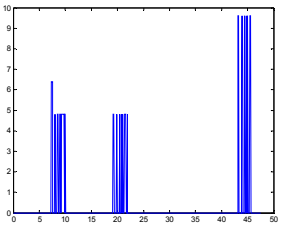
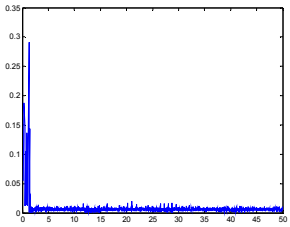
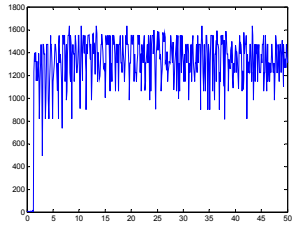
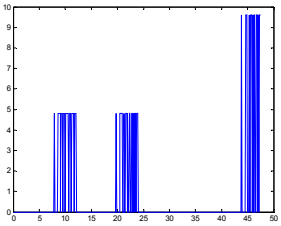
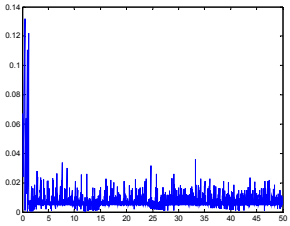
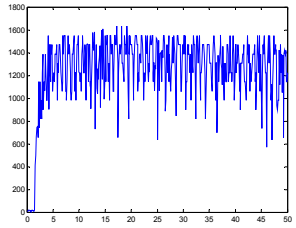
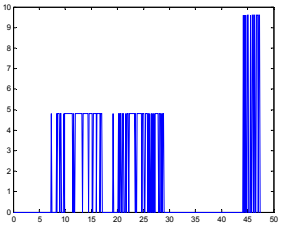
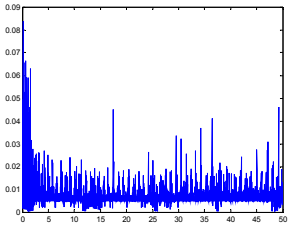
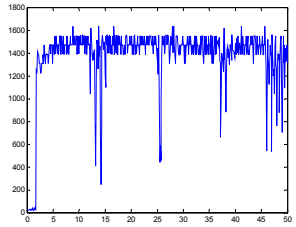
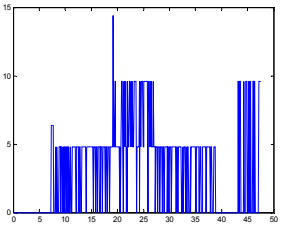
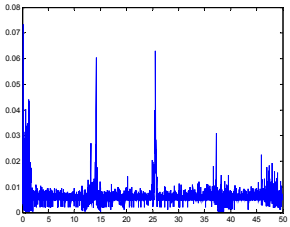
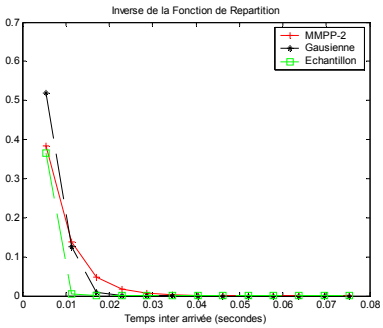
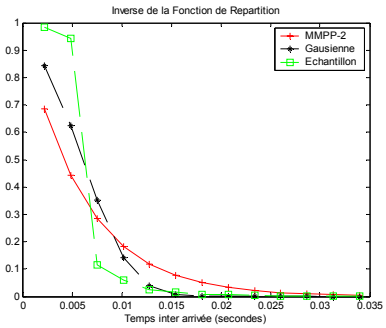
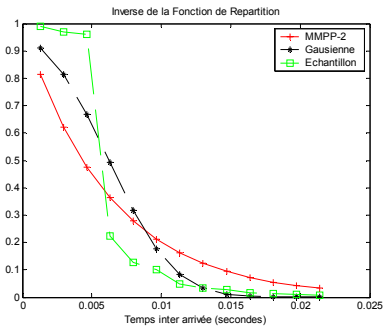
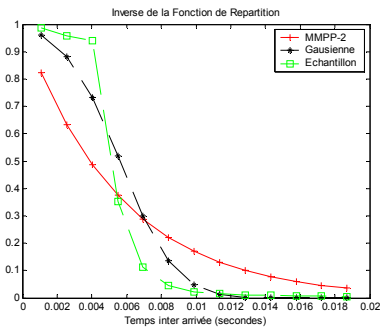
	La variation du débit utile	La variation de la perte	La variation de l'inter arrivée
30	 1413,9	 0,0061	 0,0057
50	 1308,9	 0,0118	 0,0061
100	 1252,9	 0,0280	 0,0062
200	 1369,7	 0,0463	 0,0056

Tableau 25: Les inverses des fonctions de répartition et les erreurs quadratiques moyennes des modèles MMPP-2 et Gaussien de l'inter arrivée des paquets au niveau de la couche MAC d'une base 802.11 pour des connexions FTP.

30 sources FTP		
	MMPP-2	Gaussienne
	$4{,}10\ 10^{-4}$	$7{,}77\ 10^{-4}$
50 sources FTP		
	MMPP-2	Gaussienne
	0,0082	0,0038
100 sources FTP		
	MMPP-2	Gaussienne
	0,0097	0,0047
200 sources FTP		
	MMPP-2	Gaussienne
	0,0093	0,0025

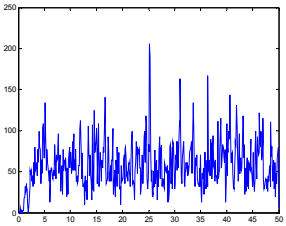
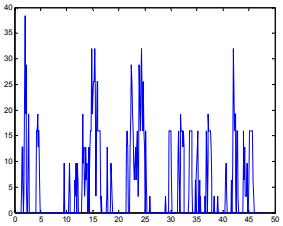
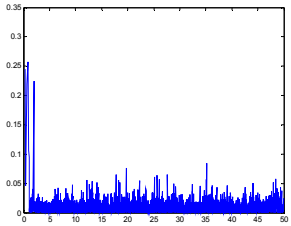
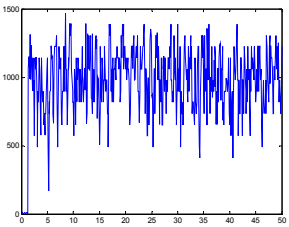
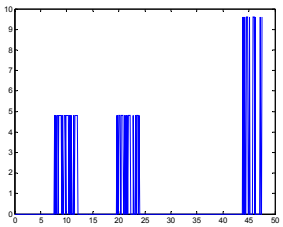
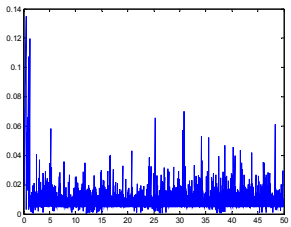
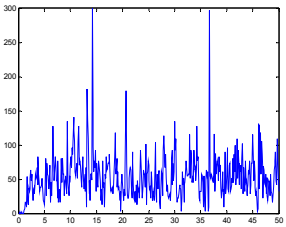
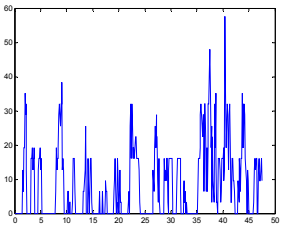
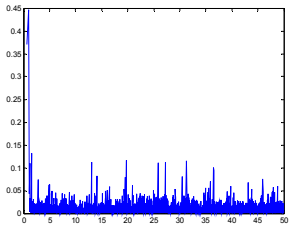
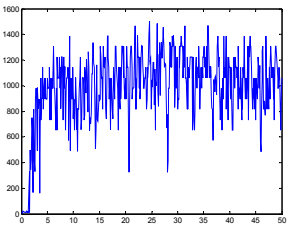
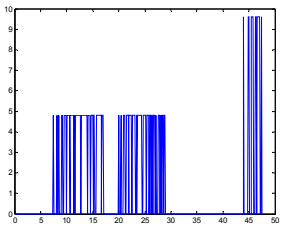
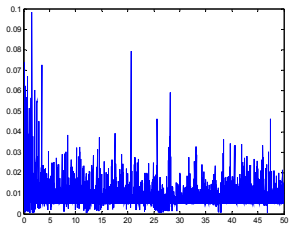
Les différentes figures comprises au Tableau 25 nous montre qu'une modélisation gaussienne est mieux adaptée à ce type de flux qu'une modélisation MMPP.

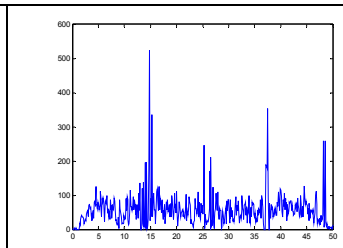
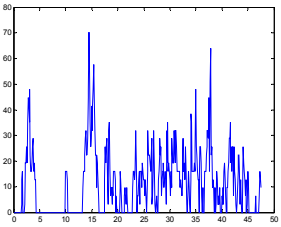
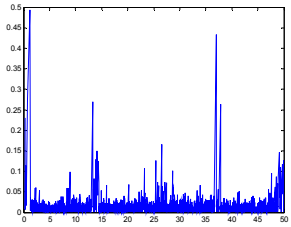
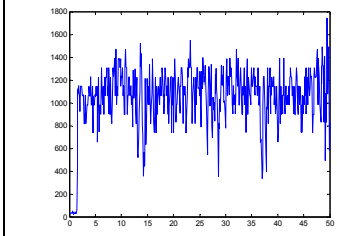
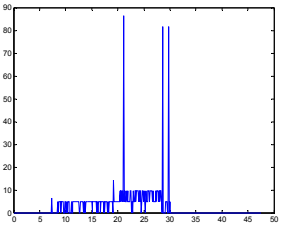
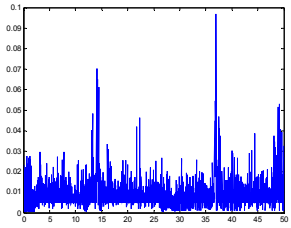
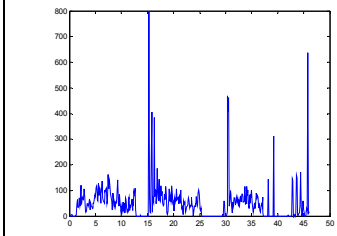
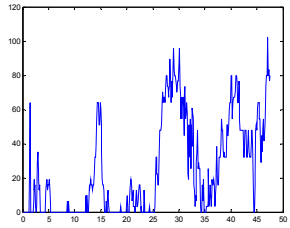
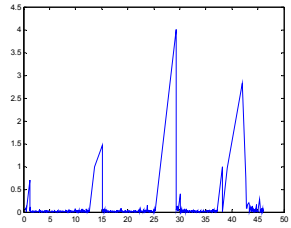
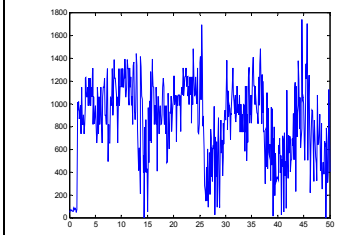
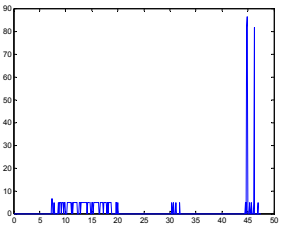
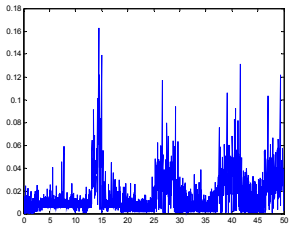
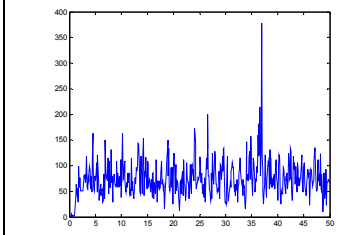
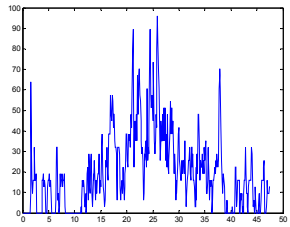
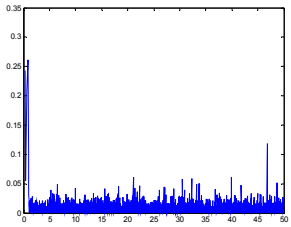
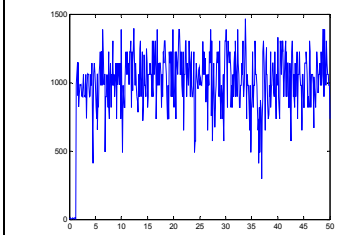
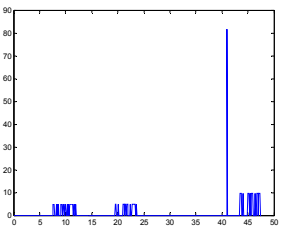
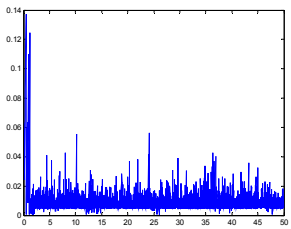
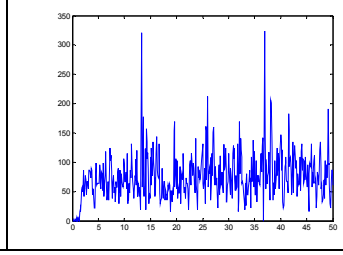
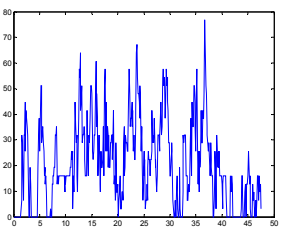
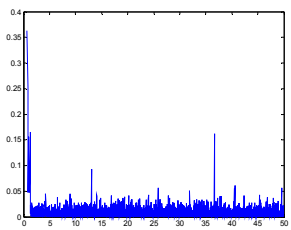
2.6.2.3. Multiplexage d'utilisateurs voix et FTP

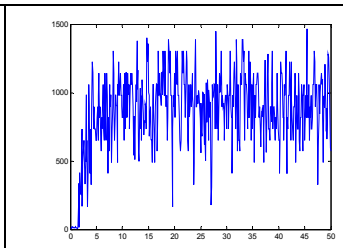
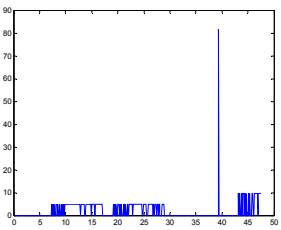
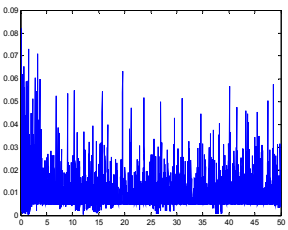
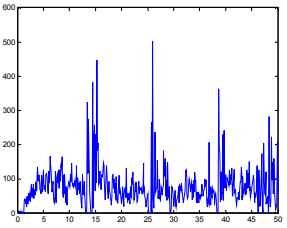
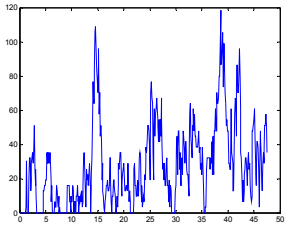
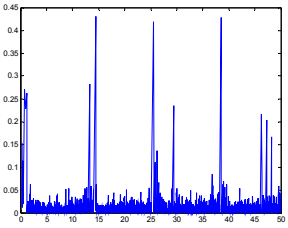
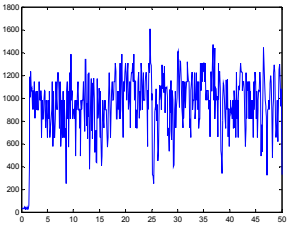
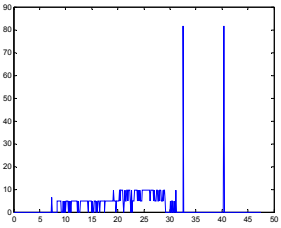
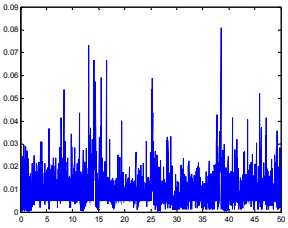
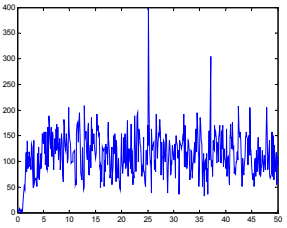
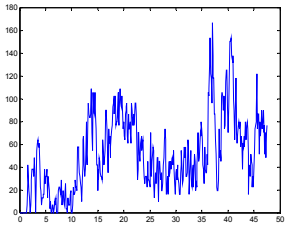
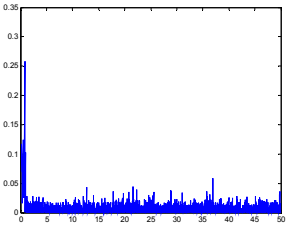
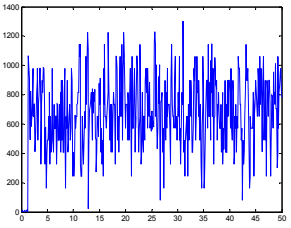
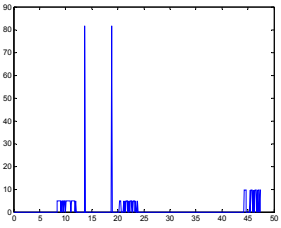
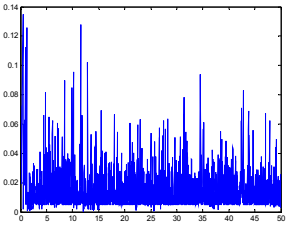
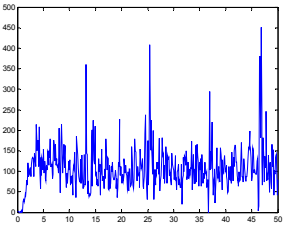
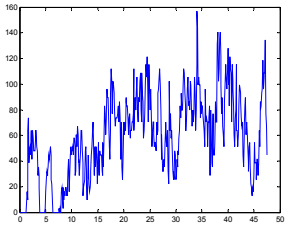
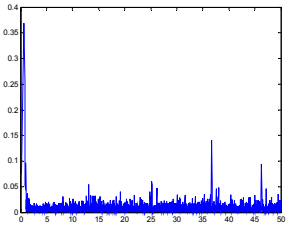
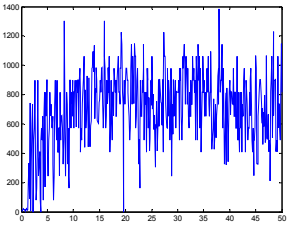
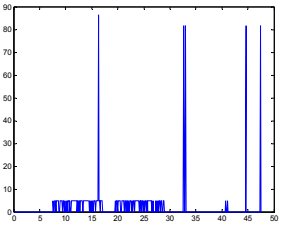
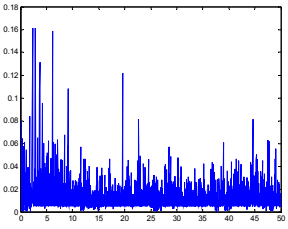
En multiplexant les deux types de sources, nous tirons les figures du Tableau 26. Le nombre d'utilisateurs de voix et de données est présenté à la première colonne du tableau. Nous avons adopté ces valeurs suite aux différentes expérimentations conclues précédemment. Ces figures nous permettent de signaler la croissance de la variabilité du débit utile dans toutes les situations: les sources voix On/Off ont provoqué une croissance du taux de perte observé sur les sources FTP gérées par TCP.

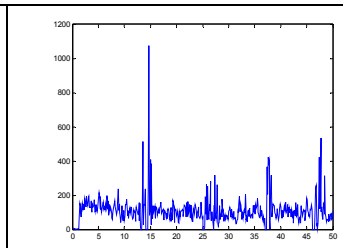
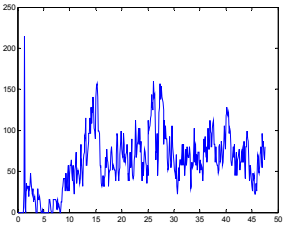
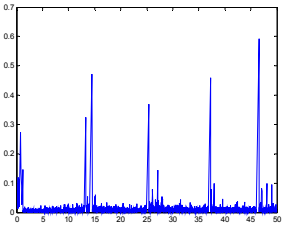
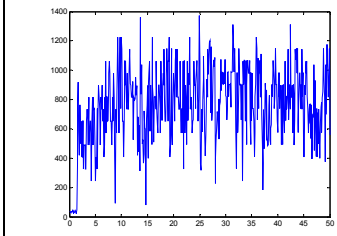
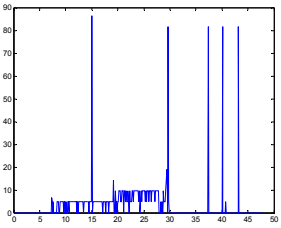
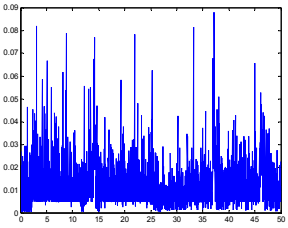
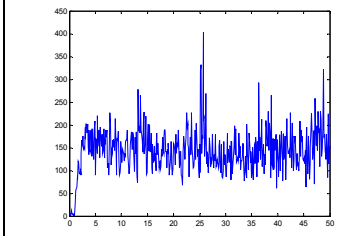
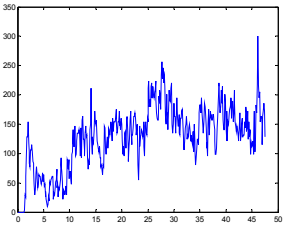
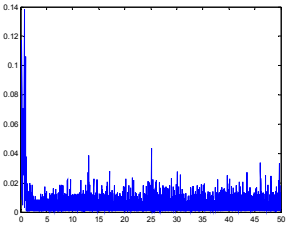
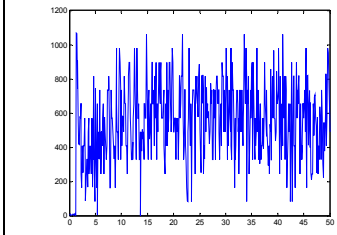
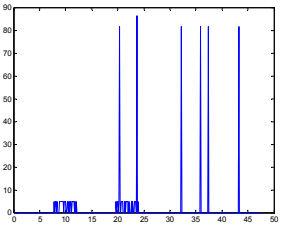
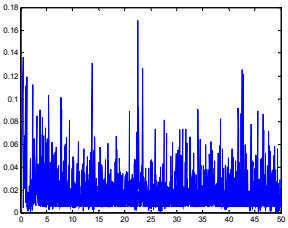
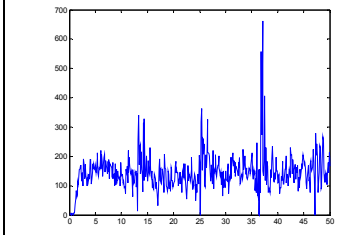
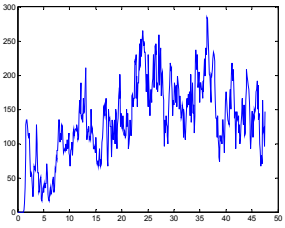
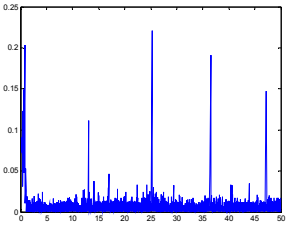
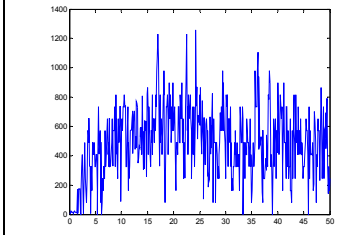
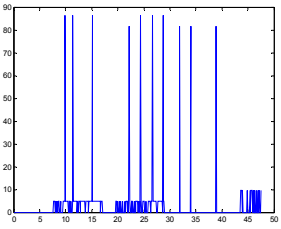
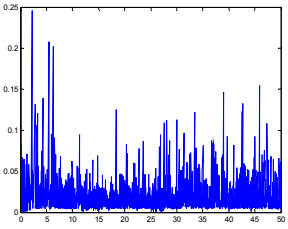
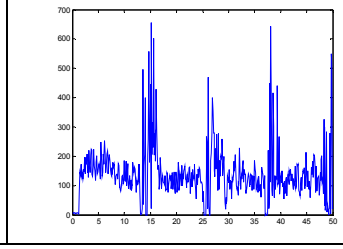
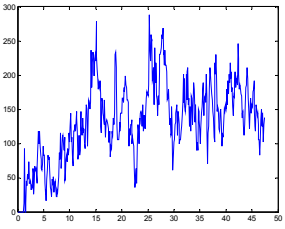
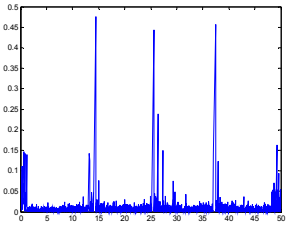
Pour une meilleure visibilité, nous avons regroupé les différentes valeurs moyennes dans les figures du Tableau 27.

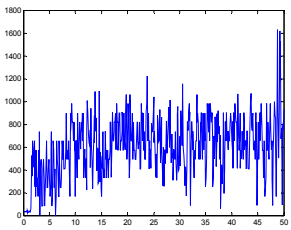
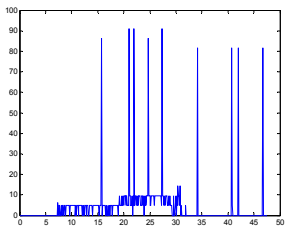
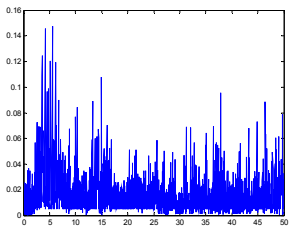
Tableau 26: La variation des caractéristiques au niveau de la couche MAC d'une base 802.11 pour un multiplexage de connexions voix et FTP.

		La variation du débit utile	La variation de la perte	La variation de l'inter arrivée
10 voix 50 FTP	Voix			
	FTP			
10 voix 100 FTP	Voix			
	FTP			

10 voix 250 FTP	Voix			
	FTP			
10 voix 500 FTP	Voix			
	FTP			
15 voix 50 FTP	Voix			
	FTP			
15 voix 100 FTP	Voix			

	FTP			
15 voix 250 FTP	Voix			
	FTP			
25 voix 50 FTP	Voix			
	FTP			
25 voix 100 FTP	Voix			
	FTP			

25 voix 250 FTP	Voix			
	FTP			
40 voix 50 FTP	Voix			
	FTP			
40 voix 100 FTP	Voix			
	FTP			
40 voix 250 FTP	Voix			

	FTP			

Dans le cas de 10 utilisateurs voix et 500 utilisateurs FTP, nous observons des valeurs grandes de l'inter arrivée des paquets voix. Ces derniers sont obligés d'attendre la libération du canal des paquets FTP avant d'être émis. Or les valeurs d'attentes sont très longues pour le service voix: elle peuvent atteindre jusqu'à 4 secondes. Ce qui n'est pas acceptable pour les utilisateurs. C'est pour cela que nous n'avons pas étudié ce cas et nous ne l'avons pas réitéré pour un nombre supérieur d'usagers voix.

En augmentant le nombre d'utilisateurs voix, la valeur moyenne du débit utile de leur agrégation augmente. Contrairement au cas des sources FTP: comme ces sources sont uniquement contrôlées par TCP, la quantité d'information qu'elles émettent est égale à la bande passante disponible. Plus les sources voix sont nombreuses, plus grand est leur débit total et moins grande est la bande passante restante à FTP. Le taux de perte est directement proportionnel au nombre de sources émettrices. Nous constatons néanmoins que le nombre de sources FTP n'influe pas le taux de perte des sources voix. Par contre, plus est grand le nombre des sources voix, plus le taux de perte des flux FTP est élevé. Quant à l'inter arrivée moyenne, elle est inversement proportionnelle au débit utile.

Tableau 27: Représentation graphique des valeurs moyennes des caractéristiques au niveau de la couche MAC d'une base 802.11 pour un multiplexage de connexions voix et FTP.

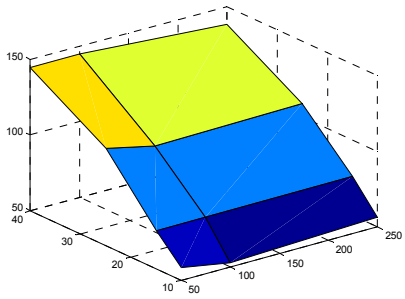
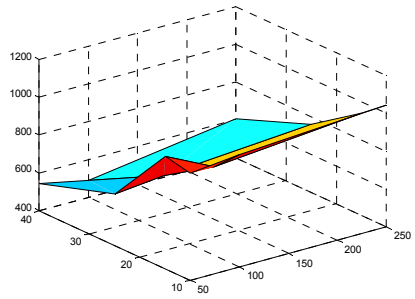
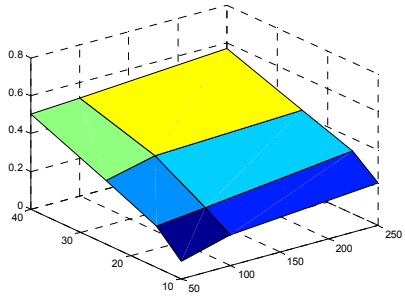
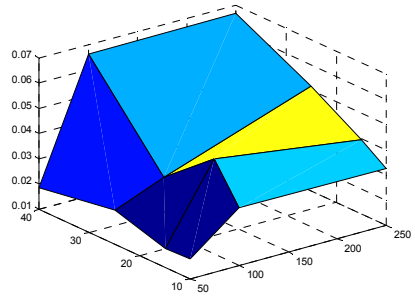
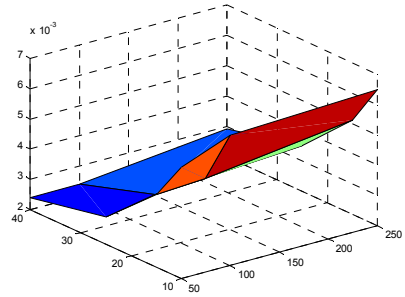
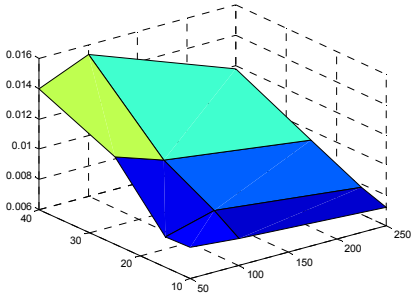
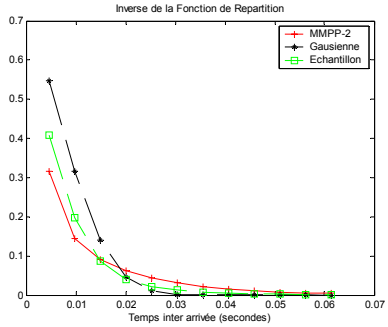
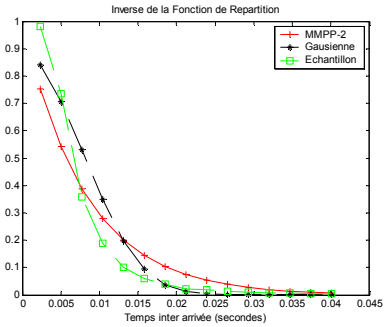
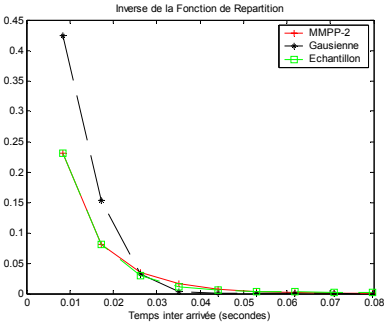
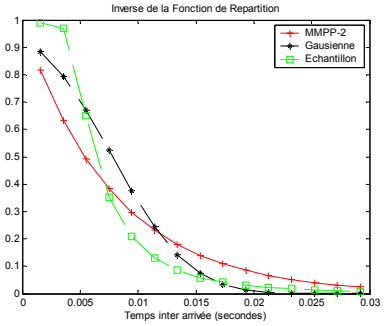
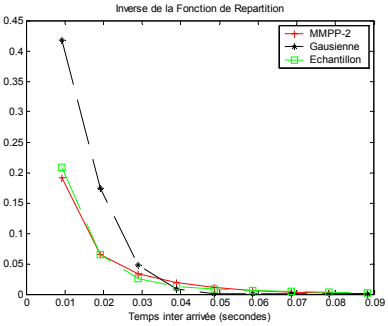
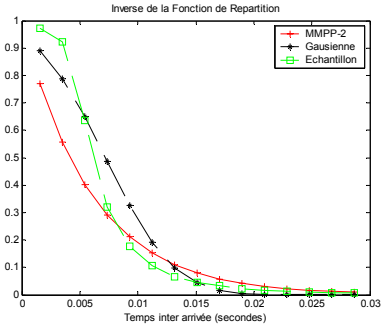
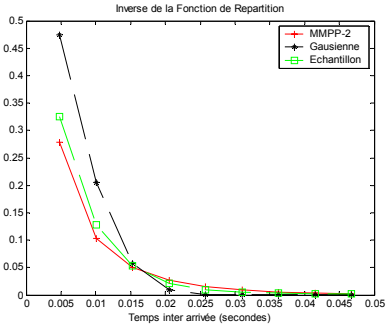
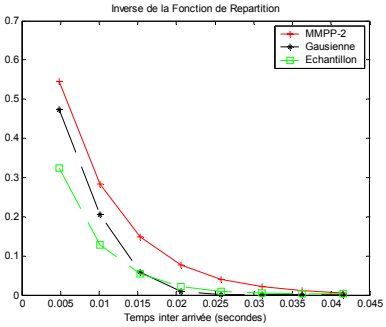
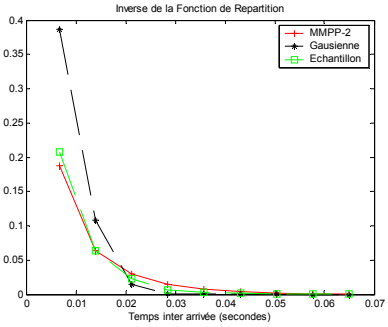
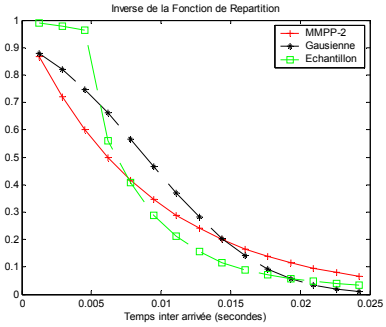
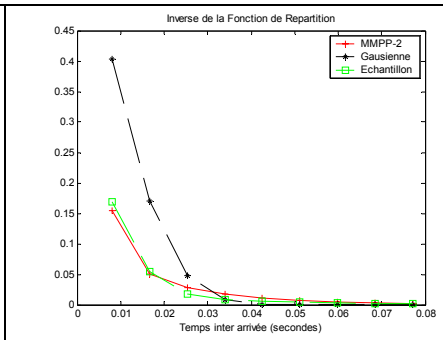
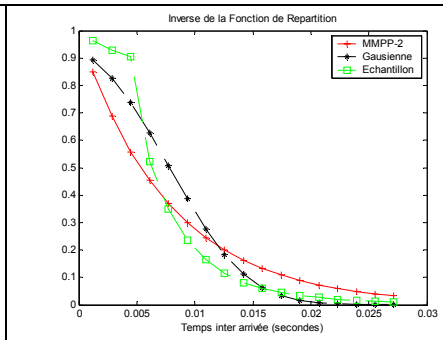
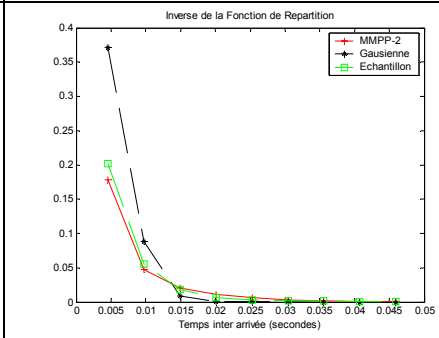
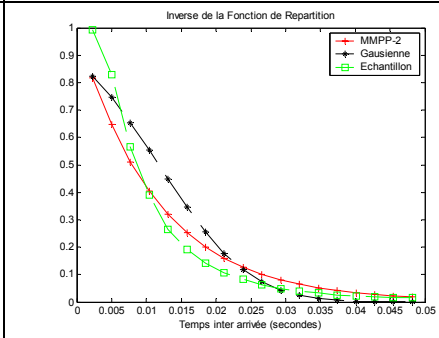
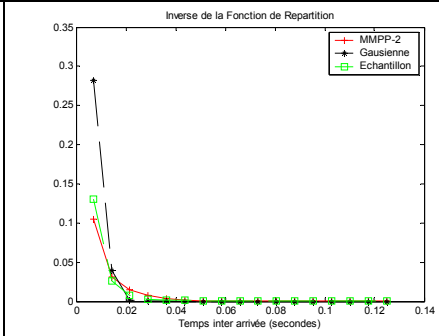
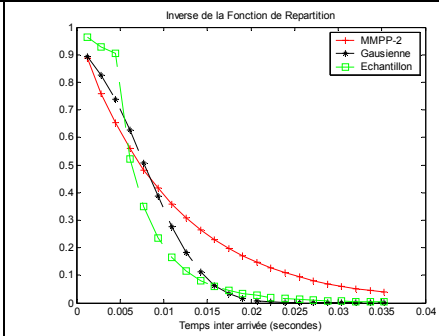
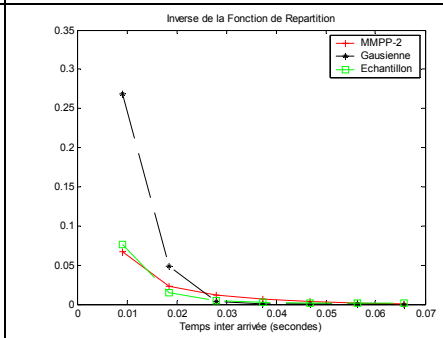
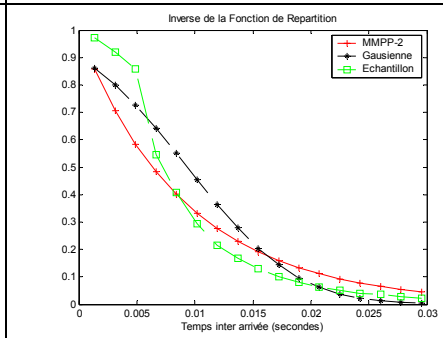
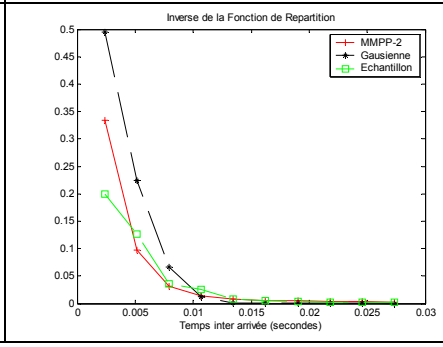
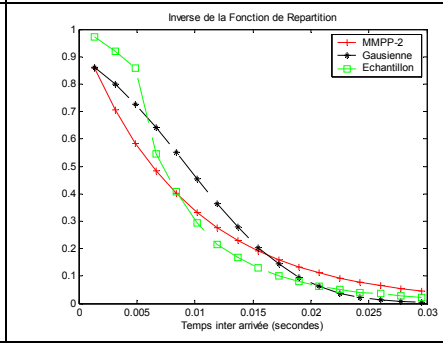
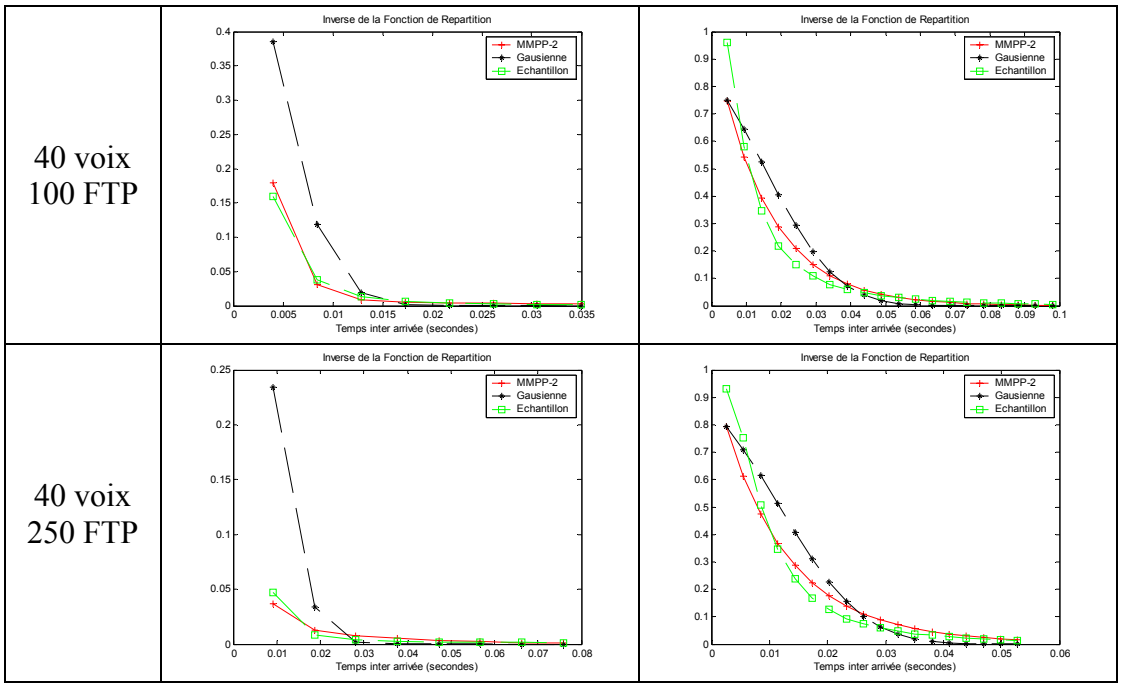
	Sources Voix	Sources FTP
Le débit utile (kb/sec)		
Le taux de perte		
L'inter arrivée moyenne (sec)		

Tableau 28: Résultats d'estimations avec un modèle suivant une distribution MMPP-2 et Gaussienne de l'inter arrivée des paquets au niveau de la couche MAC d'une base 802.11 pour un multiplexage de connexions voix et FTP.

	Les paquets voix	Les paquets FTP
10 voix 50 FTP		
10 voix 100 FTP		
10 voix 250 FTP		
15 voix 50 FTP		
15 voix 100 FTP		

15 voix 250 FTP		
25 voix 50 FTP		
25 voix 100 FTP		
25 voix 250 FTP		
40 voix 50 FTP		



Même avec un multiplexage de source, l'agrégat des flux de voix garde son caractère Markovien: il se rapproche mieux d'une source MMPP qu'une source Gaussienne. Comme nous pouvons le voir dans le Tableau 29, la modélisation de l'agrégation des sources FTP peut être mieux approximée par la loi Gaussienne pour un faible nombre de sources voix. Néanmoins, cette affirmation n'est plus valable quand les sources voix croissent. L'erreur quadratique du modèle MMPP devient plus petite que celle du modèle Gaussien.

Tableau 29: Erreur quadratique moyenne pour un multiplexage de connexions voix et FTP.

	Les sources voix		Les sources FTP	
	MMPP-2	Gaussienne	MMPP-2	Gaussienne
10 voix 50 FTP	$2{,}6205\ 10^{-4}$	$7{,}6740\ 10^{-4}$	0,0025	0,0018
10 voix 100 FTP	$1{,}4703\ 10^{-6}$	$8{,}8010\ 10^{-4}$	0,0044	0,0024
10 voix 250 FTP	$8{,}2375\ 10^{-6}$	0,0012	0,0048	0,0017
15 voix 50 FTP	$5{,}9773\ 10^{-5}$	$5{,}8329\ 10^{-4}$	0,0018	$5{,}8329\ 10^{-4}$
15 voix 100 FTP	$1{,}1065\ 10^{-5}$	$6{,}9393\ 10^{-4}$	0,0053	0,0042
15 voix 250 FTP	$9{,}5893\ 10^{-6}$	0,0014	$7{,}9467\ 10^{-4}$	0,0014
25 voix 50 FTP	$1{,}3595\ 10^{-5}$	$6{,}1114\ 10^{-4}$	0,0017	0,0030
25 voix 100 FTP	$1{,}4997\ 10^{-5}$	$4{,}7531\ 10^{-4}$	0,0079	0,0025
25 voix 250 FTP	$4{,}6031\ 10^{-6}$	$7{,}7926\ 10^{-4}$	0,0034	0,0030
40 voix 50 FTP	$3{,}8800\ 10^{-4}$	0,0020	0,0034	0,0030
40 voix 100 FTP	$1{,}0158\ 10^{-5}$	0,0012	0,0013	0,0030
40 voix 250 FTP	$3{,}4425\ 10^{-6}$	$7{,}2826\ 10^{-4}$	0,0011	0,0026

2.7. Conclusion

Dans ce chapitre, nous avons caractérisé le processus d'inter arrivée de l'agrégation de plusieurs flux sortants de deux réseaux sans fil UTRAN et 802.11. A partir de fichiers traces

générées par simulations, nous avons essayé de modéliser l'inter arrivée des paquets par un processus MMPP à 2 phases puis à 4 phases. Nous avons ensuite comparé notre échantillon à une distribution suivant la loi Gaussienne.

Approximer un processus par la loi Gaussienne est simple: il suffit de trouver la moyenne et la variance de l'échantillon étudié. Par contre, afin de trouver les paramètres d'une distribution MMPP, nous avons dû joindre deux méthodes proposées par d'autres travaux afin de trouver une solution rapide et fiable à notre interrogation.

A partir des plusieurs cas présentés et étudiés, nous pouvons déduire que le modèle Gaussien d'inter arrivée des paquets est très bien adapté pour les flux voix, Web, FTP et le multiplexage de ces trois flux entrants au réseau UTRAN utilisant la couche MAC CDMA illustrée.

Par contre, le modèle de sortie d'un réseau 802.11 correspond, pour l'agrégation de flux voix à une source MMPP à 2 états. Ce résultat confirme bien la théorie qui approxime une superposition de flux On/Off par une source MMPP à 2 états. De plus, plus le nombre d'utilisateurs augmente, plus l'approximation est meilleure. Cette constatation reste tout à fait valable avec le multiplexage de sources: même en présence de sources FTP, un modèle MMPP modélise toujours bien l'agrégation de sources voix.

La modélisation des sources FTP est moins sûre: la différence de l'erreur quadratique moyenne entre la distribution de l'échantillon et des distributions des probabilités de la source MMPP 2 ou Gaussienne qui approximent le mieux l'échantillon, n'est pas tranchante. Les sources FTP seules sont mieux modélisées par la loi Gaussienne mais quand il y a un multiplexage de sources voix et de sources FTP, la loi Gaussienne est uniquement meilleure pour un nombre réduit d'utilisateurs de voix. Si le nombre de ces derniers augmente, c'est un modèle MMPP 2 qui devient plus adapté à la modélisation.

CHAPITRE 3
Amélioration des performances par la différenciation des classes

Nous étudions dans ce chapitre plusieurs mécanismes de QoS existants dans un réseau IP. Comme nous l'avions évoqué dans le Chapitre 1, la combinaison des protocoles DiffServ et MPLS rend possible une plus grande maîtrise des performances de bout en bout du réseau. Comme DiffServ propose l'utilisation de 3 classes de services EF, AF et BE, à chaque fois que nous utilisons trois classes de services, nous référons implicitement à ce modèle. Les paquets de la classe 1 doivent subir des délais stricts. Les flux AF, accepteront un peu de perte et ne seront pas pénalisés dans le cas d'un délai pas très réduit. Quant à la classe BE, elle n'admet pas de contraintes.

Décomposé en cinq parties, ce chapitre propose en premier lieu un modèle analytique du service de l'ordonnanceur GPS à 3 classes. Les résultats intermédiaires sont inclus dans l'Annexe 3 :. Ensuite, dans le paragraphe suivant, plusieurs courbes illustrent la répartition statistique des délais moyens et des taux de pertes de l'ordonnanceur WRR avec deux distributions possibles de la taille des paquets: la distribution exponentielle et la distribution de pareto. Les figures de l'Annexe 4 : complètent les résultats inclus dans cette partie. Le paragraphe 4 présente les règles à tirer des différentes études présentées. Dans le cinquième paragraphe, nous étudions l'effet de l'utilisation des différents ordonnanceurs dans un réseau entier. Plusieurs schémas de mesures complémentaires qui corroborent aux différentes conclusions élaborées sont présentés à l'Annexe 5 :. Nous terminons le chapitre par les différentes conclusions que nous pouvons tirer.

3.1. Modèle analytique pour GPS avec trois classes de service

A partir des différentes définitions d'un ordonnanceur GPS, que nous avons indiqué au Chapitre 1, nous présentons ici les différentes équations qui régissent le modèle analytique de cet ordonnanceur admettant 3 classes de services.

Soit μ_0, le rapport de la taille moyenne d'un paquet sur la vitesse du lien. C'est le taux de service moyen du système s'il n'y avait pas de différenciation entre les trois classes. Soit ω_1, ω_2, et ω_3, les poids respectifs des classes 1, 2 et 3.

Nous définissons les paramètres $\mu_1 = \frac{\varpi_1}{\varpi_1 + \varpi_2 + \varpi_3}\mu_0$, $\mu_2 = \frac{\varpi_2}{\varpi_1 + \varpi_2 + \varpi_3}\mu_0$ et $\mu_3 = \frac{\varpi_3}{\varpi_1 + \varpi_2 + \varpi_3}\mu_0$ comme les portions du service alloué aux différentes classes quand des clients de chacune des trois classes sont présents dans la file. Le Tableau 30 ci-dessous résume les valeurs des taux de service quand uniquement une ou deux classes existent.

Tableau 30: Les taux de service des classes en présence des autres classes.

	Classe 1	Classe 2	Classe 3
Classe 1	μ_0	$\mu_{21} = \frac{\varpi_2}{\varpi_1 + \varpi_2}\mu_0$	$\mu_{31} = \frac{\varpi_3}{\varpi_1 + \varpi_3}\mu_0$
Classe 2	$\mu_{12} = \frac{\varpi_1}{\varpi_1 + \varpi_2}\mu_0$	μ_0	$\mu_{32} = \frac{\varpi_3}{\varpi_2 + \varpi_3}\mu_0$
Classe 3	$\mu_{13} = \frac{\varpi_1}{\varpi_1 + \varpi_3}\mu_0$	$\mu_{23} = \frac{\varpi_2}{\varpi_2 + \varpi_3}\mu_0$	μ_0

La Figure 23 représente schématiquement les différentes probabilités sur les axes du parallélogramme formé par les longueurs des files.

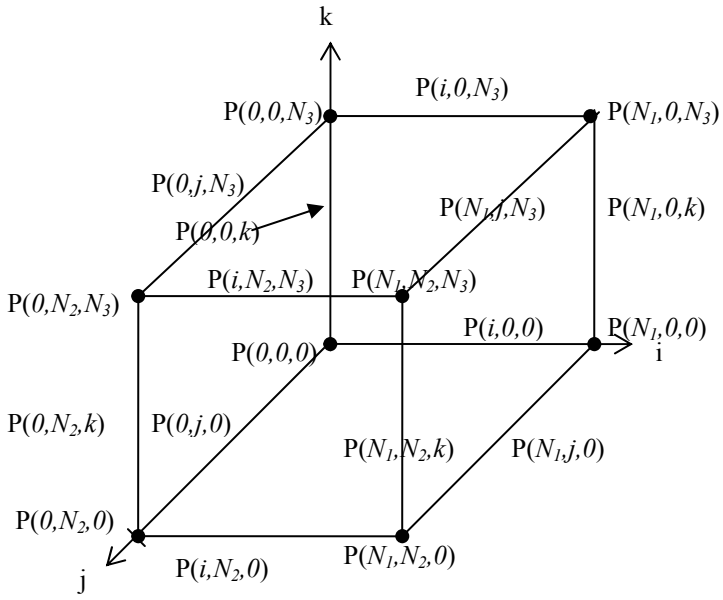


Figure 23: Les différentes probabilités sur les axes du parallélogramme.

Le modèle de file d'attente est donc tridimensionnel. Nous noterons N_1, N_2 et N_3 les longueurs maximales respectives des files des classes 1, 2 et 3. La chaîne incluse du modèle M/M/1 avec des valeurs éloignées des extrémités suit la Figure 24 (dans le "volume" du parallélogramme).

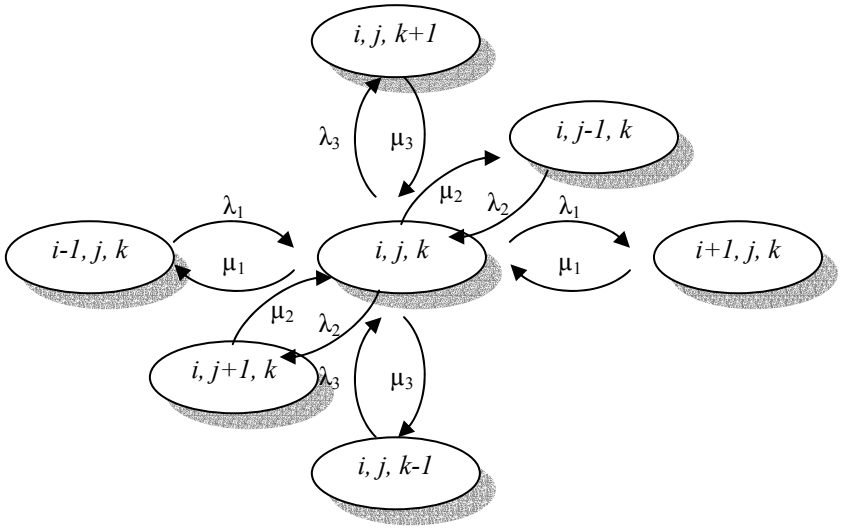


Figure 24: La chaîne incluse dans le cas général de trois classes.

Ainsi, l'équation d'équilibre autour de *P(i,j,k)* qui en découle s'écrit:

$$(\lambda_1+\lambda_2+\lambda_3+\mu_1+\mu_2+\mu_3)P(i,j,k)=\mu_1P(i+1,j,k)+\lambda_1P(i-1,j,k)+\mu_2P(i,j+1,k)+\lambda_2P(i,j-1,k)$$
$$+\mu_3P(i,j,k+1)+\lambda_3P(i,j,k-1)$$

Le Tableau 31 développe les équations autour des différents points, axes et plans particuliers. Les représentations des chaînes incluses sont exposées à l'Annexe 4 :.

Bien que les équations semblent simples, malheureusement, ces différentes équations ne sont pas tractables. Même avec des valeurs de N_1, N_2 et N_3 réduites, les combinaisons énormes des 13 différents paramètres rendent les solutions intermédiaires extrêmement volumineuses à écrire (c.f. Annexe 3 :). Ce qui nous a empêché d'obtenir une solution analytique symbolique est la capacité "limitée" des ordinateurs à notre disposition. En utilisant Maple® sur une station Sun™ pourvu de 1Goctet de mémoire vive, le système se trouve vite saturé par le nombre d'éléments à traiter dès qu'au moins une des valeurs de N_1, N_2 et N_3 dépasse l'unité.

Dans le paragraphe 3.2 suivant, nous présentons les différents résultats obtenus par simulation.

Tableau 31: Les 26 différentes équations autour des points, des droites et des plans particuliers.

Classe			Equations d'équilibre
1	2	3	
0	*j*	*k*	$(\lambda_1+\lambda_2+\lambda_3+\mu_{23}+\mu_{32})P(0,j,k)=\mu_1P(1,j,k)+\mu_{23}P(0,j+1,k)+\lambda_2P(0,j-1,k)+\mu_{32}P(0,j,k+1)+\lambda_3P(0,j,k-1)$
N_1	*j*	*k*	$(\lambda_2+\lambda_3+\mu_1+\mu_2+\mu_3)P(N_1,j,k)=\lambda_1P(N_1-1,j,k)+\mu_2P(N_1,j+1,k)+\lambda_2P(N_1,j-1,k)+\mu_3P(N_1,j,k+1)+\lambda_3P(N_1,j,k-1)$
i	*0*	*k*	$(\lambda_1+\lambda_2+\lambda_3+\mu_{13}+\mu_{31})P(i,0,k)=\mu_{13}P(i+1,0,k)+\lambda_1P(i-1,0,k)+\mu_2P(i,1,k)+\mu_{31}P(i,0,k+1)+\lambda_3P(i,0,k-1)$
i	N_2	*k*	$(\lambda_1+\lambda_3+\mu_1+\mu_2+\mu_3)P(i,N_2,k)=\mu_1P(i+1,N_2,k)+\lambda_1P(i-1,N_2,k)+\lambda_2P(i,N_2-1,k)+\mu_3P(i,N_2,k+1)+\lambda_3P(i,N_2,k-1)$
i	*j*	*0*	$(\lambda_1+\lambda_2+\lambda_3+\mu_{12}+\mu_{21})P(i,j,0)=\mu_{12}P(i+1,j,0)+\lambda_1P(i-1,j,0)+\mu_{21}P(i,j+1,0)+\lambda_2P(i,j-1,0)+\mu_3P(i,j,1)$
i	*j*	N_3	$(\lambda_1+\lambda_2+\mu_1+\mu_2+\mu_3)P(i,j,N_3)=\mu_1P(i+1,j,N_3)+\lambda_1P(i-1,j,N_3)+\mu_2P(i,j+1,N_3)+\lambda_2P(i,j-1,N_3)+\lambda_3P(i,j,N_3-1)$
0	*0*	*k*	$(\lambda_1+\lambda_2+\lambda_3+\mu_0)P(0,0,k)=\mu_{13}P(1,0,k)+\mu_{23}P(0,1,k)+\mu_0P(0,0,k+1)+\lambda_3P(0,0,k-1)$
0	N_2	*k*	$(\lambda_1+\lambda_3+\mu_{23}+\mu_{32})P(0,N_2,k)=\mu_1P(1,N_2,k)+\lambda_2P(0,N_2-1,k)+\mu_{32}P(0,N_2,k+1)+\lambda_3P(0,N_2,k-1)$
N_1	*0*	*k*	$(\lambda_2+\lambda_3+\mu_{13}+\mu_{31})P(N_1,0,k)=\lambda_1P(N_1-1,0,k)+\mu_2P(N_1,1,k)+\mu_{31}P(N_1,0,k+1)+\lambda_3P(N_1,0,k-1)$
N_1	N_2	*k*	$(\lambda_3+\mu_1+\mu_2+\mu_3)P(N_1,N_2,k)=\lambda_1P(N_1-1,N_2,k)+\lambda_2P(N_1,N_2-1,k)+\mu_3P(N_1,N_2,k+1)+\lambda_3P(N_1,N_2,k-1)$
0	*j*	*0*	$(\lambda_1+\lambda_2+\lambda_3+\mu_0)P(0,j,0)=\mu_{12}P(1,j,0)+\mu_0P(0,j+1,0)+\lambda_2P(0,j-1,0)+\mu_{32}P(0,j,1)$
0	*j*	N_3	$(\lambda_1+\lambda_2+\mu_{23}+\mu_{32})P(0,j,N_3)=\mu_1P(1,j,N_3)+\mu_{23}P(0,j+1,N_3)+\lambda_2P(0,j-1,N_3)+\lambda_3P(0,j,N_3-1)$
N_1	*j*	*0*	$(\lambda_2+\lambda_3+\mu_{12}+\mu_{21})P(N_1,j,0)=\lambda_1P(N_1-1,j,0)+\mu_{21}P(N_1,j+1,0)+\lambda_2P(N_1,j-1,0)+\mu_3P(N_1,j,1)$
N_1	*j*	N_3	$(\lambda_2+\mu_1+\mu_2+\mu_3)P(N_1,j,N_3)=\lambda_1P(N_1-1,j,N_3)+\mu_2P(N_1,j+1,N_3)+\lambda_2P(N_1,j-1,N_3)+\lambda_3P(N_1,j,N_3-1)$
i	*0*	*0*	$(\lambda_1+\lambda_2+\lambda_3+\mu_0)P(i,0,0)=\mu_0P(i+1,0,0)+\lambda_1P(i-1,0,0)+\mu_{21}P(i,1,0)+\mu_{31}P(i,0,1)$
i	N_2	*0*	$(\lambda_1+\lambda_3+\mu_{12}+\mu_{21})P(i,N_2,0)=\mu_{12}P(i+1,N_2,0)+\lambda_1P(i-1,N_2,0)+\lambda_2P(i,N_2-1,0)+\mu_3P(i,N_2,1)$
i	*0*	N_3	$(\lambda_1+\lambda_2+\mu_{13}+\mu_{31})P(i,0,N_3)=\mu_{13}P(i+1,0,N_3)+\lambda_1P(i-1,0,N_3)+\mu_2P(i,1,k)+\lambda_3P(i,0,N_3-1)$
i	N_2	N_3	$(\lambda_1+\mu_1+\mu_2+\mu_3)P(i,N_2,N_3)=\mu_1P(i+1,N_2,N_3)+\lambda_1P(i-1,N_2,N_3)+\lambda_2P(i,N_2-1,N_3)+\lambda_3P(i,N_2,N_3-1)$
0	*0*	*0*	$(\lambda_1+\lambda_2+\lambda_3)P(0,0,0)=\mu_0P(1,0,0)+\mu_0P(0,1,0)+\mu_0P(0,0,1)$
0	*0*	N_3	$(\lambda_1+\lambda_2+\mu_0)P(0,0,N_3)=\mu_{13}P(1,0,N_3)+\mu_{23}P(0,1,N_3)+\lambda_3P(0,0,N_3-1)$
N_1	*0*	*0*	$(\lambda_2+\lambda_3+\mu_0)P(N_1,0,0)=\lambda_1P(N_1-1,0,0)+\mu_{21}P(N_1,1,0)+\mu_{31}P(N_1,0,1)$
N_1	*0*	N_3	$(\lambda_2+\mu_{13}+\mu_{31})P(N_1,0,N_3)=\lambda_1P(N_1-1,0,N_3)+\mu_2P(N_1,1,N_3)+\lambda_3P(N_1,0,N_3-1)$
0	N_2	*0*	$(\lambda_1+\lambda_3+\mu_0)P(0,N_2,0)=\mu_{12}P(1,N_2,0)+\lambda_2P(0,N_2-1,0)+\mu_{32}P(0,N_2,1)$
0	N_2	N_3	$(\lambda_1+\mu_{23}+\mu_{32})P(0,N_2,N_3)=\mu_1P(1,N_2,N_3)+\lambda_2P(0,N_2-1,N_3)+\lambda_3P(0,N_2,N_3-1)$
N_1	N_2	*0*	$(\lambda_3+\mu_{12}+\mu_{21})P(N_1,N_2,0)=\lambda_1P(N_1-1,N_2,0)+\lambda_2P(N_1,N_2-1,0)+\mu_3P(N_1,N_2,1)$
N_1	N_2	N_3	$(\mu_1+\mu_2+\mu_3)P(N_1,N_2,N_3)=\lambda_1P(N_1-1,N_2,N_3)+\lambda_2P(N_1,N_2-1,N_3)+\lambda_3P(N_1,N_2,N_3-1)$

3.2. Résultats pour trois classes de service

Comme nous l'avons indiqué précédemment, nous approximerons le modèle GPS par le modèle WRR car ces deux algorithmes admettent les mêmes propriétés moyennes. Vu que le modèle analytique ne peut pas donner de résultats, nous avons utilisé NS pour simuler une file implémentant WRR dans le but de pouvoir tirer quelques règles de dimensionnement pour les réseaux. Le schéma architectural de base d'un ordonnanceur dans un routeur implémentant trois classes de services suit la Figure 25 où trois classes de service sont émises sur un même lien. Les différents paquets sont ainsi multiplexés sur ce lien commun. Le rôle du *classifieur* est de classifier chaque paquet dans la file d'attente correspondante. Dans le modèle DiffServ, le classifieur utilise le champ DSCP afin d'accomplir son rôle. L'ordonnanceur distribue le service aux classes de trafic en fonction des poids alloués à chaque file.

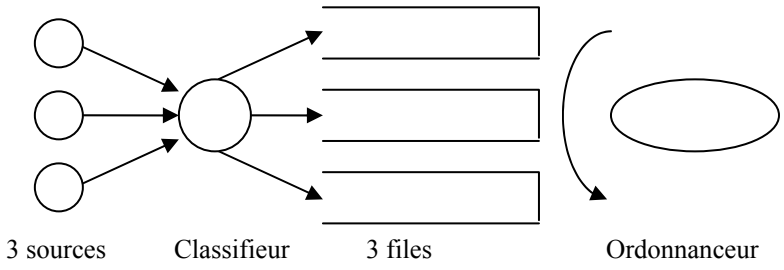


Figure 25: Schéma d'un ordonnanceur de 3 classes

Afin de systématiser le travail et les interprétations, nous avons fixé deux conditions sur le choix des poids dans notre étude:

- Nous avons gardé la somme des poids alloués aux 3 classes égale à 10.
- Nous avons toujours $\omega_1 \geq \omega_2 \geq \omega_3$ de telle sorte que la classe 1 soit toujours mieux servie que la classe 2, qui quant à elle, est mieux servie que la classe 3.

Avec ces deux conditions, nous utiliserons les valeurs citées dans le Tableau 32 comme poids utilisés des trois classes de service dans les simulations ultérieures.

Tableau 32: Les différents poids utilisés pour les trois classes.

ω_1	ω_2	ω_3
4	3	3
4	4	2
5	4	1
5	3	2
6	3	1
6	2	2
7	2	1
8	1	1

Le réseau a été chargé de cinq charges totales possibles. Les valeurs de celles-ci sont résumées dans le Tableau 33. Nous entendons par "charge totale" la charge introduite par la somme des trois classes présentes dans le réseau. Une valeur supérieure à 1 n'implique pas forcement une instabilité totale des trois classes dans le réseau. Par contre, comme nous

l'observerons plus tard, c'est au moins une des classes qui sera dans un état instable et par suite présentera de grands taux de pertes.

Tableau 33: Les charges totales utilisées.

Charge totale	0,2	0,5	0,8	1,0	1,2

Dans les différentes courbes des délais observées sur le lien, il faudra retrancher la valeur du temps de transit sur le lien. Cette valeur est de 10 msec dans tous les cas présentés dans ce chapitre et dans l'Annexe 4 :.

Pour simplifier l'étude, nous avons pris la même valeur moyenne de la taille des paquets pour les trois classes de service. C'est pourquoi, la différenciation de la charge introduite par chaque classe et par conséquent le taux d'arrivée, est fonction du débit des flux sources. Afin de donner un caractère général aux valeurs, nous avons modulé la charge du réseau par un taux relatif τ à chaque classe. Chacune des 36 colonnes du Tableau 34 sera multipliée par les différentes charges totales du réseau afin d'avoir la charge individuelle introduite par chaque classe.

Tableau 34: Les taux relatifs des trois classes.

τ_1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	4	4	5	5	5	5	6	6	6	7	7	8
τ_2	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	1	2	3	4	5	6	1	2	3	4	5	1	2	3	4	1	2	3	1	2	1
τ_3	8	7	6	5	4	3	2	1	7	6	5	4	3	2	1	6	5	4	3	2	1	5	4	3	2	1	4	3	2	1	3	2	1	2	1	1

Enfin, pour simplifier les différentes interprétations des résultats, nous avons considéré qu'aucun mécanisme de retransmission ni de contrôle d'erreur ou de congestion n'existait dans le réseau. Deux constats sont donc à noter: d'une part, un paquet perdu ne sera jamais retransmis ni par la couche réseau ni par la couche applicative, d'autre part, les temps d'émission des paquets des flux au niveau source ne seront pas affectés si le paquet précédent est perdu dans le réseau.

3.2.1. M/M/1 avec trois classes de service

En premier lieu, nous avons étudié les performances d'un ordonnanceur M/M/1: l'arrivée des paquets suit un processus de Poisson et la distribution de la taille des paquets suit une loi exponentielle. Les trois flux des trois différentes classes de service admettent les mêmes moyennes des tailles des paquets. Ainsi, c'est un multiplexage de trois flux homogènes.

Les différents tableaux de figure qui suivent représentent la variation du délai moyen des trois classes en fonction du débit introduit par les classes 1 et 2 (respectivement axe à gauche et axe à droite du point d'origine (0,0,0)). Pour mieux comparer les courbes, nous avons représenté sur une même ligne des tableaux la variation du délai avec la même charge totale introduite par les trois classes. Comme les axes des abscisses et des ordonnées sont relatifs aux débits des classes respectifs 1 et 2, la valeur du débit de la classe 3 se fait en retranchant du débit total écoulé sur le lien (obtenu grâce à la charge totale transitant sur le lien) les deux débits introduits par les deux classes. Nous illustrons ci-dessous ces variations en fonction de trois pondérations possibles: une pondération de 8/1/1 très favorable à la classe

1, ensuite une pondération de 6/3/1 qui donne un avantage de la classe 1 à la classe 2 et de la classe 2 à la classe 3, et enfin, une pondération de 4/4/2 qui différencie très peu les trois classes. L'évolution du délai moyen avec d'autres pondérations possibles est mise dans l'Annexe 4 :.

Le Tableau 35 présente les variations du délai moyen en fonction de la charge du lien pour une combinaison des poids de 8/1/1. Cette configuration permet d'écouler davantage de paquets de classe 1 que des deux autres classes. Facilement observable sur ces courbes, le délai de la classe 1 reste réduit jusqu'à ce que la portion du trafic de cette classe atteigne 80% avec une charge totale du réseau égale ou supérieure à 1. Dans ces deux derniers cas, le délai augmente fortement. Il faut noter que dans le cas d'un réseau fortement chargé (charge totale supérieure ou égale à 1), et tant que la portion du flux de la classe 1 ne dépasse pas son poids associé dans l'ordonnanceur, le délai moyen des flux reste réduit. La variation du délai moyen des classes 2 et 3 devra être identique vu que le poids relatif à chacune des classes est le même. Or, ce n'est pas le cas. Cela est dû au fait que dans l'implémentation de l'ordonnanceur, la file 2 est toujours servie avant celle de 3 en cas de présence de paquets dans les deux files. Or, comme le poids des deux classes est faible, et quand des flux des deux classes existent, la probabilité d'avoir des paquets dans les deux files est élevée. A faible charge, le délai de la classe 2 est directement fonction de la charge introduite par cette même classe: nous observons des pics de délai pour des débits élevés des flux sources de la classe 2. La classe 2 ne peut pas prendre la totalité de la bande disponible pour écouler ses flux vu qu'elle a un poids égal à la classe 3: la bande passante non utilisée par la classe 1 est équitablement répartie pour les deux classes 2 et 3. Cependant, ces pics deviennent faibles devant les délais observés à forte charge. Dans ce dernier cas, les ressources sont prises par les flux de la classe 1.

Avec le poids 6/3/1 (Tableau 36), les flux 1 admettent toujours des délais moyens réduits. Un petit surcroît est constaté sur la valeur du délai par rapport au cas précédent (poids 8/1/1). Pour la classe 2, c'est toujours la même constatation: à faible charge, le délai moyen augmente en fonction de la charge introduite par la classe 2. A forte charge, c'est le poids supérieur alloué à la classe 1 qui tend à surélever le délai sur le lien pour la classe 2. La classe 3 subit des délais supérieurs à la classe 2. De plus, les flux de la classe 1 induisent des délais plus élevés sur les flux 3 que les flux issus des flux de la classe 2.

La pondération des flux par une combinaison de 4/4/2 (Tableau 37) permet de traiter les flux 1 et 2 de la même manière: les mesures des délais observés par ces deux flux sont relativement identiques. Les courbes dévoilent des valeurs très proches les unes des autres mais surtout, des allures de courbe étroitement liées. Les flux de la classe 3 sont, quant à eux, mieux traités que dans les deux cas précédents (poids 1 dans les deux cas précédents). Cependant, ils sont moins bien traités que les flux 1 et 2. Ces deux derniers flux exercent le même effet sur les flux de la classe 3 vu que le poids associé est le même.

Tableau 35: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1.

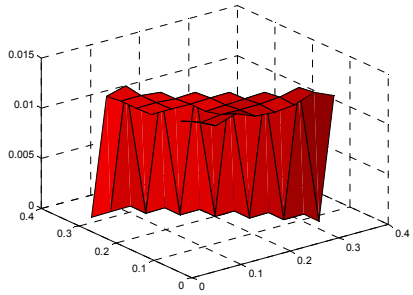
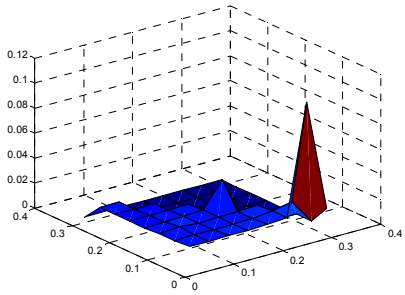
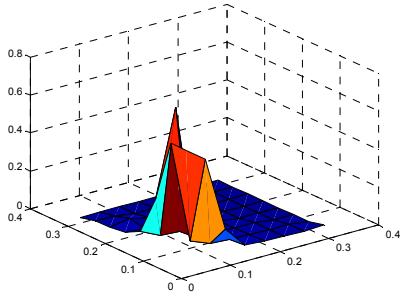
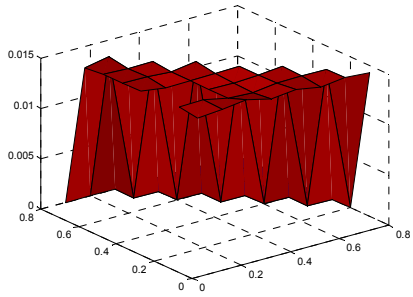
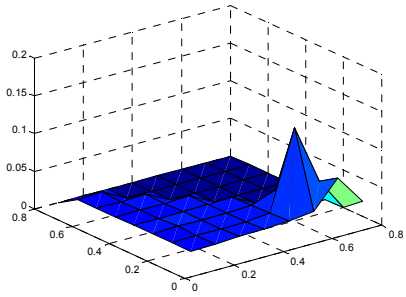
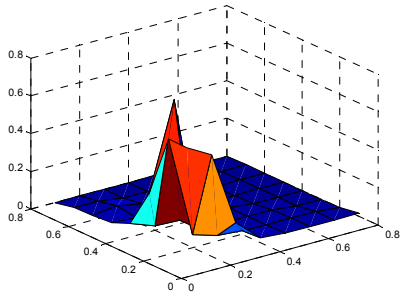
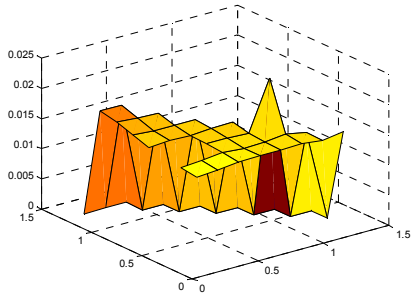
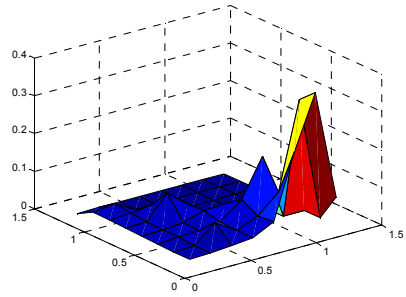
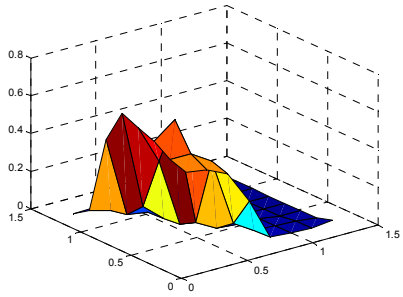
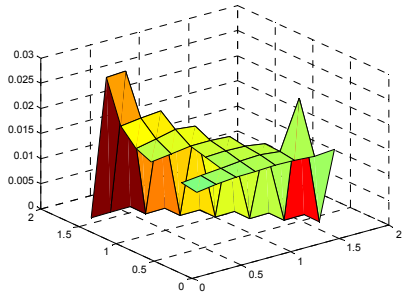
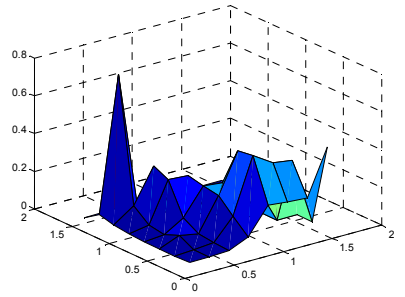
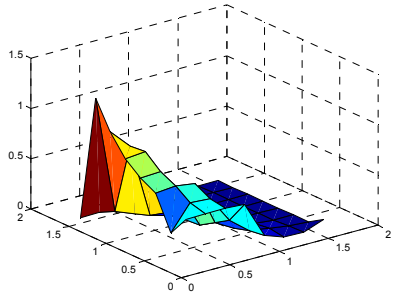
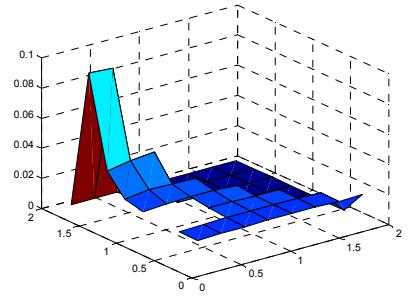
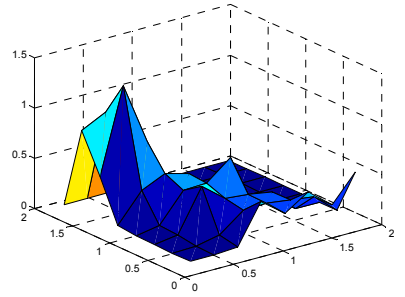
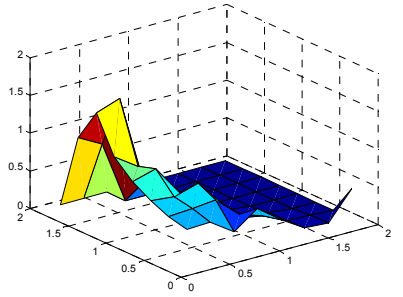
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 36: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1.

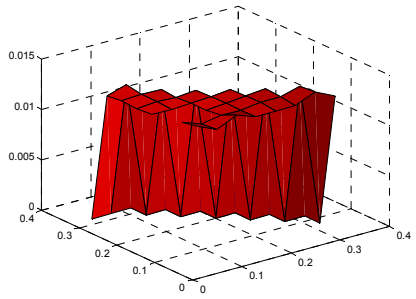
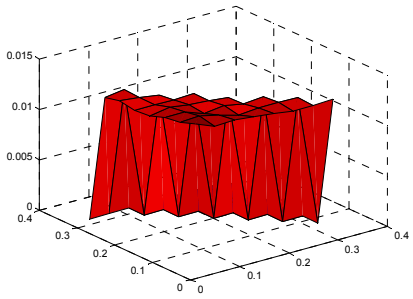
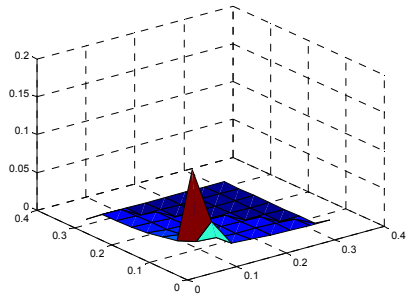
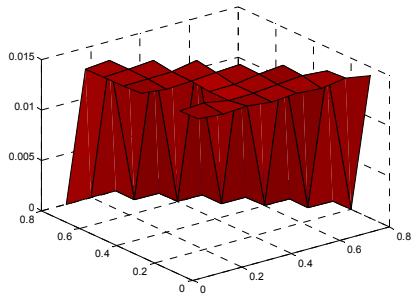
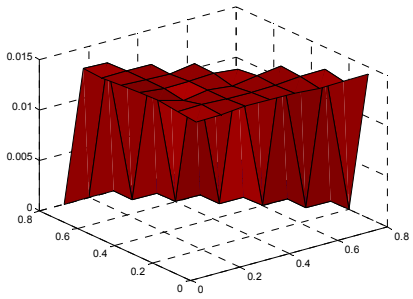
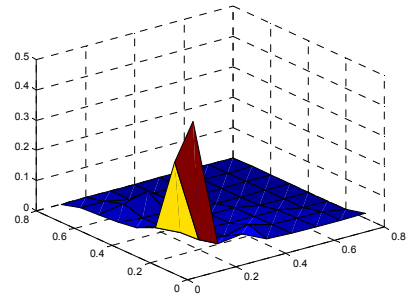
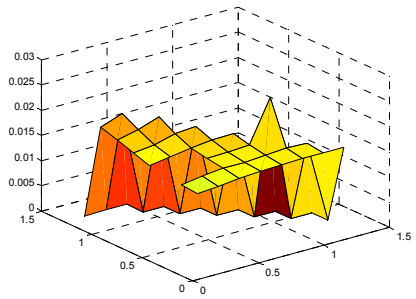
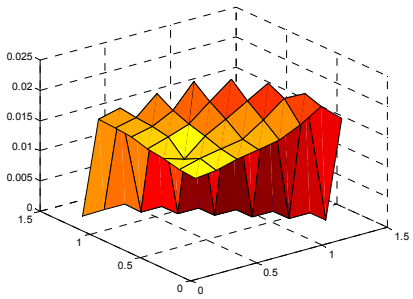
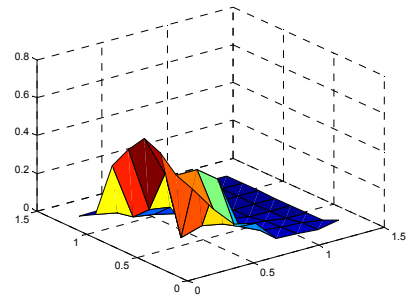
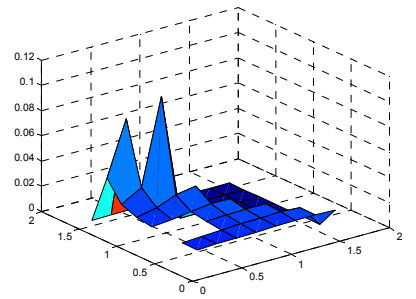
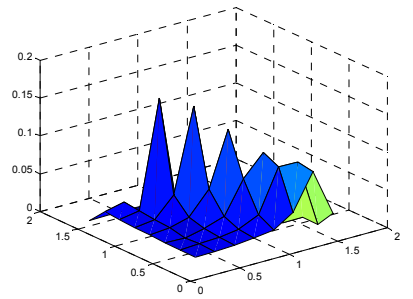
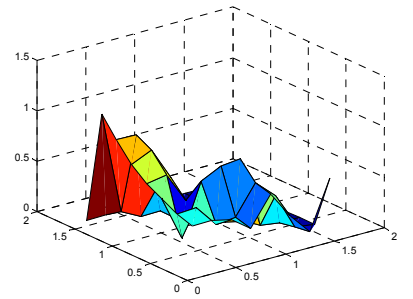
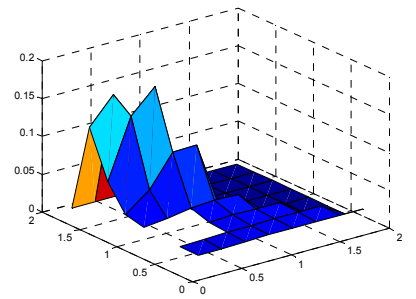
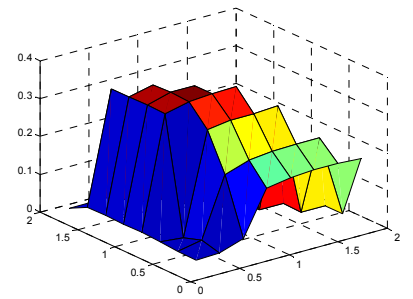
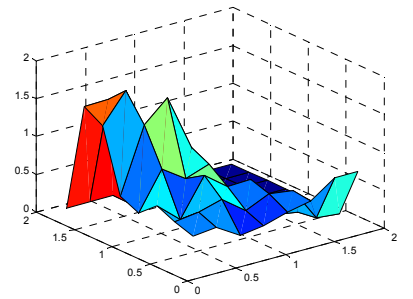
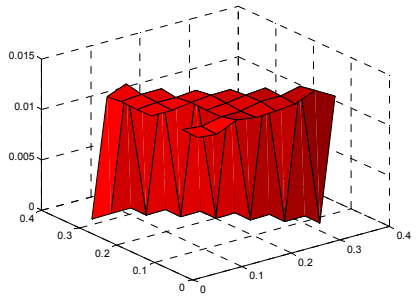
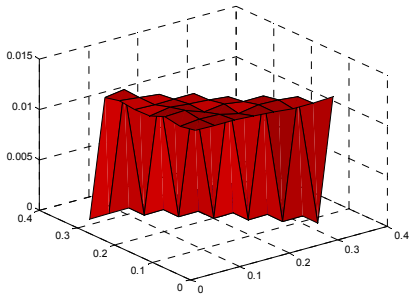
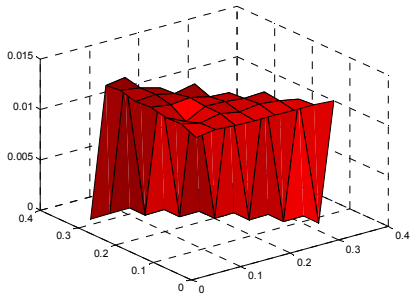
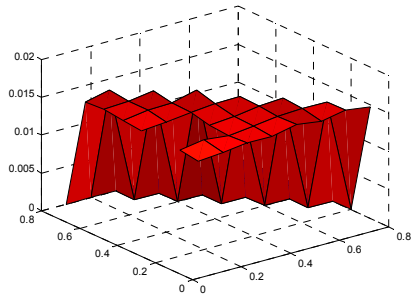
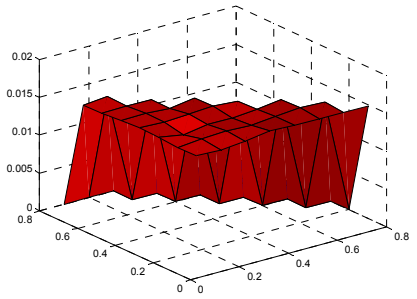
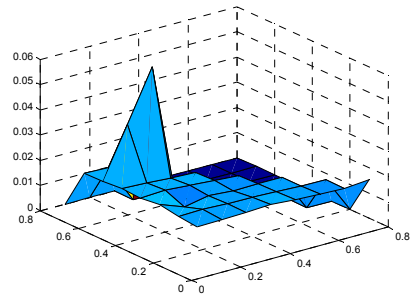
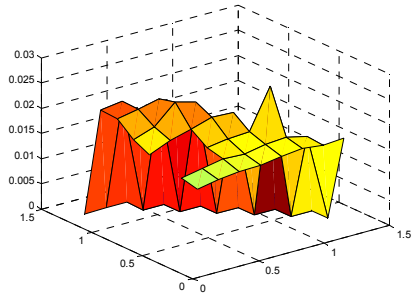
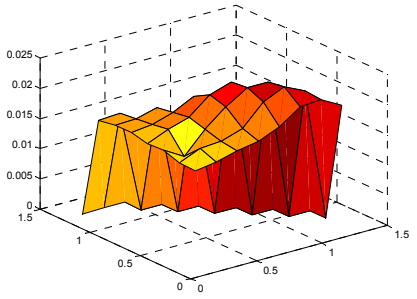
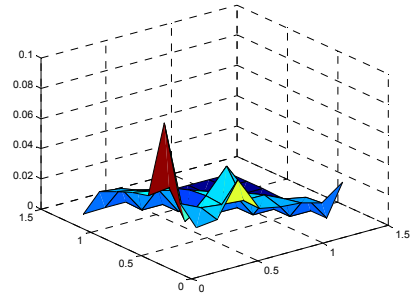
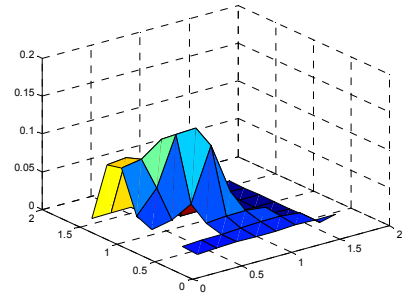
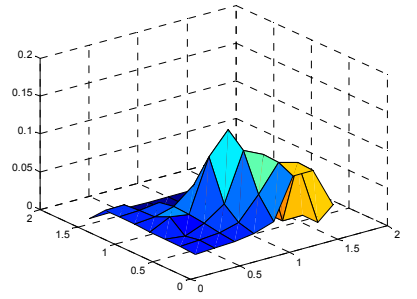
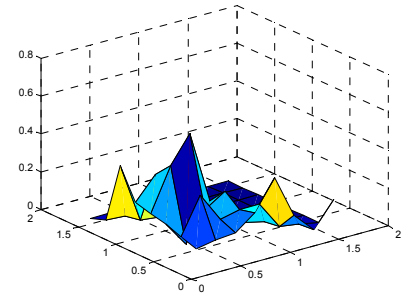
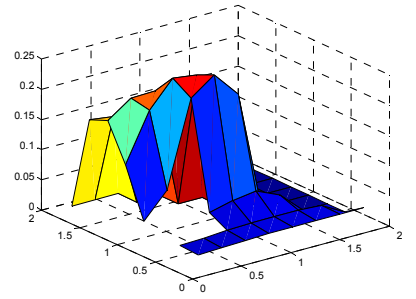
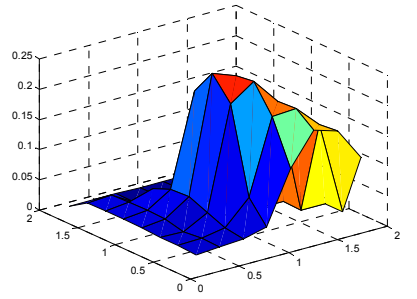
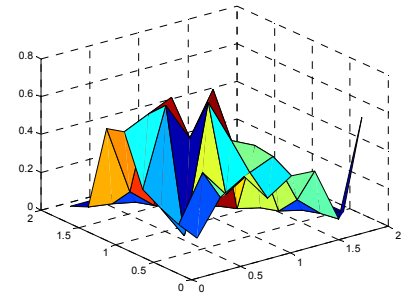
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 37: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2.

Classe 1	Classe 2	Classe 3
		
		
		
		
		

3.2.2. M/Pareto/1 avec trois classes de service

Sachant que les distributions réelles des tailles des paquets qui transitent sur l'Internet ne suivent pas la loi de Poisson mais plutôt la distribution de Pareto, nous avons étudié l'effet de cette dernière distribution sur les délais moyens observés sur un lien implémentant un ordonnanceur WRR. Tant qu'une solution analytique au problème n'existe pas, nous avons procédé à une étude par simulation. Comme dans le paragraphe 3.2.1. , nous présentons ci-dessous l'évolution du délai moyen en fonction de la charge introduite par le flux des trois classes. La pondération retenue des classes est la même que dans le paragraphe précédent: 8/1/1, 6/3/1 et 4/4/2. D'autres pondérations possibles sont incluses dans l'Annexe 4 :.

Les caractéristiques de la distribution de Pareto utilisée suivent les recommandations de l'ETSI [19]. La fonction de distribution de la loi de Pareto s'écrit $f(x) = \frac{\alpha.k^{\alpha}}{x^{\alpha+1}}$ sachant que la fonction est définie pour $x \geq k$. La valeur de α retenue est de 1,1. Celle de k est de 81,5 octets. Cette dernière représente la taille minimale d'un paquet. De plus, la distribution est tronquée à une valeur maximale égale à 60koctets, qui correspond à la taille maximale d'un paquet. Nous pouvons déterminer la valeur moyenne de la taille des paquets:

$$\mu = \int_{-\infty}^{+\infty} x \cdot f(x)dx = \int_{k}^{M} x \cdot f(x)dx + \int_{M}^{+\infty} M \cdot f(x)dx$$

$$\mu = \int_{k}^{M} x \cdot \frac{\alpha \cdot k^{\alpha}}{x^{\alpha+1}}dx + \int_{M}^{+\infty} M \cdot \frac{\alpha \cdot k^{\alpha}}{x^{\alpha+1}}dx = \alpha \cdot k^{\alpha} \int_{k}^{M} \frac{1}{x^{\alpha}}dx + M \cdot \alpha \cdot k^{\alpha} \int_{M}^{+\infty} \frac{1}{x^{\alpha+1}}dx$$

$$\mu = \frac{\alpha \cdot k^{\alpha}}{\alpha - 1}\left[\frac{1}{k^{\alpha-1}} - \frac{1}{M^{\alpha-1}}\right] + M \cdot k^{\alpha}\left[\frac{1}{M^{\alpha}}\right]$$

$$\mu = \frac{\alpha \cdot k - M \cdot \left(\frac{k}{M}\right)^{\alpha}}{\alpha - 1}$$

Avec les différentes valeurs utilisées, nous obtenons μ=480 octets. Celle-ci est très proche de la valeur moyenne de la distribution exponentielle du paragraphe 3.2.1. . Ainsi, nous pourrons mieux comparer l'effet de la distribution de la taille des paquets (Pareto ou exponentielle) sur le délai moyen.

La variation du délai moyen en fonction de la charge induite par les différentes classes présente les mêmes propriétés que dans le cas M/M/1: la classe 1 est la mieux servie dans le cas des poids 6/3/1 et 8/1/1 et le délai n'augmente fortement que lorsque la charge introduite par la classe 1 est supérieure à son propre poids alloué. De même que pour les classes 2 et 3. Mais ce qui diffère par rapport aux tableaux précédents (Tableau 35, Tableau 36 et Tableau 37) est la valeur des délais moyens sachant que la taille moyenne des paquets est la même pour les cas M/M/1 et M/Pareto/1. Pour les faibles charges, le délai moyen des paquets qui suivent la distribution de Pareto est supérieur à celle des paquets qui admettent une variation de leur taille suivant la loi exponentielle. A forte charge, et dans le cas d'une surcharge d'un flux, ce sont les délais de ce flux M/M/1 excédentaires qui admettent des valeurs supérieures à ceux des M/Pareto/1. Mais dans le cas les flux ne dépassant pas leurs portions allouées par le poids associé à la classe, c'est toujours la M/M/1 qui donne les meilleurs résultats. Les différents taux de perte observables dans à l'Annexe 4 : affirment cette constatation: à faible charge, les flux M/M/1 admettent des probabilités de perte moindre. A forte charge, (pour les charges 1 et 1,2) les flux en excès sont tous perdus, quelque soit la nature de leurs arrivées.

Ces constatations découlent des équations des délais pour une file M/G/1. Nous savons que la valeur moyenne du délai d'attente dans cette file s'écrit pour l'ordonnanceur FIFO:

$$E[s] = E[X] + \frac{\rho \cdot E[X]\left(1 + CV^2(x)\right)}{2(1-\rho)}$$

avec $CV^2(x) = \frac{\sigma_x^2}{(E[x])^2} = \frac{E[x^2] - E[x]^2}{(E[x])^2}$

Pour un flux markovien, $CV^2(x) = 1$.

Pour le flux de Pareto tronqué utilisé:

$$E[x^2] = \int_k^M x^2 \cdot \frac{\alpha \cdot k^\alpha}{x^{\alpha+1}} dx + \int_M^{+\infty} M^2 \cdot \frac{\alpha \cdot k^\alpha}{x^{\alpha+1}} dx = \alpha \cdot k^\alpha \int_k^M \frac{1}{x^{\alpha-1}} dx + M^2 \cdot \alpha \cdot k^\alpha \int_M^{+\infty} \frac{1}{x^{\alpha+1}}$$

$$E[x^2] = \frac{\alpha \cdot k^\alpha}{\alpha - 2}\left[\frac{1}{k^{\alpha-2}} - \frac{1}{M^{\alpha-2}}\right] + M^2 \cdot k^\alpha \left[\frac{1}{M^\alpha}\right]$$

$$E[x^2] = \frac{\alpha \cdot k^2 - 2M^2 \cdot \left(\frac{k}{M}\right)^\alpha}{\alpha - 2}$$ quelque soit la valeur de α.

Avec les valeurs numériques utilisées:

$$E[x^2] = \frac{1{,}1 \cdot 81{,}5^2 - 2 \cdot 60000^2 \cdot \left(\frac{81{,}5}{60000}\right)^{1{,}1}}{1{,}1 - 2} = 5607492$$

Notons que cette valeur très grande ne nous a pas surpris. Dans le cas d'une distribution de Pareto non tronquée, la variance pour une valeur de α<2 est infinie. Dans notre cas, la distribution est tronquée à 60000 octets.

Donc, en remplaçant la valeur $CV^2(x)$ =23,3 pour déterminer le délai moyen d'attente, nous constatons que $E[s]_{pareto}$ est beaucoup plus grande que $E[s]_{exponentielle}$. D'ailleurs, nous distinguons cela dans les différents schémas des tableaux ci-dessous.

Tableau 38: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1.

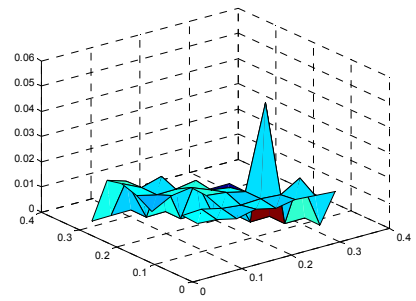
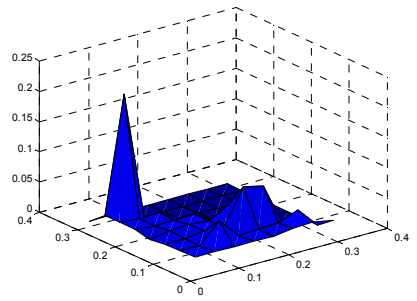
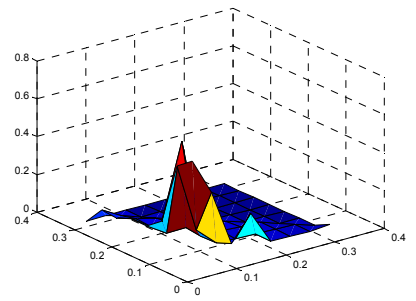
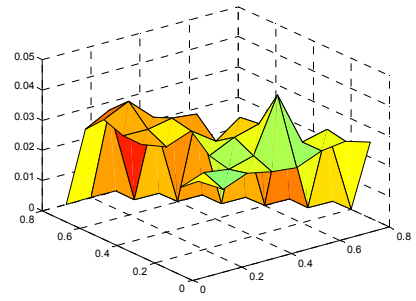
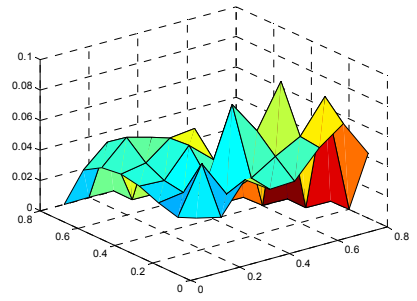
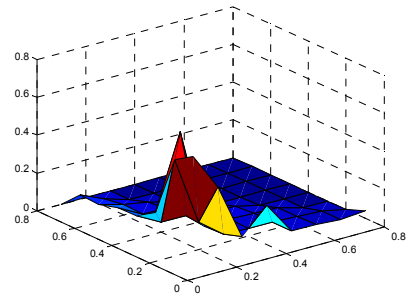
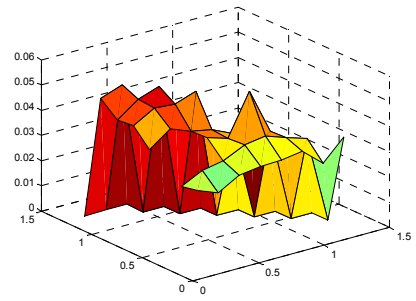
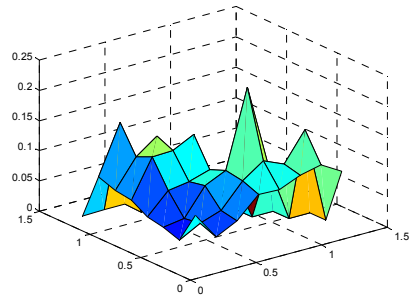
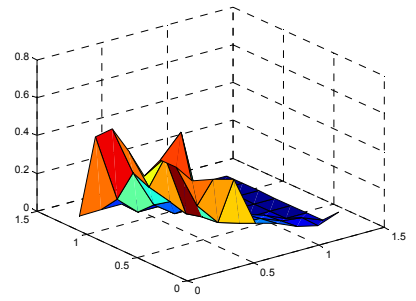
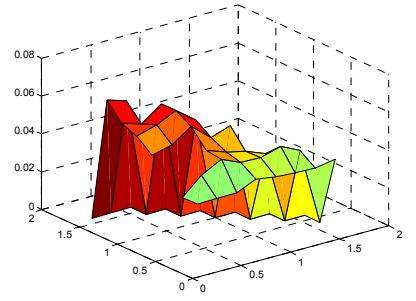
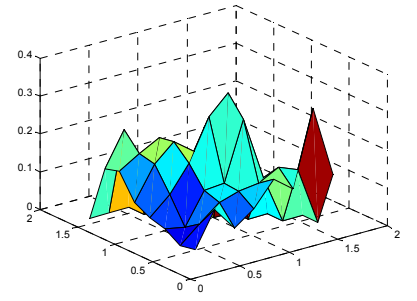
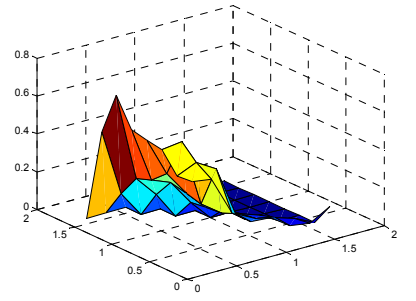
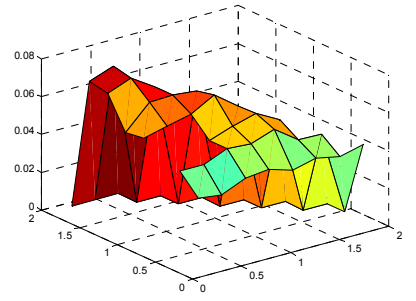
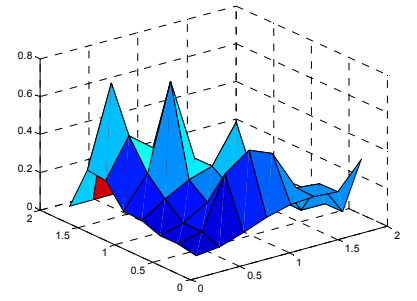
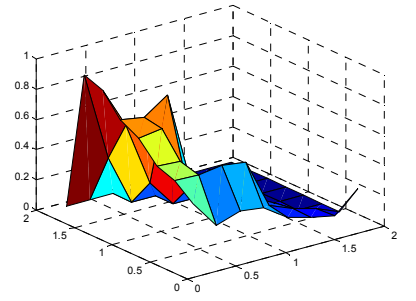
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 39: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1.

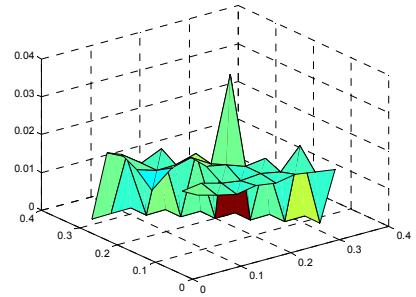
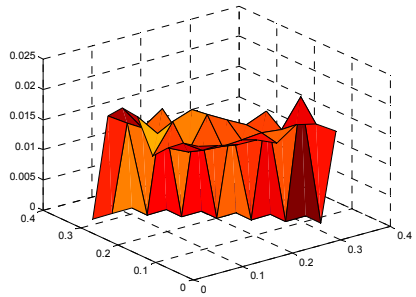
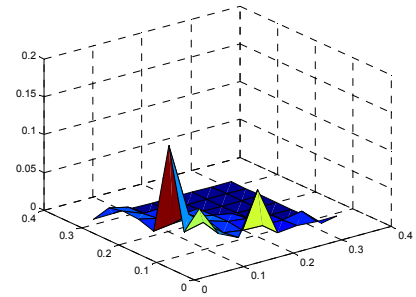
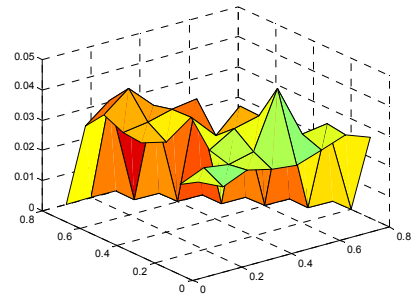
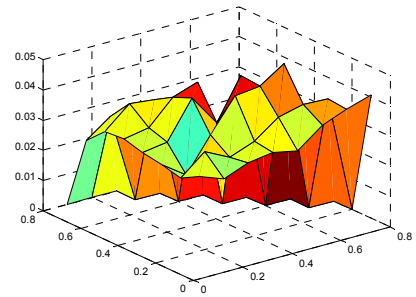
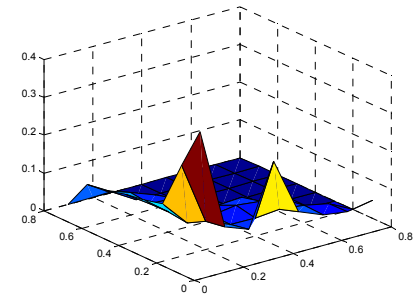
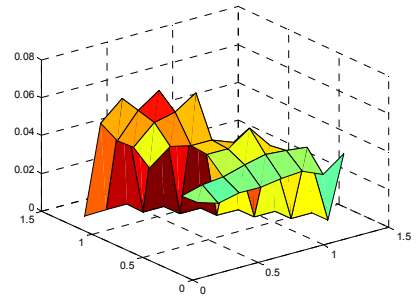
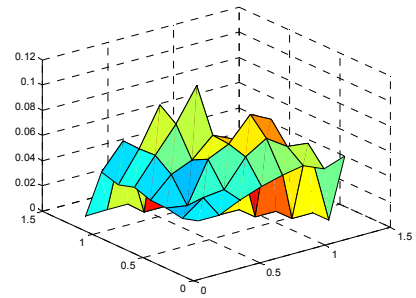
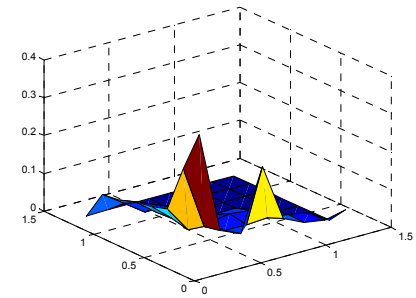
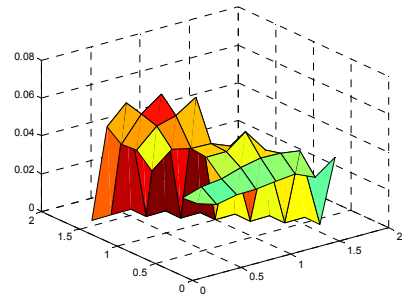
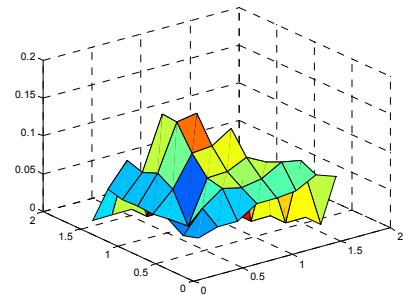
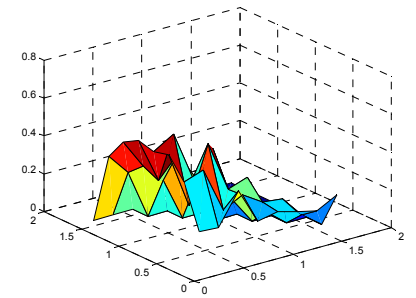
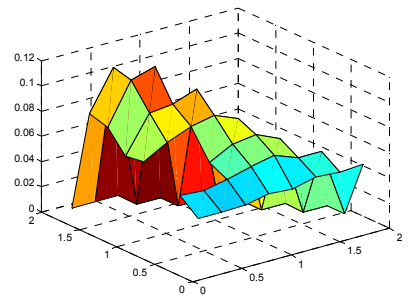
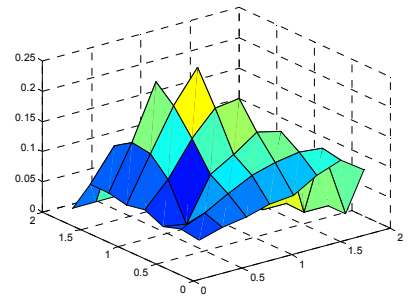
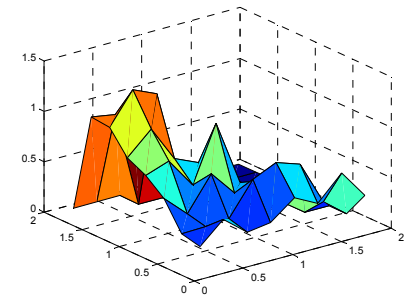
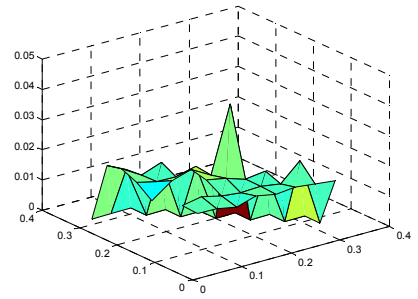
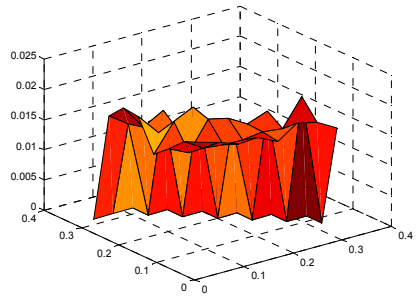
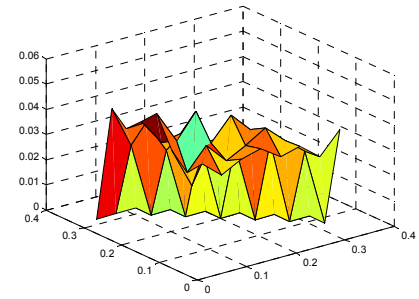
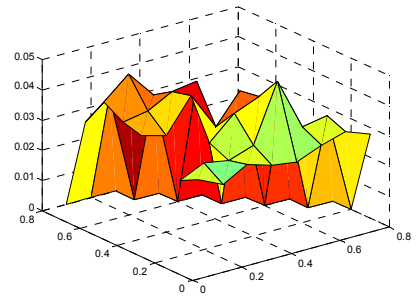
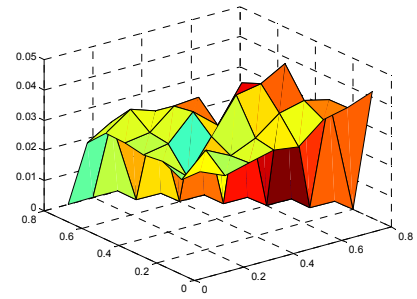
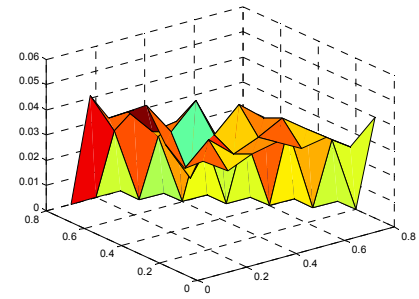
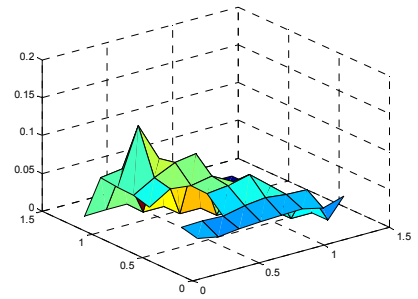
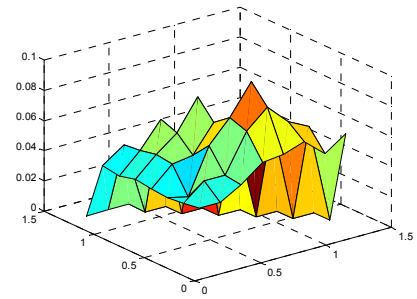
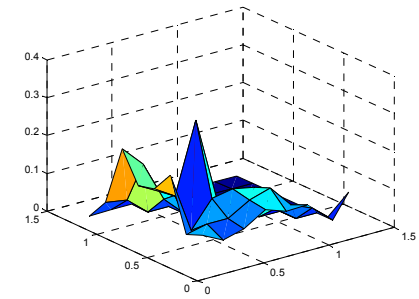
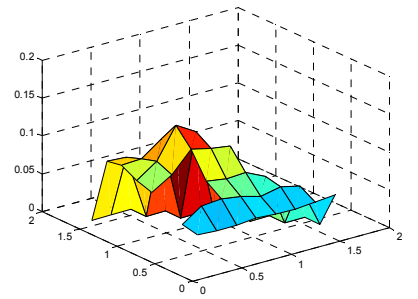
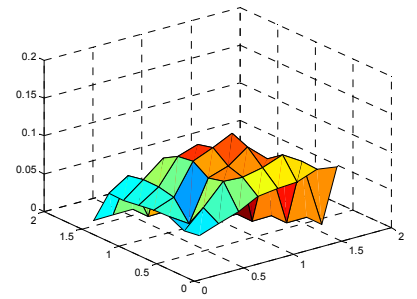
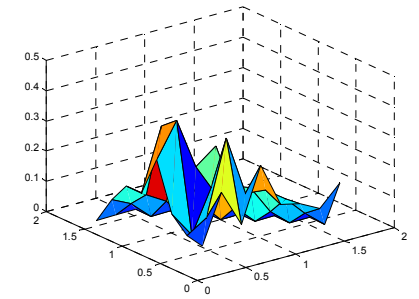
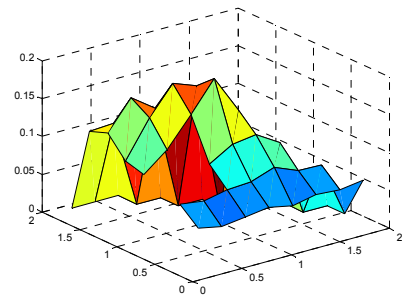
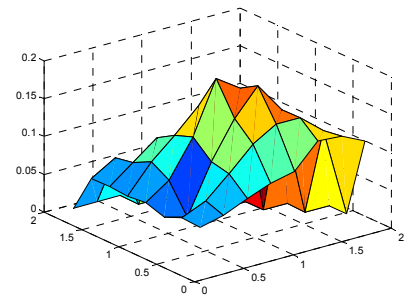
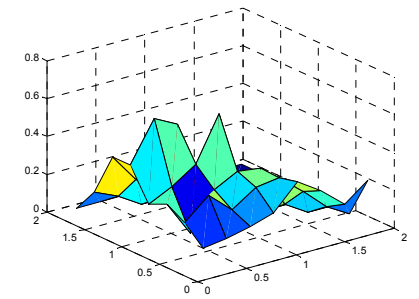
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 40: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2.

Classe 1	Classe 2	Classe 3
		
		
		
		
		

3.3. Les règles à tirer

Une solution à ce problème consisterait alors à ajuster dynamiquement les poids utilisés en fonction de la QoS perçue par les différentes classes. Plusieurs articles, notamment [11] et [4] proposent une modification dynamique des poids attribués aux différentes classes. Dans [11] les poids ω des différentes classes présents dans le réseau sont modifiés de sorte à toujours garantir un rapport d_i / d_{i+1}, des délais subits par les classes *i* sur le délai de la classe *(i+1)* de moindre contrainte, constant et fixé par avance. Dans [4] , la maximisation recherchée est une maximisation de la fonction de rentabilité du réseau. Les valeurs des poids sont fonctions non seulement du nombre d'utilisateur actif de chaque classe de service, mais aussi du tarif facturé par l'opérateur pour chaque quantité de donnée transitant par son réseau et requérant tel ou tel autre QoS. Mais, même si un gain est acquis par chaque méthode, personne n'a encore étudié les différents coûts et complexités supplémentaires ajoutés au réseau.

3.4. Application sur un réseau

Après avoir étudié les différents algorithmes d'ordonnancement sur un lien unique, nous modélisons ici un réseau entier implémentant différents mécanismes de QoS. Le modèle protocolaire du cœur du réseau est basé sur DiffServ et MPLS. Nous avons procédé à une étude par simulation afin d'analyser les différents paramètres de performance du réseau.

3.4.1. Le modèle du réseau

Pour pouvoir simuler facilement, et pour pouvoir interpréter les résultats obtenus, nous avons choisi un réseau de référence dont la topologie est illustrée dans la Figure 26 ci-dessous.

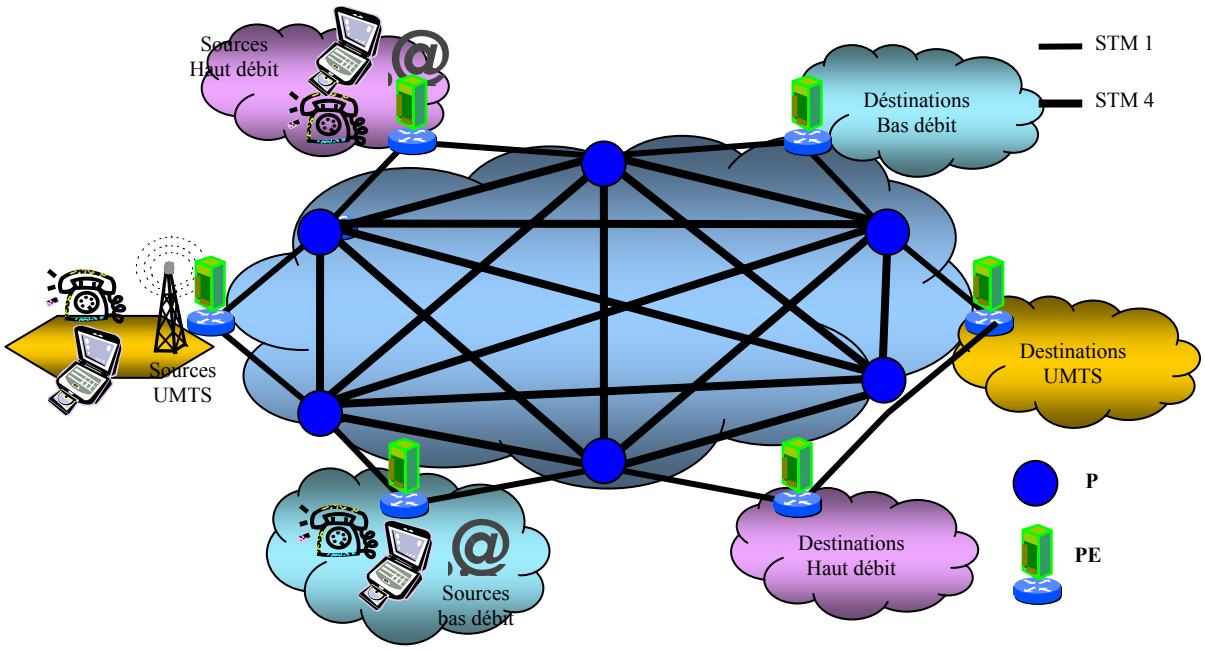


Figure 26: La topologie du réseau considéré

Le cœur du réseau utilisé est complètement maillé par des liens à haut débit: comme un scénario de réseaux futurs où des *téra-routeurs* seroont interconnectés par les liaisons térabits. Une liaison STM-1 offre 155Mbits/sec de débit. Quant à un lien STM-4, il présente 622Mbits/sec comme débit utilisable.

Trois types de sources débitent sur le cœur du réseau : des sources UMTS, des sources haut débit, et des sources bas débit. Le profil des trafics est différent suivant le type du réseau d'accès.

Les liens STM-4 et STM-1 offrent des délais de propagation constants de 10 millisecondes. Ces délais comportent aussi les différents temps de traitement comme l'encapsulation/désencapsulation, le temps d'accès mémoire, le temps de vérification du CRC, etc.

Les trois types de réseaux d'accès connectés au cœur du réseau, regroupent les différentes possibilités des réseaux existants.

- L'accès des réseaux UMTS représente la famille des réseaux mobiles : GSM, GPRS mais aussi IEEE 802.11. La mobilité et le partage de la bande passante confèrent à ces réseaux d'accès mobiles des demandes différentes que ceux des réseaux terrestres fixes. Les réseaux satellites ont également des demandes différentes. Ce dernier type de réseau, bien qu'il soit prometteur pour les services multimédia, ne sera pas considéré dans notre étude.
- Les réseaux d'accès à bas débit regroupent dans notre cas un agrégat de sources bas débits individuelles.
- Les réseaux d'accès à hauts débits sont représentés par les connexions qui relient les grandes sociétés ou les entreprises aux cœurs des réseaux des opérateurs.

Le modèle de différenciation de service utilisé dans DiffServ se concrétise par la Figure 27.

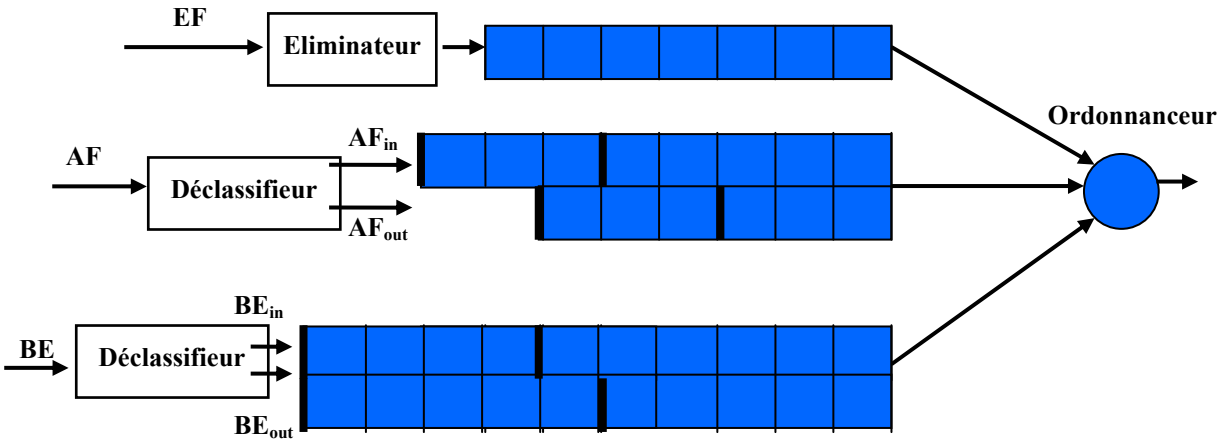


Figure 27: La différentiation de service dans les nœuds du réseau

Le trafic est d'abord classifié en EF, AF et BE. Puis chaque flux sera mis dans la file d'attente appropriée. Si le trafic de la classe EF excède le contrat négocié, les paquets excédentaires seront éliminés. Le trafic conforme, sera mis dans l'unique file d'attente de EF. Pour les trafics AF et BE, les flux conformes seront mis dans la file AF_{in} et BE_{in} respectivement et les paquets non conformes seront déclassifiés en AF_{out} et BE_{out}. Pour

simplifier la configuration, nous avons mis un seul type de flux AF qui représente les données critiques.

Initialement, l'implémentation de MPLS dans le cœur d'un réseau était motivée par deux avantages. D'une part, les paquets seront **commutés**, et non plus routés. Cela augmenterait la vitesse de service des flux. Et d'autre part, la possibilité de créer différents chemins en fonction de la qualité de service demandée (Ingénierie de trafic). Or, avec les nouvelles technologies de routage et la vitesse des mémoires des routeurs qui augmentent de plus en plus, le premier avantage n'est plus déterminant. En effet, le gain en vitesse d'un commutateur de label par rapport à un routeur de paquet n'est plus toujours significatif. Par contre, les avantages des LSP pour l'ingénierie de trafic, rendent MPLS incontournable pour offrir la qualité de service requise par les applications. Dans le cadre de l'étude effectuée, nous avons utilisé MPLS pour implémenter un équilibrage de charge entre différents liens du réseau.

Sachant que le protocole réseau des applications temps réel (généralement UDP) ne s'adapte pas aux contraintes imposées par la congestion dans le cœur du réseau, une bonne politique pour garantir un faible délai et minimiser les pertes des paquets est la politique FIFO.

Les paramètres des sceaux à jetons au niveau des classifieurs situés aux entrées du réseau –entre CE et PE– sont :

Tableau 41: Les caractéristiques des Sauts à Jetons

	EF	AF	BE
Débit Soutenu	5Mb/s	800kb/s	200kb/s
Taille du burst	200 octets	1 koctets	500 octets

La taille maximale acceptable d'une rafale appartenant à la classe EF est de 200 octets, qui est moindre que la taille du paquet (500 octets en général). Cela signifie que si le débit est plus grand que le débit soutenu (5 Moctets dans notre cas), le paquet est rejeté immédiatement; il ne sera pas déclassifié. Cela représente le contrôle d'admission des flux EF. Le mécanisme de gestion des files d'attente des classes AF et BE est RED. Pour les paquets conformes, la taille de la file est de 40 paquets. A partir du vingtième paquet, ces derniers seront rejetés avec une probabilité de 2%. Les paquets non conformes (qui violent le sceau à jeton) de chaque classe, sont déclassifiés et mis dans une file de même classe mais de priorité inférieure: la taille maximale de la seconde file dans chaque classe est de 20 paquets. A partir du dixième paquet, la probabilité de perte est de 10%.

3.4.2. Les Modèles de trafics

Dans les réseaux mobiles, et spécialement les réseaux mobiles publics – GSM, GPRS et UMTS– le goulot d'étranglement est la liaison radio entre le mobile et le réseau terrestre. Pour avoir un partage équitable et optimal de la bande passante entre les différents terminaux connectés au réseau, la couche MAC doit fragmenter les paquets en petits segments. Sans entrer ici dans les détails des techniques mises en place, cette intervention contribue d'une façon importante à la caractérisation statistique du trafic.

Nous supposerons que le flux en sortie du réseau UMTS est EF pour la voix, et AF pour les données. C'est-à-dire qu'il n'y a pas de trafic de type BE en sortie des réseaux mobiles. En effet, cela se justifie par le fait que si un paquet passe le réseau d'accès radio, qui est le goulot d'étranglement, il ne sera pas permis de le perdre dans le cœur du réseau, où les ressources ne sont pas aussi rares. Même pour les flux de trafic « natifs » en BE, autrement dit

le mail, cela s'applique du fait que les serveurs *mail* sont en amont du serveur SGSN à partir duquel, dans toutes les architectures UMTS, les mails sont envoyés aux différents destinataires (voir le Chapitre 2 pour plus de détails).

Dans le cas de l'UMTS, nous avons pris [26] un trafic MMPP-2 de paramètres :

$$Q = \begin{pmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{pmatrix} \quad \text{et} \quad \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Les débits et les durées des phases sont décrits ci-dessous :

Tableau 42: Caractéristiques des flux UMTS

	Haut Débit		Bas Débit		UMTS	
	EF	AF	EF	AF	EF	AF
Débit moyen (Mb/s)	1	2	0,2	0,2	2	0,7
Débit de l'état 1 (Mb/s)	1,2	2,6	0,3	0,3	2,6	1
Débit de l'état 2 (Mb/s)	0,7	1	0,1	0,1	1	0,2

Les caractéristiques des durées des phases et des tailles des paquets sont résumées dans le Tableau 43 ci-dessous :

Tableau 43: Les paramètres MMPP-2 des flux UMTS

	Taille des paquets (Octets)	Durée des phases σ_i (secondes)
Etat 1	500	0,5
Etat 2	100	0,3

Le temps inter arrivée des paquets est exponentiellement distribué avec pour paramètre taille(bit)/débit(bit/sec). La Figure 28 ci-dessous donne la fonction de répartition du temps inter arrivé d'un MMPP-2 par rapport aux deux exponentielles formées par les paramètres λ_1 et λ_2.

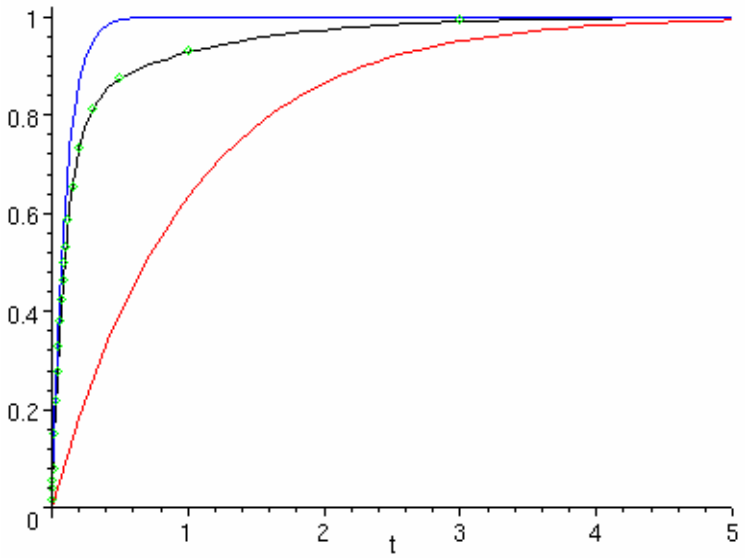


Figure 28: La fonction de répartition du temps inter arrivée d'une distribution MMPP-2.

Les trafics des sources notées "haut débit" et "bas débit" sont supposés ON-OFF de durées des phases qui suivent la distribution de Pareto. Cela engendrera des flux auto similaires après agrégation au niveau des commutateurs P. La durée moyenne de la phase ON est de 0,5 sec, celle de la phase OFF est de 0,2 sec. La taille des paquets est constante de 500 octets. Le paramètre α de mise à l'échelle est de 1,5.

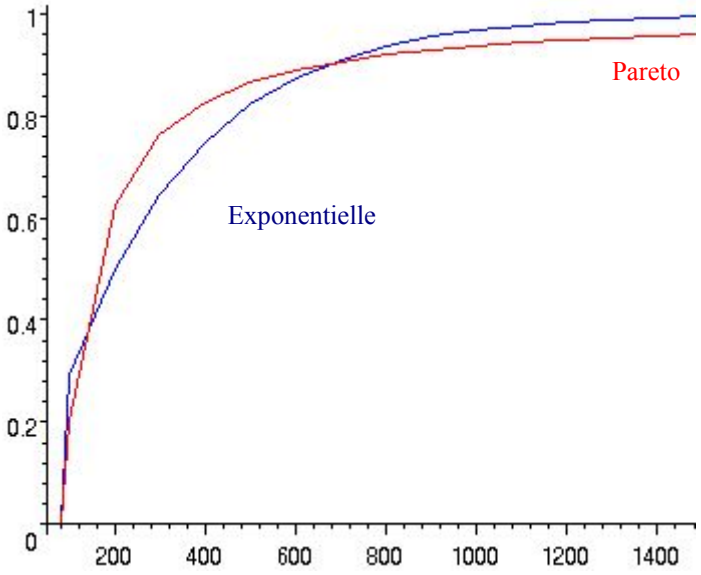


Figure 29: La fonction de répartition du temps inter arrivée d'une distribution de Pareto et d'une exponentielle de même moyenne.

3.4.3. Les paramètres des scénarios de simulations

Les débits des sources en fonction des classes, sont exprimés en Mb/s :

Tableau 44: Les débits des sources par QoS.

	Débit total par classe de source			Débit total par Source
	EF	AF	BE	
Haut Débit	5.5	15	4	24.5
Bas Débit	0.8	1.4	0.8	3.0
UMTS	3.2	2.9	0	6.1
Total par destination	9.5	19.3	4.8	33.6

Les débits des sources en fonction de leur destination, sont exprimés en Mb/s des sources :

Tableau 45: Les débits des émissions

Récepteur / Emetteur	Haut Débit			Bas Débit			UMTS			Total par Source	Débit du lien d'interconnexion
	EF	AF	BE	EF	AF	BE	EF	AF	BE		
Haut Débit	4	8	2	0.5	2	1	1	5	1	24.5	32
Bas Débit	0.5	1	0.5	0.2	0.2	0.2	0.1	0.2	0.1	3.0	6
UMTS	1	2		0.2	0.2		2	0.7		6.1	10
Total par destination	5.5	11	2.5	0.9	2.4	1.2	3.1	5.9	1.1	33.6	

A partir du tableau ci-dessus, nous constatons que 24 flux de trafic transitent dans le réseau. Pour faciliter l'interprétation des résultats, le protocole de transport utilisé pour tous les flux est UDP : il n'y a pas de retransmission des paquets perdus ni de gestion de fenêtres de congestion. Nous considérons que le modèle de source prend en compte tous ces aspects.

3.4.4. Les résultats

Le but recherché dans ces simulations est la comparaison des différentes méthodes d'ordonnancement pour définir laquelle est la plus adaptée à la QoS recherchée. Dans toutes les simulations, trois résultats sont mesurés :

1. Le délai de transit dans le réseau: c'est le temps passé entre l'instant de création du paquet par la source applicative et son instant d'arrivée à la couche applicative destinatrice.
2. La gigue des paquets: c'est la valeur absolue de la différence du délai entre deux paquets consécutifs.
3. Le taux de perte: c'est le rapport entre le nombre de paquets perdus sur le nombre de paquets émis.

L'abscisse des courbes fait référence à l'état du réseau simulé: chaque état correspond à un ensemble de sources UMTS, haut débit et bas débit. Ces différents états sont synthétisés dans le tableau ci-dessous.

Tableau 46: Les états simulés

	UMTS	Haut Débit	Bas Débit	Débit total
Etat 1	2	5	2	140.7 Mb/s
Etat 2	2	10	2	253.2 Mb/s
Etat 3	2	15	2	385.7 Mb/s
Etat 4	2	20	2	508.2 Mb/s

Le Tableau 47 présente les différents rapports de chaque flux de trafic par rapport au flux total présent dans chaque état étudié:

Tableau 47: Proportion des trafics

	EF (%)	AF (%)	BE (%)	Débit total
Etat 1	25.2	59.4	15.4	140.7 Mb/s
Etat 2	20.9	62.7	16.4	253.2 Mb/s

Etat 3	23.4	60.6	16.0	385.7 Mb/s
Etat 4	23.3	60.7	16.0	508.2 Mb/s

3.4.4.1. Sans QoS

Comme base des différentes comparaisons que nous allons effectuéer, les premières simulations sont réalisées pour un réseau sans QoS, qui correspond à l'état actuel des réseaux IP. Aucune différentiation de traitement entre les différents flux n'est mise en place et tous les flux sont routés sur la base du plus court chemin.

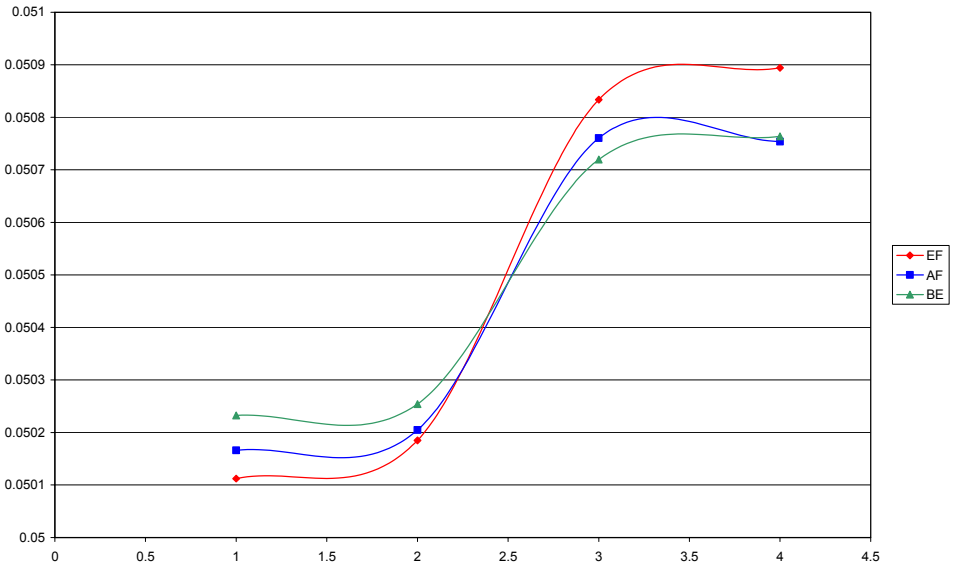


Figure 30: Délai (m sec) sans QoS

Les résultats de performance de ce type de réseau sont prévisibles: tous les flux sont traités de la même manière. A partir d'une certaine charge moyenne, le réseau n'offre plus de délai acceptable pour les communications temps réels. La QoS correspondante devient donc médiocre.

Le délai des paquets, de l'ordre de 50ms, est surtout dû au temps de traversé du réseau. Le délai de propagation sur chaque lien est de 10ms. Or, en moyenne, chaque paquet traverse 5 liens pour arriver à destination.

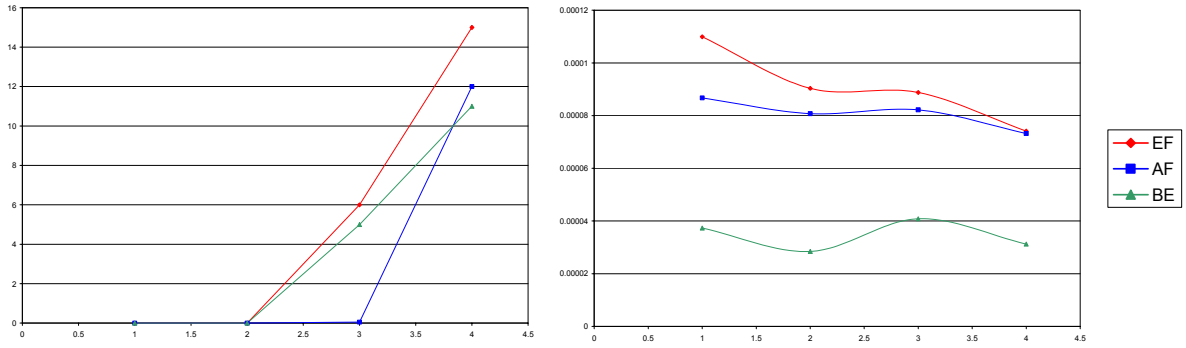


Figure 31: Taux de perte et gigue sans QoS

Tout se passe bien jusqu'au scénario 4. A partir de 15 sources HD, le taux de perte devient très important. Comme les files les moins courtes sont celles de la classe EF, cette dernière est la plus pénalisée.

La première méthode de différenciation est la priorité absolue entre les flux.

3.4.4.2. Priorité « Head Of the Line » non préemptive

En donnant une priorité absolue de la classe EF sur la classe AF et de la classe AF sur la classe BE, le délai des paquets EF (Figure 32) reste relativement constant lorsque le nombre de sources augmente. Ce sont les paquets BE qui subissent le plus grand délai.

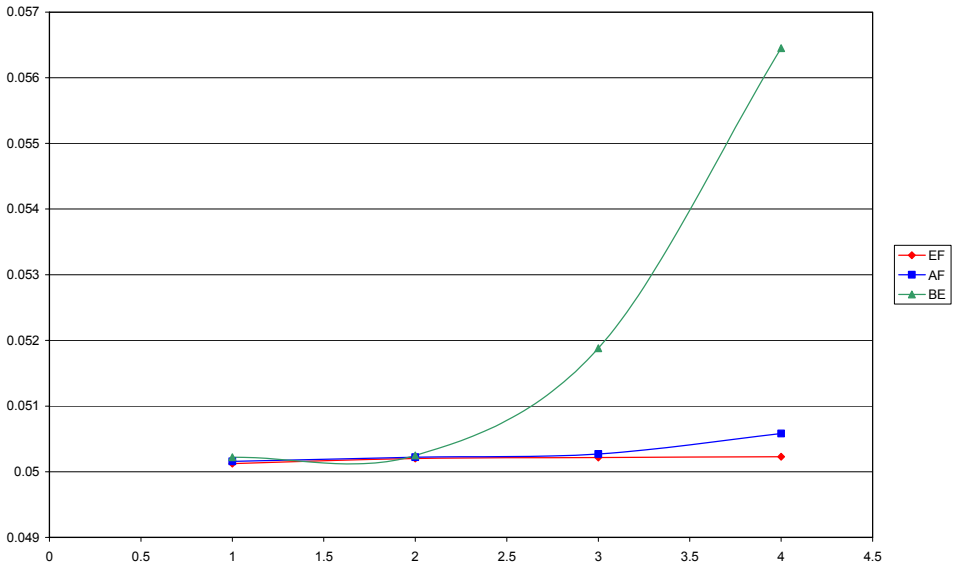


Figure 32: Délai (msec) pour la priorité absolue

La priorité absolue est aussi visible dans les courbes des taux de perte et de la gigue (Figure 33).

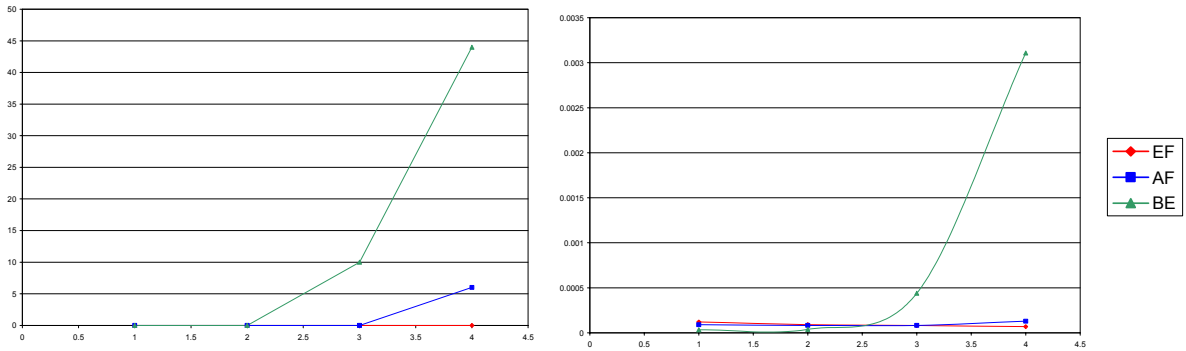


Figure 33: Taux de perte et gigue pour la priorité absolue

Le taux de perte important - 45% - des paquets BE justifie le peu d'utilisation de cette méthode dans les réseaux réels.

La deuxième amélioration est celle d'un ordonnancement en fonction du poids alloué à chaque service.

3.4.4.3. WRR

Dans le cas d'un ordonnanceur WRR, nous rappelons que les files sont servies proportionnellement au poids affecté à chacune des files. Si une file est vide ou n'utilise pas totalement sa part, le reste sera réparti équitablement, en fonction du poids, entre le reste des classes.

L'effet du poids choisi dans l'ordonnanceur WRR est primordial pour avoir la QoS recherchée. C'est ce qui sera illustré dans ce qui suit.

3.4.4.3.1. WRR avec poids EF/AF/BE : 6/3/1

La première série de mesures est pour un poids respectif des classes EF/AF/BE de 6/3/1. Cela favorise le passage des flux EF par rapport aux flux AF et BE.

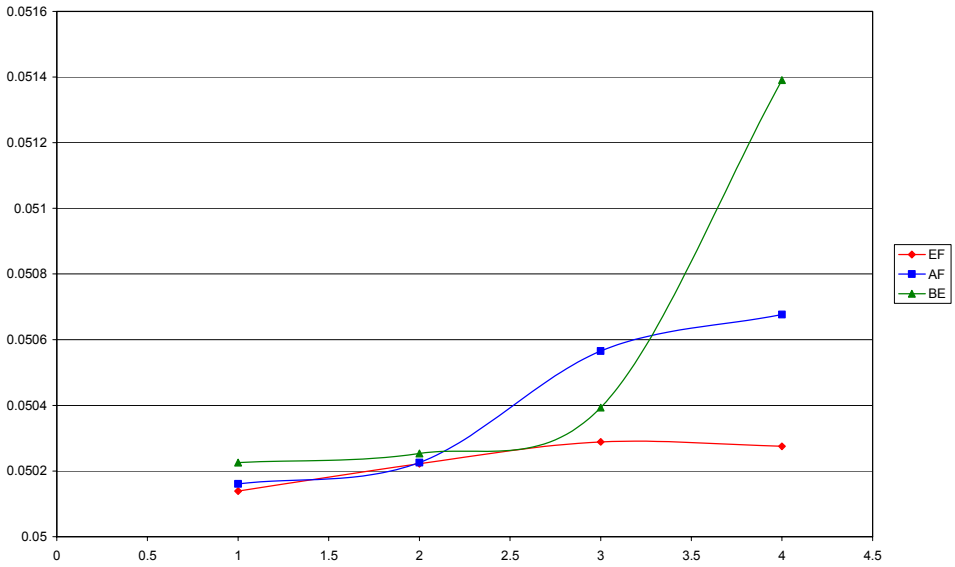


Figure 34: Délai avec WRR (6/3/1)

Dans le cas de cette combinaison de poids, et sachant que les proportions suivent le Tableau 47, nous constatons que :

- Les flux EF sont les mieux servis : 60% de la bande passante leur est alloué avec une proportion de 20 à 25% de trafic réel.
- Les flux AF sont servis toujours moins bien que les flux EF, mais mieux que BE sauf au point 3. En ce point, le rapport de la bande passante allouée à AF (la partie spécifique à AF plus la partie non utilisé de EF) sur la charge de AF devient plus faible que ce même rapport pour BE.
- Les flux BE sont les moins bien servis: 10% de la bande passante leur sont alloués avec une proportion de 15 à 16% de trafic réel.

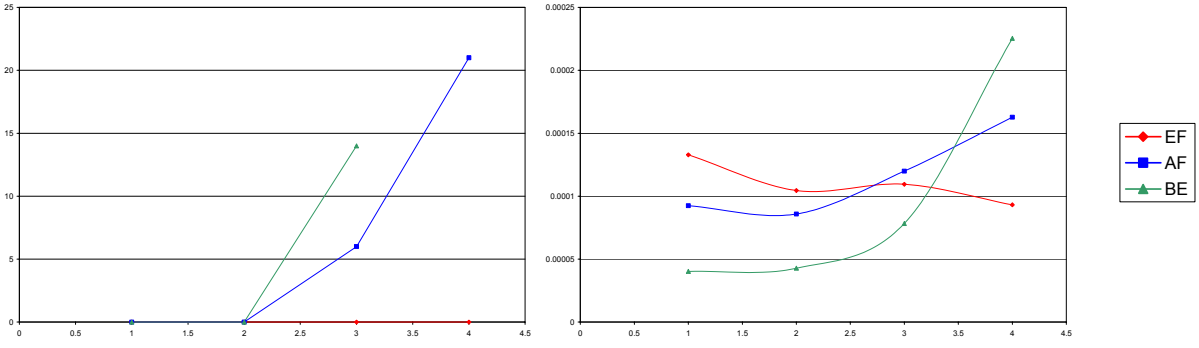


Figure 35: Taux de perte et gigue pour WRR (6/3/1)

Avec ces poids, les QoS recherchées sont bien obtenues.

3.4.4.3.2. WRR avec poids EF/AF/BE : 4/4/2

En changeant le poids des files, nous obtenons d'autres résultats: le délai des paquets EF devient le plus grand lorsque la charge est importante (cf. Figure 36) et les paquets BE sont les mieux servis.

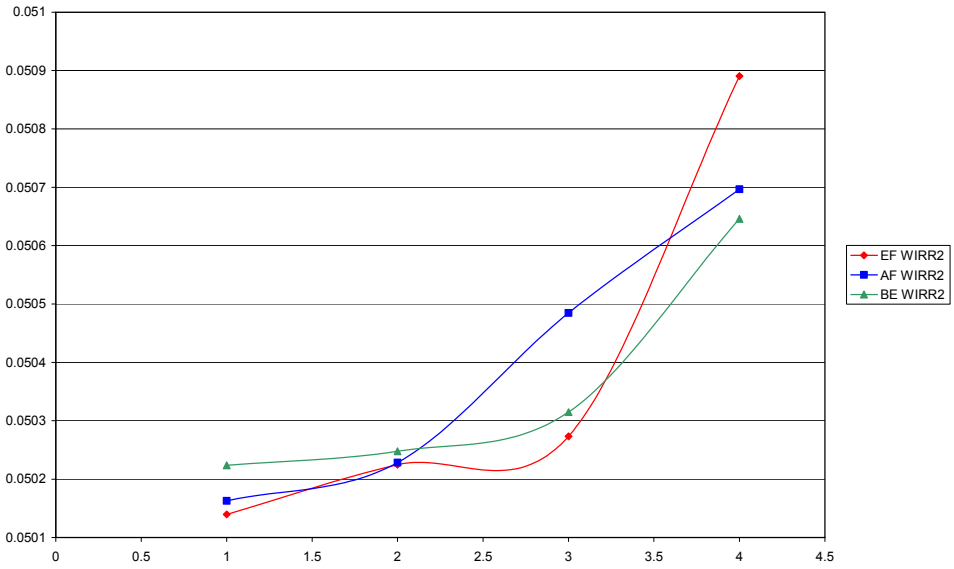


Figure 36: Délai pour un WRR 4/4/2.

Dans les deux figures ci-dessous, i.e. Figure 37, nous notons d'autre conséquences du mauvais choix des poids: les paquets BE ne subissent pas de pertes alors que ceux de EF en ont. Les flux AF subissent la plus mauvaise QoS.

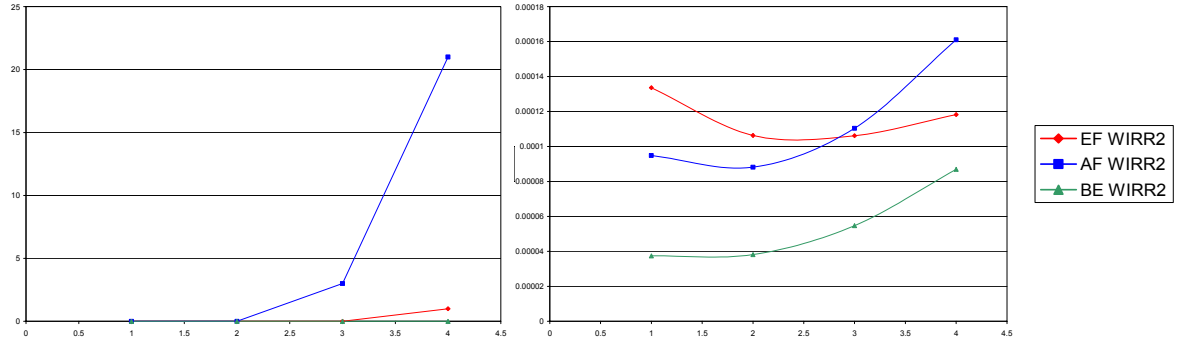


Figure 37: Taux de perte et gigue pour WRR (4/4/2)

La même observation peut être faite sur la gigue : la classe BE est toujours la mieux servie.

3.4.4.3.3. WRR avec poids EF/AF/BE : 6/3/1 et ingénierie de trafic

Afin d'équilibrer la charge des liens et des nœuds, MPLS offre de bonnes méthodes. Dans le cas étudié ci-dessous en Figure 38, les flux de trafic émis par les sources ***Hauts Débits*** sont divisés. Ainsi, la charge des liens est divisée, ce qui se répercute sur les indices de performance du réseau.

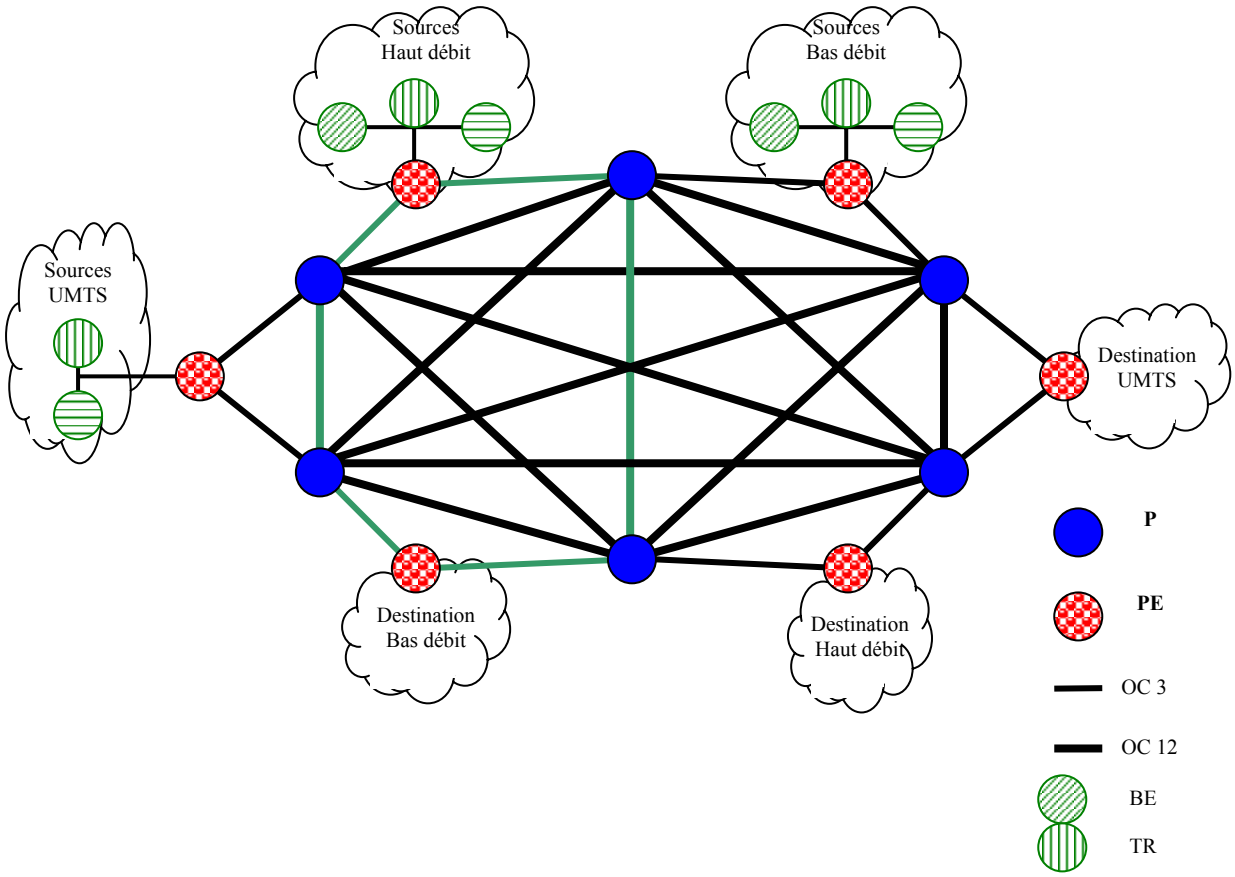


Figure 38: Exemple d'équilibrage de charge entre les liens pour la destination bas débit.

Nous pourrons alors admettre plus d'utilisateurs dans un même réseau tout en ayant la même QoS. Le délai EF est le plus petit, celui de BE est le plus grand.

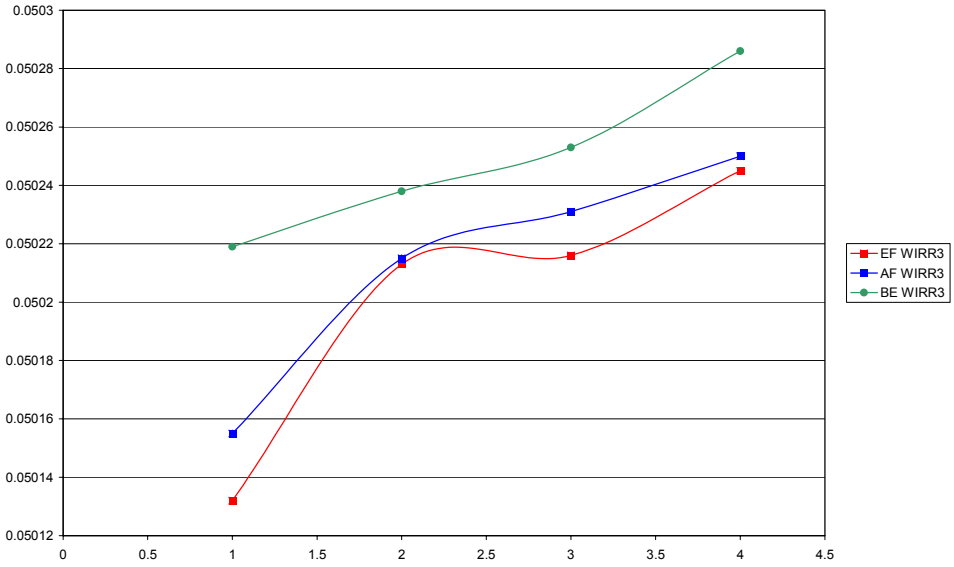


Figure 39: Délai avec les LSP

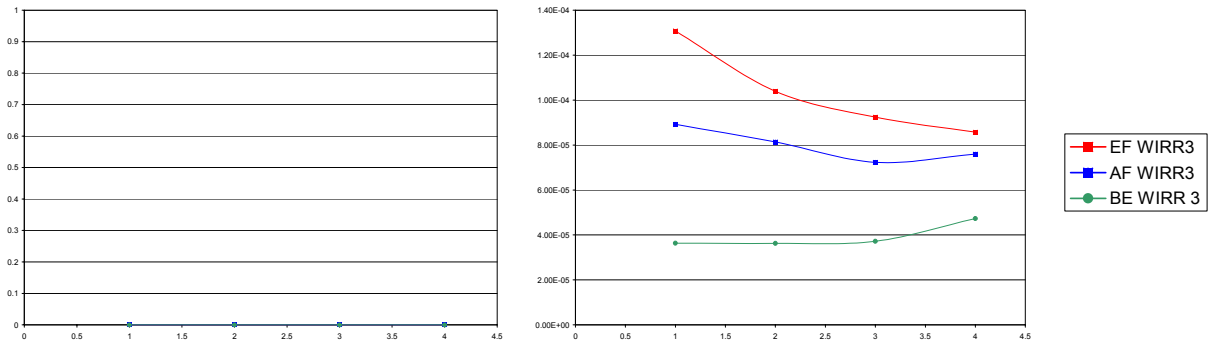


Figure 40: Taux de perte et gigue avec LSP

Avec les LSP, la charge est divisée entre les liens, ce qui se répercute sur les performances des trafics : le taux de perte est nul pour tous les flux EF, AF et BE.

Si les poids sont bien choisis, WRR est un bon compromis mais son inconvénient est que le trafic temps réel EF peut ne pas être toujours le mieux servi. Pour combler ce défaut, nous étudions un nouveau type d'ordonnancement qui combine la priorité et la différenciation par poids.

3.4.4.4. CB-WRR : Class Based WRR

Le trafic EF est servi prioritairement ; les trafics AF et BE sont servis en fonction du poids WRR alloué à chacune de ses files. Cette technique garantit d'une part un délai faible aux flux EF et d'autre part, un service des flux AF et BE en fonction du poids respectif et ainsi de la QoS voulue pour chaque type de classe.

Une première approche propose l'observation de l'effet du rapport du poids AF et BE sur les performances de ces derniers. La QoS du trafic EF n'est point affecté par ce choix.

Pour alléger ce paragraphe, nous exposerons ci-dessous uniquement la variation des délais. Les autres mesures (voir Annexe 5 :) corroborent aux mêmes interprétations.

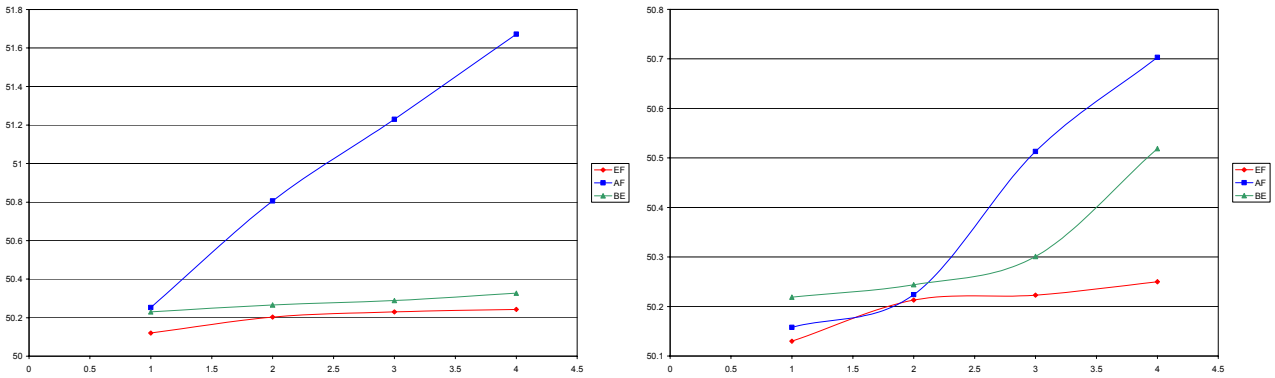


Figure 41: Délai des flux (1/1) et (2/1)

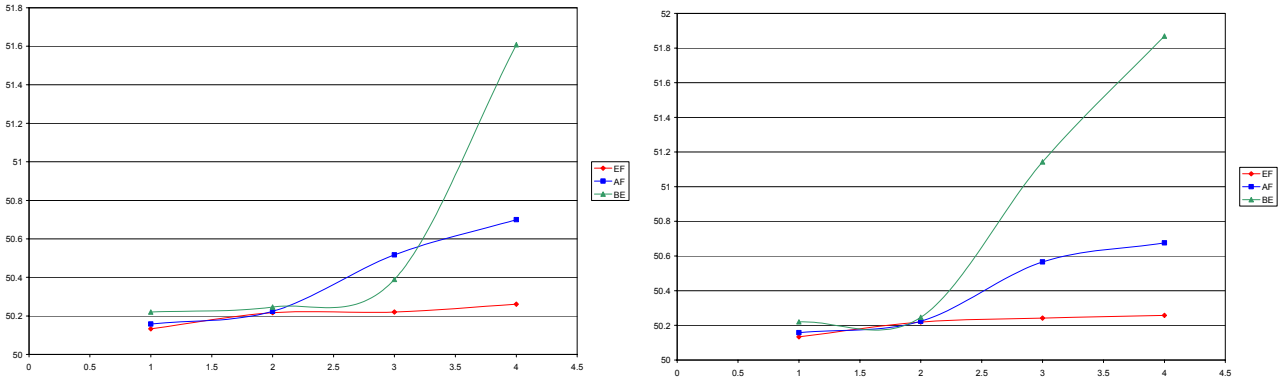


Figure 42: Délai des flux (3/1) et (4/1)

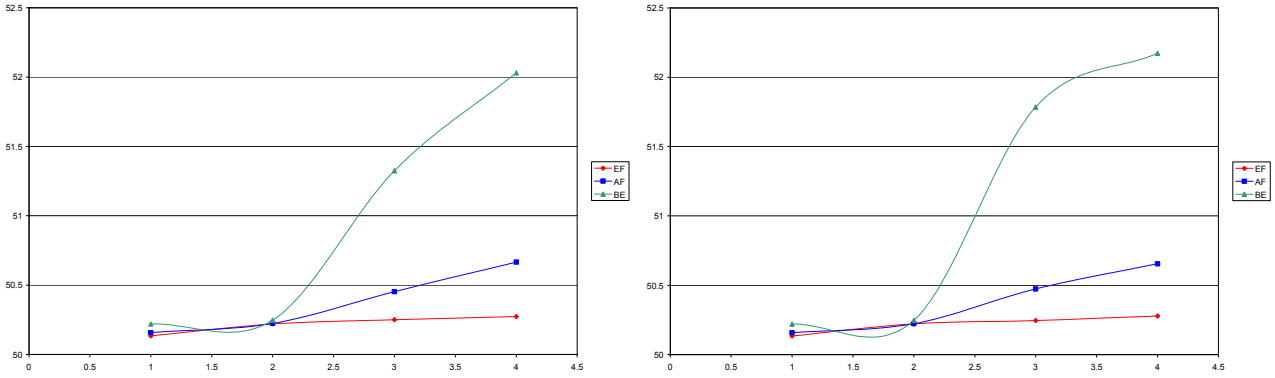


Figure 43: Délai des flux (5/1) et (6/1)

Les Figure 41, Figure 42 et Figure 43, montrent que plus le poids alloué à AF est grand, plus le délai est réduit, plus la QoS est meilleure. Il faut noter que EF est toujours servi de la même manière: EF est dans tous les cas de figures strictement prioritaire. Or comme AF doit être mieux servi que BE, le poids retenu est 6/1. Pour les autres combinaisons de poids, BE est parfois mieux servi que AF. Cette valeur est due au rapport entre le trafic AF et BE. Une comparaison entre les différentes méthodes d'ordonnancement pour la même classe est donnée en annexe.

3.4.4.4.1. Choix des poids du CB-WRR

Avec la configuration déjà citée, le meilleur poids qui donne la QoS souhaitée est celui qui alloue au trafic AF 6 fois de ressources que ceux allouées au flux BE. En changeant la matrice de trafic, et pour une même QoS, le poids dans les ordonnanceurs change aussi. Dans ce paragraphe, nous effectuons les mêmes simulations que dans le paragraphe précédent mais avec une distribution différente de la matrice de trafic. Les flux AF étaient majoritaires dans le cas précédent (noté cas 1). Deux cas de figures sont étudiés ci-dessous : un avec les flux EF majoritaires (noté cas 2) et un avec les flux BE majoritaires (noté cas 3). Le Tableau 48 ci-dessous résume les pourcentages des flux dans chacun des cas.

Tableau 48: Les débits absolus et relatifs des sources par QoS dans les cas simulés.

	EF (Débit)	EF (%)	AF (Débit)	AF (%)	BE (Débit)	BE (%)
Cas 1	9.5	28.27	19.3	57.44	4.8	14.28
Cas 2	16.2	48.21	12.1	36.01	5.3	15.77
Cas 3	9.4	27.97	9.9	29.46	14.3	42.55

Pour le cas 2, (EF majoritaire), la matrice de trafic est donnée dans le Tableau 49.

Tableau 49: La matrice de trafic.

	Haut Débit			Bas Débit			UMTS			Total par Source	Débit du lien d'interconnexion
	EF	AF	BE	EF	AF	BE	EF	AF	BE		
Haut Débit	7.1	5.2	3.2	0.6	2	0.6	1.2	2.4	0.7	24.5	32
Bas Débit	1.8	0.8	0.5	0.3	0.2	0.2	0.4	0.2	0.1	3.0	6
UMTS	1.6	0.5		0.5	0.4		2.7	0.4		6.1	10
Total par destination	10.5	6.5	3.7	1.4	2.6	0.8	4.3	3	0.8	33.6	

Ce qui donne les débits en Mb/s des sources en fonction des classes :

Tableau 50: Les débits des sources par QoS.

	Débit total par classe de source			Débit total par source
	EF	AF	BE	
Haut Débit	8.9	9.6	4.5	23
Bas Débit	2.5	1.2	0.8	4.5
UMTS	4.8	1.3	0	6.1
Total par destination	16.2	12.1	5.3	33.6

Les abscisses des courbes font références à l'état du réseau de simulation : chaque état correspond à un ensemble de sources UMTS, Haut débit et Bas débits. Ces différents états sont regroupés dans le tableau ci-dessous.

Tableau 51: Les états simulés

	UMTS	Haut Débit	Bas Débit	Débit total
Etat 1	2	5	2	136.2 Mb/s
Etat 2	2	10	2	251.2 Mb/s
Etat 3	2	15	2	366.2 Mb/s
Etat 4	2	20	2	481.2 Mb/s

Ce qui donne les proportions suivantes des trafics :

Tableau 52: Proportion des trafics

	EF (%)	AF (%)	BE (%)	Débit total

Etat 1	43.39	38.91	17.69	136.2 Mb/s
Etat 2	41.24	40.20	18.55	251.2 Mb/s
Etat 3	40.44	40.68	18.86	366.2 Mb/s
Etat 4	40.02	40.93	19.03	481.2 Mb/s

Pour le cas 3, (BE majoritaire), les débits en Mb/s des sources en fonction des classes :

Tableau 53: Les débits des sources par QoS.

	Débit total par classe de source			Débit total par Source
	EF	AF	BE	
Haut Débit	4.5	5.8	12.4	24.5
Bas Débit	1.2	1.8	1.9	3.0
UMTS	3.7	2.3	0	6.1
Total par destination	9.4	9.9	14.3	33.6

La matrice de trafic est donnée dans le Tableau 54.

Tableau 54: La matrice de trafic.

	Haut Débit			Bas Débit			UMTS			Total par Source	Débit du lien d'interconnexion
	EF	AF	BE	EF	AF	BE	EF	AF	BE		
Haut Débit	7.1	5.2	3.2	0.6	2	0.6	1.2	2.4	0.7	24.5	32
Bas Débit	1.8	0.8	0.5	0.3	0.2	0.2	0.4	0.2	0.1	3.0	6
UMTS	1.6	0.5		0.5	0.4		2.7	0.4		6.1	10
Total par destination	10.5	6.5	3.7	1.4	2.6	0.8	4.3	3	0.8	33.6	

Les abscisses des courbes font références à l'état du réseau simulé : chaque état correspond à un ensemble de sources UMTS, Haut débit et Bas débits. Ils sont regroupés dans le tableau ci-dessous.

Tableau 55: Les états simulés

	UMTS	Haut Débit	Bas Débit	Débit total
Etat 1	2	5	2	135.3 Mb/s
Etat 2	2	10	2	248.8 Mb/s
Etat 3	2	15	2	362.3 Mb/s
Etat 4	2	20	2	475.8 Mb/s

Ce qui donne les proportions suivantes des trafics :

Tableau 56: Proportion des trafics

	EF (%)	AF (%)	BE (%)	Débit total
Etat 1	23.87	27.49	48.63	135.3 Mb/s
Etat 2	22.02	26.60	51.36	248.8 Mb/s
Etat 3	21.33	26.27	52.38	362.3 Mb/s
Etat 4	20.97	26.10	52.92	475.8 Mb/s

Pour alléger le document, seules les distributions du délai seront comparées dans les paragraphes ci-dessous. Le taux de perte et la gigue pour les cas 2 et 3 sont mis en annexe.

3.4.4.4.2. CB-WRR : Class Based WRR pour le cas 2

Les figures ci-après (Figure 44, Figure 45 et Figure 46) présentent les délais des flux EF, AF et BE pour le cas 2 (EF majoritaire) et pour les différents poids.

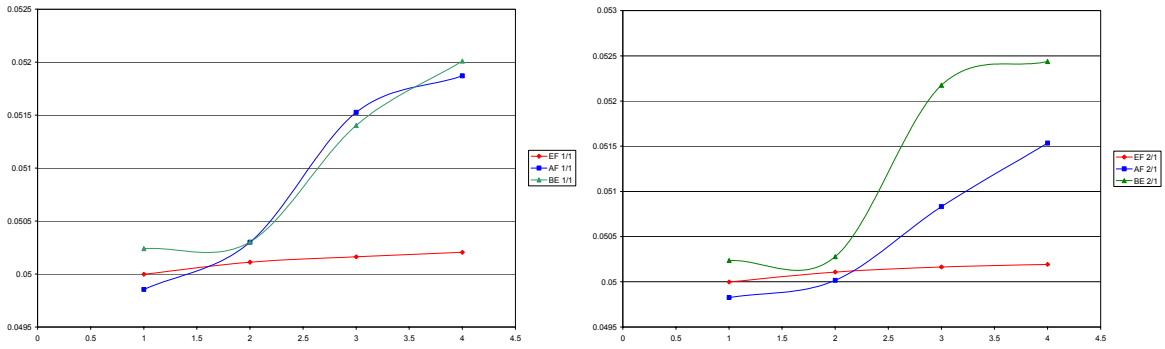


Figure 44: Délai des flux (1/1) et (2/1)

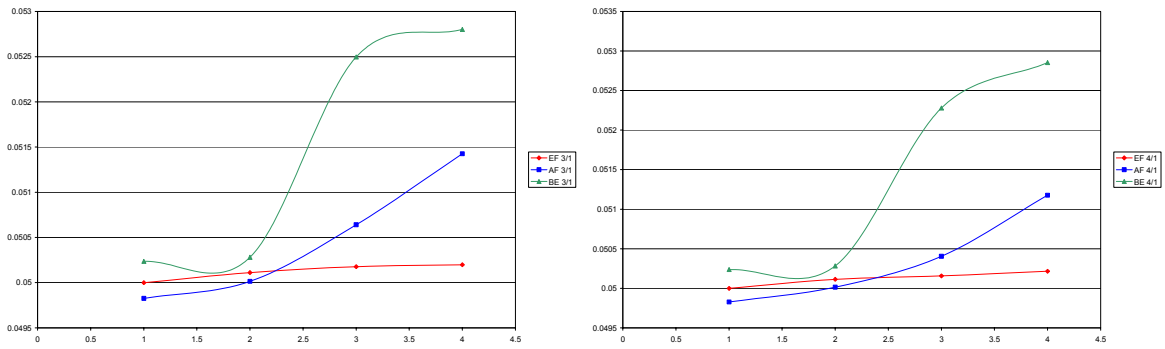


Figure 45: Délai des flux (3/1) et (4/1)

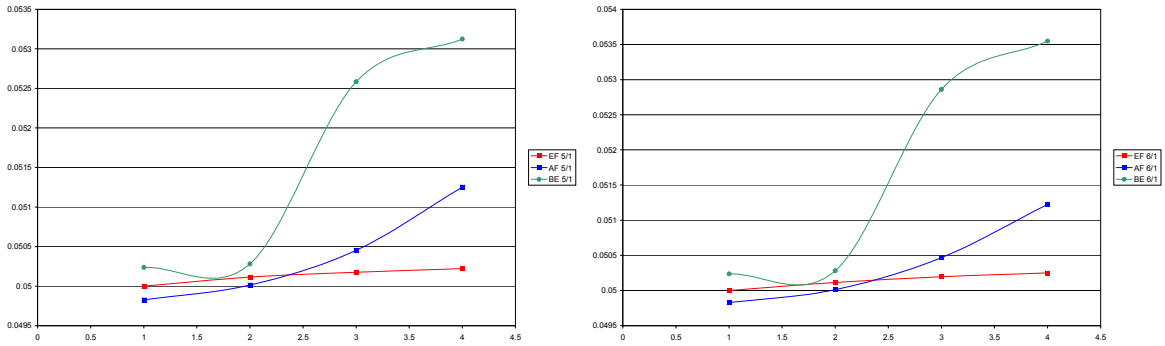


Figure 46: Délai des flux (5/1) et (6/1)

3.4.4.4.3. CB-WRR : Class Based WRR pour le cas 3

Les figures ci-après (Figure 47, Figure 48 et Figure 49) présentent les délais des flux EF, AF et BE pour le cas 3 (BE majoritaire) et pour les différents poids.

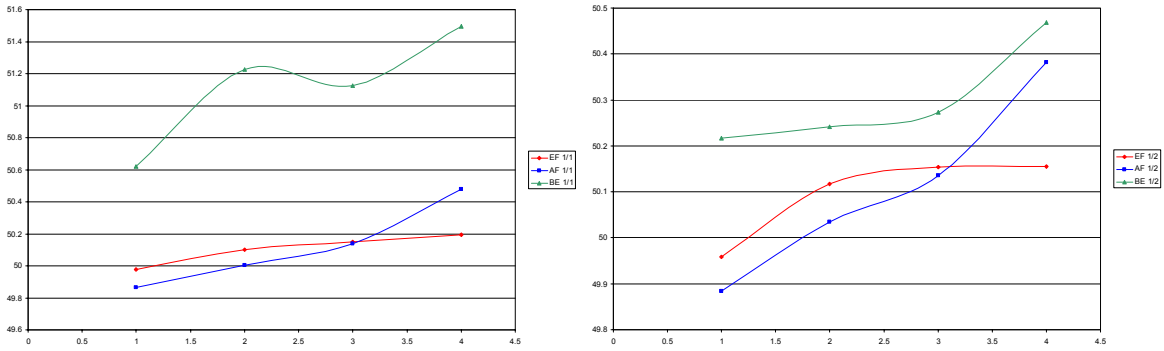


Figure 47: Délai des flux (1/1) et (1/2)

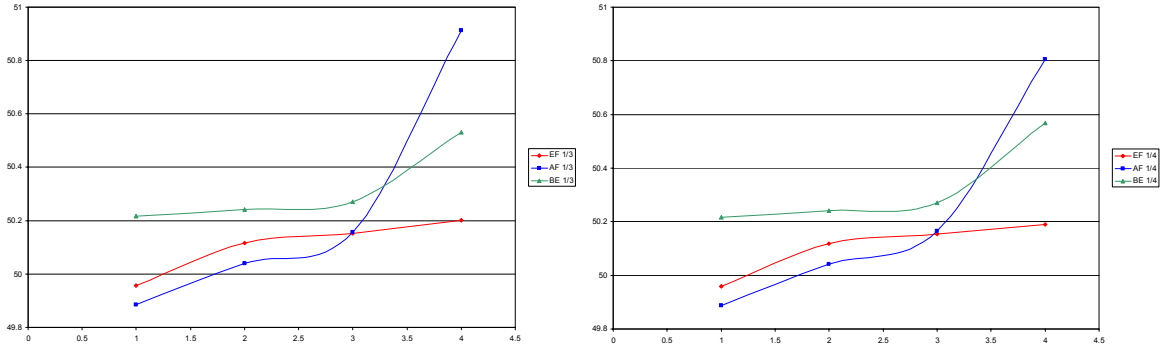


Figure 48: Délai des flux (1/3) et (1/4)

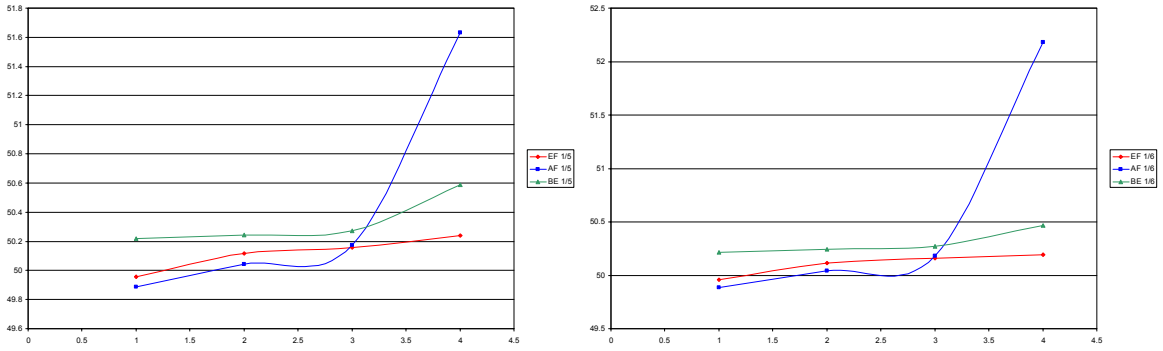


Figure 49: Délai des flux (1/5) et (1/6)

3.4.4.4.4. Interprétation des résultats

Les flux AF doivent être servis mieux que les flux BE avec un meilleur taux de perte et un meilleur délai. Or, plus le poids associé à AF est grand, mieux les flux AF seront servis, cependant les délais et le taux de perte des flux BE seront élevés. Si nous devons choisir parmi ces poids, nous opterons, pour le cas 2, un poids de 3/1 ou 4/1. Pour le scénario 3, nous opterons pour les poids 1/2 ou 1/3. Un poids supérieur pour les flux AF dégrade beaucoup les flux BE sans toutefois avoir un impact très favorable sur les connexions AF.

3.4.5. Interprétation des résultats

A partir des résultats dans les paragraphes ci-dessus et ceux fournis en Annexe 5 :, nous pouvons affirmer que pour offrir une bonne QoS aux flux EF, il vaut mieux leur donner une priorité absolue au niveau des routeurs. Pour garantir une certaine QoS aux flux AF et BE, il faudra veiller à ce que la charge du réseau introduite par les flux EF ne dépasse pas une certaine valeur. Cela est fait par un contrôle d'admission à l'entrée du réseau. Nous avons réalisé cela par le sceau à jetons lors du marquage des flux EF.

Pour les flux AF et BE qui n'ont pas de fortes contraintes de délai et dans le but de garder un taux de perte acceptable pour les deux types de flux, il est préférable de les différencier par un ordonnanceur à poids comme le WRR, le plus simple des ordonnanceurs.

D'une manière générale, le poids dépend fortement de la matrice de trafic. Dans les divers cas simulés ci-dessus qui donnent une priorité absolue pour les EF et afin de fixer les poids des ordonnanceurs pour les flux AF et BE, il faudra trouver le rapport des débits des flux entre ces deux classes. Il faudra alors fixer le poids en excès pour AF et en défaut pour BE afin de garantir une QoS des flux AF supérieure à celle offerte aux flux BE.

3.5. Conclusion

Dans ce chapitre, nous avons tout d'abord étudié analytiquement l'ordonnanceur GPS avec trois classes de service. Comme une solution analytique n'est pas tractable, nous avons validé notre étude par plusieurs batteries de simulations. A partir des différentes études que nous avons effectué au niveau du choix du poids de l'ordonnanceur, nous constatons que plusieurs aspects entrent en jeu: non seulement le taux d'arrivée de chaque file influe sur les caractéristiques de service des différents flux présents dans le réseau, mais aussi, la nature statistique du trafic modifie les valeurs moyennes obtenues pour chaque classe. Ainsi, si pour une raison ou pour une autre, les caractéristiques temporelles d'un flux change, la combinaison des poids qui garantissait la QoS recherchée peut ne plus satisfaire les applications.

Ce que nous pouvons proposer à un administrateur de réseau, c'est de toujours affiner les différents poids utilisés afin de garantir les différentes contraintes demandées. Pour garantir de flux EF subissant toujours les plus faibles délais de transit dans le cas de l'utilisation d'ordonnanceur avec poids, il faudra s'assurer que le poids relatif associé à cette classe est supérieur à la portion de la charge induite par cette classe.

CHAPITRE 4
Adaptation dynamique du chemin par la tomographie des réseaux

Ce chapitre propose un mécanisme d'adaptation dynamique du chemin utilisé afin de garantir l'utilisation de la route offrant la meilleure QoS entre deux extrémités du réseau. En envoyant des paquets d'une extrémité émettrice vers sa correspondante réceptrice et en appliquant l'algorithme ci-dessous, nous pourrons estimer la fonction de répartition du délai sur chaque tronçon des chemins observés. Nous pourrons alors déterminer la liaison qui offre les meilleures caractéristiques entre ces deux points. Le chemin choisi peut être celui qui offre le délai moyen le plus faible et/ou celui qui propose la valeur la plus faible de la probabilité de dépassement d'un seuil donné. Le premier schéma est utile pour le cas des liaisons entre des applications temps réels, comme le *streaming* d'audio ou de vidéo, qui nécessitent des délais faibles. Le second schéma est utile pour les applications nécessitant aussi des délais bornés, comme les conversations entre individus ou entre applications.

La motivation de ce travail réside dans la recherche du chemin qui offre la meilleure QoS au temps présent. Nous avons favorisé la vitesse de réactivité et la garantie des contraintes de QoS des connexions face au maintien des routes. Nous verrons plus loin dans ce chapitre que le facteur de mise à l'échelle de ce mécanisme est garanti grâce à l'utilisation du protocole MPLS de création des chemins: les tables de routage des nœuds intermédiaires ne sont pas affectées par les modifications requises par ce mécanisme.

4.1. La tomographie dans les réseaux

4.1.1. Introduction

Les technologies sous-jacentes hétérogènes et l'organisation non structurée d'Internet rendent le routage dynamique, la réservation de la bande passante, ou la détection d'anomalies des demandes difficiles à résoudre. Ces interrogations sont encore plus complexifiées par la coopération minimale des serveurs et des routeurs pour aider à la collecte des informations relatives aux différents flux et aux différentes connexions acheminés par le réseau. En outre, l'observation et l'inférence du réseau prennent un aspect similaire à d'autres problèmes d'observations où des valeurs clés du système ne sont pas observées. La reconstitution d'images et l'identification de systèmes sont des exemples similaires rencontrés dans le domaine du traitement du signal.

L'Internet des années 1970 a beaucoup évolué: d'un petit réseau contrôlé, utilisé par quelques scientifiques et militaires favorisés, il est devenu aujourd'hui un réseau immense formé d'un grand nombre de routeurs, de terminaux et de plateformes. Contrairement aux réseaux téléphoniques qui évoluent lentement et d'une manière très contrôlée, l'Internet évolue

rapidement et d'une façon chaotique. L'absence d'un contrôle centralisé et le caractère ouvert de son environnement, aboutissent à une problématique importante:

- Quelle est la connectivité du réseau?
- Quelles sont les bandes passantes des liaisons qui joignent telles extrémités?
- Quelles sont les performances du réseau en fonction de l'espace (sur chaque lien) et du temps?

Afin de déterminer les caractéristiques des réseaux, plusieurs utilitaires sont disponibles tel que *ping* et *traceroute*. Des informations relatives à ces utilitaires ainsi que des projets de détermination des caractéristiques des réseaux sont disponibles sur le site internet de CAIDA (Cooperative Association for Internet Data Analysis) [10]. Cependant, ces techniques offrent une vue partielle du réseau puisqu'elles n'effectuent pas de mesures quantitatives de performance. Comme il est indiqué précédemment, la nature décentralisée de l'Internet rend ces types de mesures difficiles: nous ne pouvons pas nous fier uniquement aux différents serveurs et routeurs pour qu'ils laissent passer les informations de statistiques relatives au réseau tel le débit des flux, le débit des liens ou le taux de perte. Ces différentes mesures imposent aux serveurs et routeurs internes des entêtes inacceptables en terme de temps de calcul, de débit de communication et parfois d'ajout de matériels. En plus, même si ces statistiques peuvent être collectées, un ISP (Internet Service Provider) peut les considérer comme secrètes, donc inaccessibles à un tiers. De plus, l'envoi de toutes les mesures à un centre de contrôle unique et centralisé, ajoute un trafic important qui pourrait introduire des délais et des congestions supplémentaires et nuisibles au bon fonctionnement du réseau.

Cette première partie présente les aspects d'observation et d'inférence dans un réseau tel que l'Internet. Par ***inférence***, nous différencions les méthodes utilisées ci-dessous, des méthodes de mesure directe qui donnent les statistiques du réseau (des mesures de proche en proche) ou celles qui comptent sur une coopération complète entre les différents routeurs. L'enjeu d'inférer un réseau à partir de ses extrémités est d'estimer un nombre de paramètres distribués dans l'espace et dans le temps e.g. le taux de perte par lien, la distribution du délai sur chaque lien, la connectivité dans le réseau ou même, le débit de chaque lien.

Les méthodes de mesure requièrent des logiciels pour d'une part, créer des générateurs de flux et d'autre part, des sondes pour capturer les paquets intéressants. De plus, ces logiciels implémentent des modèles statistiques des processus à mesurer et des stratégies d'échantillonnage afin de recueillir les données. Ces derniers, incluent des méthodes d'estimation basées moment ou basées vraisemblance, l'algorithme *Expectation-Maximization*, l'algorithme de *Monte Carlo par chaînes de Markov* (MCMC) et d'autres méthodes d'optimisations itératives.

Plusieurs scientifiques se sont intéressés à mesurer les paramètres internes d'un réseau et à caractériser mathématiquement ou statistiquement le comportement d'un réseau. Ils ont aussi développé des logiciels et des protocoles pour mesurer les délais des liens, ou pour détecter des intrus dans le réseau, ou aussi, pour isoler des erreurs dans les tables de routage ou des disfonctionnements sur un lien. Plusieurs outils se sont d'ailleurs développés pour mesurer activement ou passivement les réseaux. Ces outils requièrent une coopération supplémentaire entre les routeurs. Par exemple, dans les sessions RTCP (Real Time Control Protocol), des résumés des rapports émetteurs/récepteurs sur la gigue et le taux de perte, sont distribués à tous les participants à cette session [49]. Les outils de sondage actif tel que *ping*, *pathchar* (*pchar*), *clink* et *traceroute* mesurent les attributs de la couche transport des paquets. (voir [10] pour plus de détails). Par conséquent, dans le cas d'un grand réseau, où les routeurs ne coopèrent pas forcement, les méthodes de tomographie présentées ci-dessous, seront vraisemblablement les seules méthodes utilisables.

Avant d'entamer les détails, nous définissons la terminologie utilisée en l'illustrant sur la Figure 50 ci-dessous :

- Chaque nœud représente un ordinateur, un routeur ou un sous réseau (formé de plusieurs routeurs/ordinateurs).
- Une connexion entre deux nœuds est appelée chemin.
- Chaque chemin est composé d'un ou de plusieurs liens, une connexion directe entre deux nœuds intermédiaires. Un lien peut représenter une chaîne de liens physiques connectés par des routeurs intermédiaires.

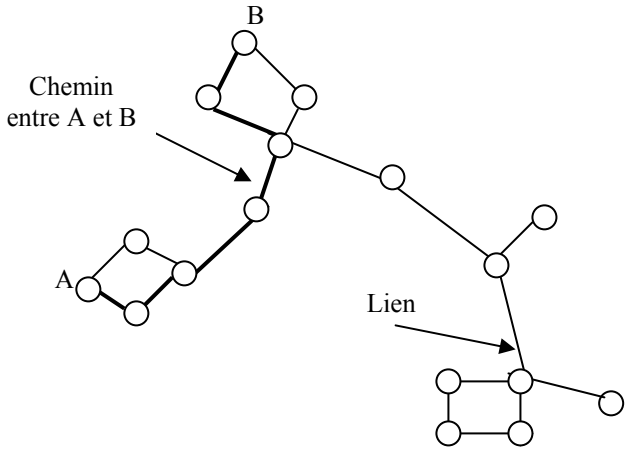


Figure 50: Noeuds, liens et chemins dans un réseau.

4.1.2. L'inférence au niveau lien et au niveau émetteur/récepteur

L'inférence dans les réseaux peut être classifiée en fonction du type de donnée recherché et des paramètres à estimer. Deux sortes d'estimations par la tomographie sont décrites dans la littérature:

1- Estimation des paramètres au niveau des liens en se basant sur des mesures de bout en bout des chemins : les observations aux extrémités du réseau déterminent la charge intérieure au réseau. Ces méthodes d'estimations sont utilisées pour déterminer les performances sur une route afin d'éviter les congestions.

2- Estimation de l'intensité de trafic émetteur/récepteur sur un chemin en se basant sur des mesures de performance des liens. C'est en observant sur chaque nœud le trafic provenant d'un lien et routé vers un autre lien que nous pourrons déterminer le débit des flux de chaque source vers chaque destination. Ces méthodes sont utilisables par un administrateur de réseau afin de déterminer la matrice de trafic de son réseau.

Le système d'équation des deux types de tomographie peut s'écrire:

$$Y = A.\theta + \varepsilon \qquad \textbf{(Équation 1)}$$

avec:

Y, le vecteur des mesures e.g. le nombre de paquets, le délai de bout en bout, pris à plusieurs endroits.

A, la matrice de routage (donc la topologie est connue).

θ, le vecteur des paramètres du réseau e.g. le délai moyen, le log de la probabilité de transmission sur le lien ou le vecteur de trafic origine - destination (OD).

ε, le bruit.

Typiquement, A est une matrice binaire : l'élément (i, j) est soit un *zéro* soit un *1*. Il décrit la topologie du réseau, si une liaison entre i et j existe. Un des problèmes de l'inférence est d'inverser la matrice A qui peut être pour un grand réseau de taille énorme. Ce problème est traité dans la littérature du traitement du signal [48], statistique [46] et mathématique appliquée [24]. La solution de l'inversion de la matrice A dépend de la nature du bruit ε. Généralement, les composants de ε sont supposés être des variables aléatoires indépendantes et identiquement distribuées suivant des processus gaussiens, poissonniens, binomiales ou multinomiales.

Quand le bruit est gaussien avec une covariance indépendante de Aθ, des méthodes comme le moindre carré récursif linéaire peuvent être utilisées en utilisant le gradient conjugué, Gauss-Seidel et d'autres méthodes itératives. Quand le bruit est poissonnien, binomial ou multinomial, des méthodes statistiques plus sophistiquées sont utilisées, comme le maximum de vraisemblance via EM et maximum à posteriori via Monté Carlo par Chaîne de Markov.

Les exemples ci-dessous présentent quelques applications à l'équation ci-dessus :

1. Soit θ le logarithme de la probabilité de réussite sur chaque lien. Y, le logarithme de la probabilité de réussite source-destination est égale à Aθ, avec A la matrice binaire de routage décrite plus haut. En supposant un nombre connu de paquets envoyés par chaque source vers chaque destination, un modèle de mesure binomiale peut être adopté [40]. Quand le nombre envoyé est très grand, ce modèle peut être réduit à un modèle gaussien, ce qui rend le système un modèle linéaire classique.
2. Supposons que les délais source-destination de bout en bout sont mesurés. Nous voulons maintenant estimer la distribution de la probabilité du délai sur chaque lien. Dans ce cas, θ sera le vecteur composé de la fonction génératrice cumulée de la densité du délai sur chaque lien, et Y est toujours liée à θ par $Y=A\theta+\varepsilon$.
3. Dans le cas de l'estimation de la matrice de trafic source-destination, Y mesure le nombre de paquets sur chaque lien et θ est l'intensité du trafic source-destination. Avec des suppositions gaussiennes sur la variance et la moyenne du trafic source-destination [11], l'équation liant Y à θ devient linéaire:

$$Y=A\theta.$$

Dans les trois cas précédents, le bruit ε est corrélé et peut avoir une covariance dépendante de A et/ou de θ, ce qui donne un problème d'inférence moins trivial. De plus, dans plusieurs cas, la quantité limitée de données rend l'approximation gaussienne non applicable, et des modèles discrets (e.g. binomiaux) peuvent être plus adéquats pour décrire la nature de donnée discrète et paquétisée. Ces modèles nécessitent des moyens plus développés pour déduire l'inférence tel que l'algorithme EM ou un modèle de simulation Monté Carlo.

Généralement, tous les éléments dépendent du temps:

$$Y_t = A_t\theta_t + \varepsilon_t. \quad \text{où t représente le temps.}$$

En effet, le scénario dépendant du temps représente plus fidèlement la dynamique du réseau réel. [41] et [30] décrivent des méthodes pour suivre la variation du réseau comme par exemple le filtre de kalman.

Une autre variante de l'équation de Y suppose que la matrice de routage A n'est pas totalement connue. Cela nous conduit à l'identification de topologie [53], [44] et [45].

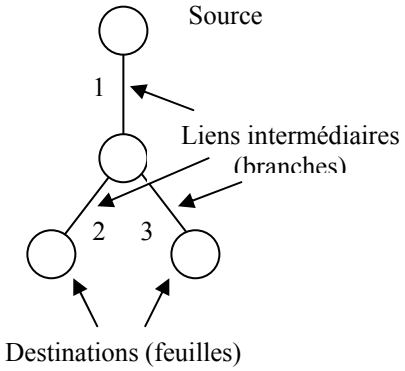


Figure 51: Exemple de tomographie dans un réseau.

Comme notre contribution se situe dans le domaine de l'inférence au niveau lien, nous exposerons ci-dessous, plus en détail ce type de tomographie. Le projet MINC [42] est le pionnier de l'utilisation du multicast pour l'inférence des caractéristiques du réseau, produisant un grand travail dans ce domaine.

Considérons le cas de la Figure 51: le comportement de bout en bout peut être inféré par une mesure synchronisée entre les émetteurs et les récepteurs. Les émetteurs peuvent déterminer si un paquet a atteint sa destination ou non, e.g. il a été perdu. Dans le cas d'un paquet reçu par la destination, le délai pourra être calculé. La source n'a pas les moyens de déterminer directement le lien sur lequel le paquet a été perdu, ni le délai sur chaque lien.

Chaque nœud est numéroté, $j= 0..m$ et chaque lien prend le numéro du nœud en aval. Soit N, le nombre de mesures distinctes sur les chemins du réseau (d'une source vers une destination). Soit a_{ij}, la probabilité que la $i^{ème}$ mesure passe par le lien j. En principe, a_{ij} vaut zéro ou 1. Cependant, pour garder la généralité de l'étude, il est utile de prendre en compte le routage adaptatif qui donne une valeur réelle aux différentes a_{ij} comprises entre 0 et 1. Soit A la matrice de routage d'élément a_{ij}. Les lignes de A correspondent au chemin de la source vers la destination et les colonnes de A correspondent aux liens individuels sur ces chemins.

La Figure 51 illustre un réseau simple formé d'une source (source zéro), d'un nœud intermédiaire (nœud 1) et de deux destinations (nœuds 2 et 3). Des mesures de bout en bout sont possibles, i.e. les chemins (0,2) et (0,3). La matrice de routage A s'écrit alors:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Plusieurs suppositions sont faites afin d'appliquer des méthodes mathématiques:

- La matrice de routage est supposée connue et constante sur la durée des mesures.
- Les performances statistiques des différents liens sont supposées indépendantes.

Deux méthodes de communications existent: les réseaux multicast et unicast. Nous développerons leurs aspects liés à la tomographie dans le paragraphe qui suit.

4.1.3. L'avantage du multicast

La tomographie multicast [47] a été la première approche d'inférence dans les réseaux. Nous voulons estimer le taux de perte dans le réseau de la Figure 51. Si un paquet multicast est envoyé par la source 0, puis est reçu par le nœud 2 non par le nœud 3, alors nous pouvons immédiatement conclure que la perte a eu lieu sur le lien 3. En effectuant de telles mesures,

nous pouvons déterminer le taux de perte sur les liens 2 et 3. Une fois ces estimations faites et en utilisant les mesures déjà effectuées, nous pourrons alors déterminer le taux de perte au niveau du lien 1.

Pour illustrer, soient θ_1, θ_2, et θ_3 les logarithmes des probabilités de succès des 3 liens. Soit $p_{2/3}$ le quotient du nombre de paquets reçus simultanément par les destinations 2 et 3 sur le nombre de paquets reçus uniquement par l'extrémité 3. $p_{2/3}$ est alors la probabilité empirique de succès du lien 2 conditionnellement au succès du lien 3 qui donne une estimation de θ_2. Nous définissons de même $p_{3/2}$. p_2 et p_3 sont les probabilités de réception des paquets sur le nombre total de paquets émis respectivement sur les liens 2 et 3.

Nous écrivons alors:

$$\begin{pmatrix} \log p_2 \\ \log p_3 \\ \log p_{2/3} \\ \log p_{3/2} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix}$$

D'autres algorithmes pour les réseaux plus complexes sont développés dans [47] et [40].

Des procédures similaires sont applicables pour obtenir le délai en supposant que le délai de propagation des liens est connu. En envoyant un paquet multicast à 2 et 3 nous pouvons déterminer le délai sur les deux chemins. Le délai est le même pour les deux destinations sur le tronçon 0-1. La différence est celle des délais sur 1-2 et 1-3.

La détermination de la tomographie par l'unicast est plus compliquée mais est plus réaliste du fait que tous les réseaux n'implémentent pas le multicast. L'idée est que si deux paquets consécutifs sont émis à deux destinations distinctes, ils auront le même délai sur le tronçon commun avant de se séparer. La différence est donc sur les liens qui ne sont pas en commun.

4.2. Méthodes d'estimation du délai

Après cet aperçu des différentes techniques et méthodes utilisées par la tomographie des réseaux, nous présenterons dans les paragraphes suivants un algorithme d'estimation du délai que nous avons ensuite amélioré pour être implémentable dans un réseau réel.

4.2.1. L'estimation complète

L'estimation proposée dans [41], basée sur l'algorithme EM (Expectation Maximisation) suit les étapes décrites ci-dessous.

Par souci de clarté des notations, nous supposerons un schéma de réseau avec une source et uniquement deux destinations (cf. Figure 52). Cet algorithme reste applicable avec un nombre de destination plus grand. Nous noterons $y_1(k)$ et $y_2(k)$ les délais recueillis aux deux extrémités du réseau à l'instant k, $k \in \{1, 2, \ldots, K\}$. Le délai sur chaque lien est discrétisé en L valeurs $\{d_0, d_1, \ldots, d_L\}$. L'algorithme estime l'ensemble des probabilités $p_i = \{p_{i,0}, p_{i,1}, \ldots, p_{i,L}\}$ sur chaque lien $i \in \{1, 2, 3\}$.

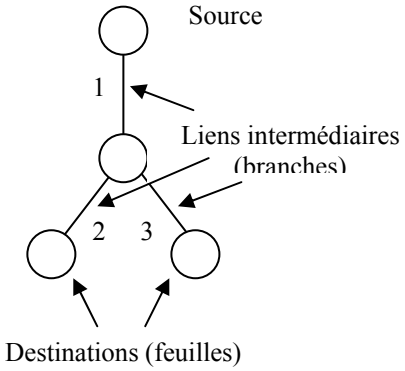


Figure 52: Schéma de l'étude.

Soit $y=\{y_1(k), y_2(k)\}$, la paire des délais observés à l'instant k. Pour déterminer $p=\{p_i\}$ nous utiliserons une estimation du maximum de vraisemblance (MLE) en se basant sur les y. La vraisemblance de chaque paire de mesure, qui est notée $l(y/p)$, est une convolution des probabilités sur chaque lien.

Soit $z_i(k)$ la valeur du délai sur le lien i à l'instant k. Nous noterons $z_i=\{z_i(k)\}$ l'ensemble des délais sur un lien donné, et $z=\{z_i\}$, l'ensemble des délais dans tout le réseau. Les valeurs de z ne sont pas ***observées***. Nous définissons les données complètes $x=\{y,z\}$. La vraisemblance de x peut s'écrire:

$$l(x/p) = f(y/z)g(z/p)$$

où f est la probabilité conditionnelle de y sachant z et g la vraisemblance de z sachant p. Or la détermination de z entraîne celle de y. Or comme la fonction $f(y/z)$ n'est pas liée à p, pour maximiser la vraisemblance $l(x/p)$, il suffit de maximiser $g(z/p)$ qui s'écrit:

$$g(z/p) = \prod_{i,j} p_{i,j}^{m_{i,j}}$$

avec $m_{i,j}$ le nombre total de paquets qui ont transité sur le lien i et subi un délai j.

$$m_{i,j} = \sum_{k=1}^{N} 1_{z_i(k)=j}$$

La vraisemblance complète s'écrit donc:

$$l(x/p) \propto \prod_{i,j} p_{i,j}^{m_{i,j}}$$

Les valeurs $p_{i,j}$ qui maximisent la vraisemblance correspondent au rapport du nombre de paquets passant par le lien i avec un délai j sur le nombre total de paquets traversant le lien i:

$$\hat{p}_{i,j} = \frac{m_{i,j}}{\sum_{l=1}^{L} m_{i,l}}$$

Si les différentes valeurs de z étaient connues, la maximisation serait triviale. Cependant, ce sont des valeurs non observées. Or, l'algorithme itératif EM résout le problème en alternant entre le calcul, d'une part, de la moyenne du logarithme de la vraisemblance complète ayant les observations y (Etape E, Expectation) et, d'autre part, en maximisant cette quantité sur p (Etape M, Maximisation).

Le logarithme de la vraisemblance complète s'écrit:

$$\log l(x/p) \propto \sum_{i,j} m_{i,j} \log p_{i,j}$$

Donc, l'étape E consiste à calculer la moyenne de $m=\{m_{i,j}\}$ ayant les observations y. Cette moyenne est:

$$E_{p^l}\left[m_{i,j} / y\right] = E_{p^l}\left[\sum_{k=1}^{N} 1_{z_i=j} / y\right] = \sum_{k=1}^{N} E_{p^l}[1_{z_i=j} / y_1(k), y_2(k)]$$

$$E_{p^l}\left[m_{i,j} / y\right] = \sum_{k=1}^{N} p^l\left(z_i(k) = j / y_1(k), y_2(k)\right)$$

En d'autres termes, la valeur moyenne de $m_{i,j}$, pour une distribution de probabilité des délais à l'itération l de p^l, sachant l'observation y, est égale à la somme, sur tous les échantillons, des probabilités que le délai de la $k^{ème}$ observation est, sur le lien i de j connaissant les deux observations $y_1(k)$ et $y_2(k)$.

Comme toute méthode itérative, nous commençons par une probabilité $p_{i,j}$ quelconque de la distribution du délai. Ici, nous avons pris une distribution uniforme sur les l valeurs possibles du délai. Puis, à partir de l'échantillon, nous procédons à une optimisation des probabilités. L'algorithme Forward – Backward ou Upward – Downward [9] permet de déduire $z_i(k)$ grâce aux valeurs de $y_1(k)$ et $y_2(k)$.

L'étape M de maximisation de la probabilité effectue le calcul basé sur l'expression $\hat{p}_{ij} = \frac{m_{ij}}{\sum_{l=1}^{L} m_{il}}$ en remplaçant $m_{i,j}$ par la valeur trouvée à l'étape E. La distribution p^{l+1} est réinjectée dans l'algorithme jusqu'à aboutir à la convergence avec la précision voulue.

4.2.2. Les limitations de l'algorithme

Généralement, l'inférence des réseaux permet uniquement de différencier 2N-1 chemins, avec N le nombre d'extrémités - feuilles. Les figures ci-dessous (Figure 53, Figure 54 et Figure 55) illustrent quelques cas.

Cette méthode d'estimation nécessite une synchronisation temporelle entre l'émetteur et les différents récepteurs. Cela peut se faire, par exemple, grâce à une synchronisation GPS. Dans le cas de notre mécanisme, il y aura uniquement un seul récepteur quelque soit le nombre de routes observées. De ce fait, le coût de l'installation restera le même tout au long de l'existence du réseau.

Afin de prendre en compte l'aspect actuel du trafic, nous avons développé cet algorithme pour traiter un nombre constant d'échantillons: nous aurons donc la distribution de la probabilité sur le dernier intervalle de temps T donné. En fonction des ces valeurs, nous déciderons de maintenir ou d'effectuer une modification du chemin actuel utilisé pour acheminer le flux utile.

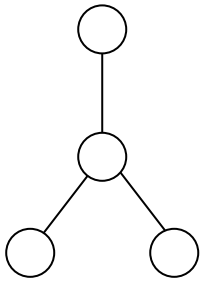


Figure 53: Deux feuilles, trois branches.

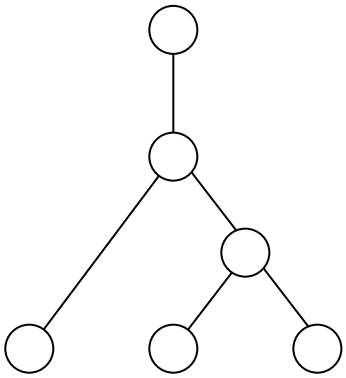


Figure 54: Trois feuilles, cinq branches

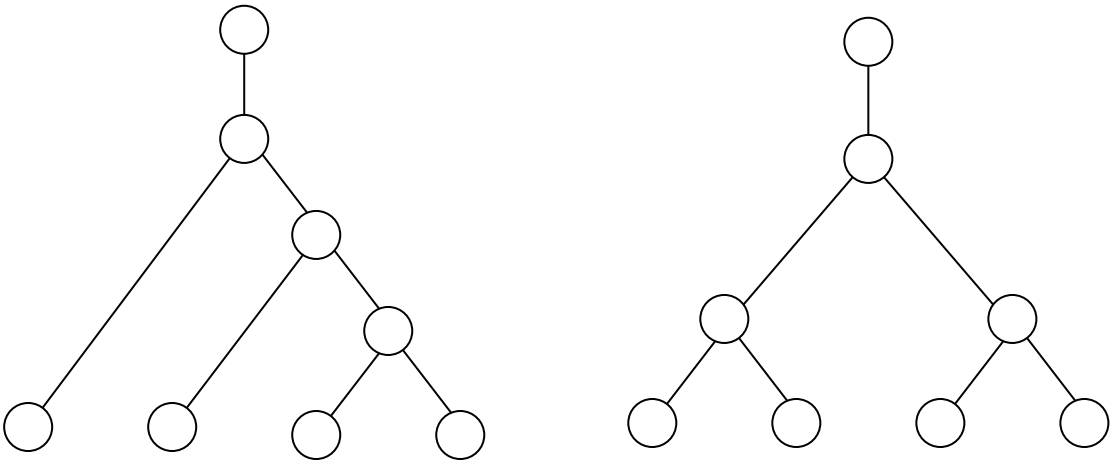


Figure 55: Quatre feuilles, sept branches

Cependant, dans une configuration de détermination de la probabilité du délai à des intervalles de temps constants, cet algorithme présente un inconvénient: il ne tient pas compte dans le calcul à un instant donné des caractéristiques aux intervalles précédents bien qu'il y ait un certain nombre de paquets appartenant aux différents intervalles.

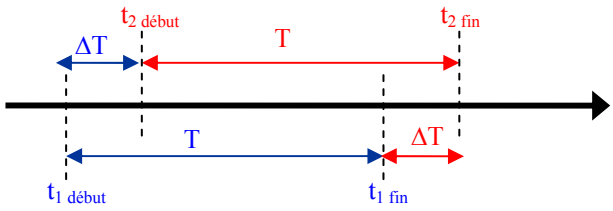


Figure 56: Les intervalles d'estimation.

Considérons à présent les deux intervalles de la Figure 56 I_1 et I_2 de durée T, espacés de ΔT. Pour l'estimation de la probabilité de I_2, nous pouvons décomposer les couples des valeurs allant de $t_{1\ début}$ à $t_{2\ fin}$ en 3 parties:

- Les échantillons de ΔT allant de $t_{1\ début}$ à $t_{2\ début}$ ne sont plus utiles.
- Ceux de $t_{2\ début}$ à $t_{1\ fin}$ ont fait l'objet de l'estimation de I_1 et feront aussi partie des valeurs de I_2.

- Ceux allant de $t_{1\ fin}$ à $t_{2\ fin}$ sont nouveaux: ils n'ont pas été pris en compte dans le calcul de la probabilité sur I_1.

En profitant du résultat obtenu à $t_{1\ fin}$, nous "échangeons" les délais de $t_{1\ début}$ à $t_{2\ début}$ par ceux de $t_{1\ fin}$ à $t_{2\ fin}$: c'est l'objet de notre proposition d'estimation simplifiée beaucoup plus rapide en terme de complexité de calcul.

4.2.3. L'estimation simplifiée

A partir de la dernière constatation et en tirant profit de la stabilité statistique du réseau, nous procédons comme suit: après chaque insertion de mesure, i.e. génération du couple $y = \{y_1(k_{t_1\ fin}+1), y_2(k_{t_1\ fin}+1)\}$, nous retirons l'échantillon correspondant à $t = t_{1\ début}$ de sorte à toujours obtenir le même nombre d'échantillons *N_ech*.

Notons $E_p{}^T[m_{ij}/y]$, la valeur moyenne des $m_{i,j}$ connaissant les observations sur l'intervalle de temps de durée T=[$t_{1\ début}$, $t_{1\ fin}$]; et $E_p{}^{T+1}[m_{ij}/y]$ la moyenne des $m_{i,j}$ sur la fenêtre temporelle (T+1), correspondant aux échantillons $y = \{y_1(k_{t_1\ début}+1), y_2(k_{t_1\ début}+1)\}$ jusqu'à $y = \{y_1(k_{t_1\ fin}+1), y_2(k_{t_1\ fin}+1)\}$. Les deux fenêtres ont une même durée T. Le passage de $E_p{}^T[m_{ij}/y]$ à $E_p{}^{T+1}[m_{ij}/y]$ se fait comme suit:

D'une part:

$$E_p{}^T[m_{ij}/y] = \sum_{k=1}^{N} p(z_i(k) = j / y_1(k), y_2(k))$$

D'autre part:

$$E_p{}^{T+1}[m_{ij}/y] = \sum_{k=2}^{N+1} p(z_i(k) = j / y_1(k), y_2(k))$$

En supposant que la variation des probabilités est relativement faible, nous insérons $E_p{}^T[m_{ij}/y]$ dans $E_p{}^{T+1}[m_{ij}/y]$. Nous pouvons alors écrire:

$$E_p{}^{T+1}[m_{ij}/y] = \sum_{k=1}^{N} p(z_i(k) = j / y_1(k), y_2(k)) - p(z_i(1) = j / y_1(1), y_2(1)) + p(z_i(k+1) = j / y_1(k+1), y_2(k+1))$$

Et nous obtenons :

$$E_p{}^{T+1}[m_{ij}/y] = E_p{}^T[m_{ij}/y] - p(z_i(1) = j / y_1(1), y_2(1)) + p(z_i(k+1) = j / y_1(k+1), y_2(k+1))$$

Ensuite, nous remplacerons dans $\hat{p}_{ij} = \dfrac{m_{ij}}{\sum_{l=1}^{L} m_{il}}$ la valeur $m_{ij} = E_{p'}[m_{ij}/y]$.

Par cette méthode, chaque estimation pour une fenêtre T se résume par la détermination de $m_{i,j}$, qui se réduit à une opération simple de soustraction d'un seul élément et de l'ajout d'un seul élément. Cette opération est répétée $i \times j$ fois, pour toutes les valeurs de *i* (nombre de tronçons) et de *j* (nombre d'intervalles d'estimations).

La seule limitation de ce nouvel algorithme est la nécessité d'une variabilité réduite dans le réseau: dans le cas d'une mesure $y = \{y_1(k_{t_1\ fin}+1), y_2(k_{t_1\ fin}+1)\}$ qui correspond à une

probabilité $p_{i,j}$ nulle, le calcul de $p(z_i(k+1)=j/y_1(k+1),y_2(k+1))$ sera nul aussi, ce qui engendrera une erreur dans les estimations. La solution que nous avons adoptée pour résoudre ce problème est d'effectuer une estimation complète à chaque fois que nous rencontrerons cette limitation. Le mécanisme développé ci-dessous prend en compte cet aspect.

4.3. Implémentation

Afin de garder une stabilité dans les routes tout en minimisant une éventuelle oscillation entre les chemins empruntés, le mécanisme ne modifiera pas le chemin tant que la différence de performance entre la nouvelle route et celle utilisée actuellement reste inférieure à un seuil défini. Nous avons opté pour une valeur de 5 % pour le délai moyen et de 3% pour la probabilité de dépassement du seuil.

Nous avons utilisé le protocole MPLS pour implémenter le mécanisme dans notre réseau de validation. Il favorise le facteur d'échelle du mécanisme vu que les tables de routage des routeurs internes dans le cœur du réseau ne seront point affectées dans le cas d'une modification du chemin utilisé. C'est uniquement à l'entrée des LSP, au niveau de l'affectation des FEC aux flux entrants, qu'il y aura une modification : les paquets issus d'une connexion seront acheminés via un autre chemin – un autre LSP. De même, avec MPLS, il n'y aura pas de temps d'attente pour propager les modifications sur les différents routeurs des routes en question: les différents chemins étant déjà établis et maintenus tout au long de l'existence du flux à optimiser.

Même si nous n'avons pas traité cet aspect dans notre étude, ce mécanisme peut être cependant utilisé pour observer des chemins avec plusieurs QoS. En supposant que le réseau implémente DiffServ, nous pourrons estimer p_{i,j_c}, la probabilité du délai pour chaque classe *c*. En combinant MPLS et DiffServ, nous aurons la possibilité d'acheminer les flux par plusieurs LSP, en faisant la correspondance (***mapping***) entre la QoS demandée et celle disponible sur les différents liens.

Puisque notre estimation propose d'adapter le chemin emprunté par les paquets d'un flux, il faudra estimer la QoS des liens sur chaque intervalle de temps ΔT. D'une part, pour une meilleure estimation, il faudra prendre un échantillon de longue durée. D'autre part, il faudra estimer le plus vite possible pour modifier la route dès que le lien devient chargé. Pour prendre en considération ces deux contraintes dans notre modèle, nous avons introduit deux intervalles de temps:

- T, la durée de la fenêtre temporelle qui détermine le nombre d'échantillons.
- ΔT, la différence de temps entre deux estimations.

En prenant ces deux fenêtres temporelles, nous avons pu dissocier ces deux valeurs.

Après avoir décrit les différents algorithmes utilisés ainsi que les techniques mises en place pour implémenter le mécanisme d'adaptation du chemin, nous présentons dans le paragraphe 4.4 la validation que nous avons effectuée.

4.4. Validation

Nous avons procédé à une validation sur un réseau des deux algorithmes présentés précédemment. Elle a été conduite avec le simulateur événementiel NS (Network simulator). Dans toutes les courbes présentées, nous avons fixé la durée de l'intervalle de temps T de l'estimation à 20 secondes pour les deux méthodes d'estimation. De plus, chaque ΔT de 4 secondes, nous opérons à une réévaluation de l'estimation en maintenant une convergence relative de 1% tout au long des estimations itératives. La convergence est supposée atteinte quand $\frac{p_{i,j}^{l+1} - p_{i,j}^{l}}{p_{i,j}^{l+1}} < 0{,}01$ pour toutes les valeurs de i et j.

4.4.1. Le réseau expérimental

Nous avons mis en place le réseau de la Figure 57. Deux chemins entre la **Source** S et la **Destination** D sont observés. Le premier passe par le routeur C_2, tandis que le second passe par C_3.

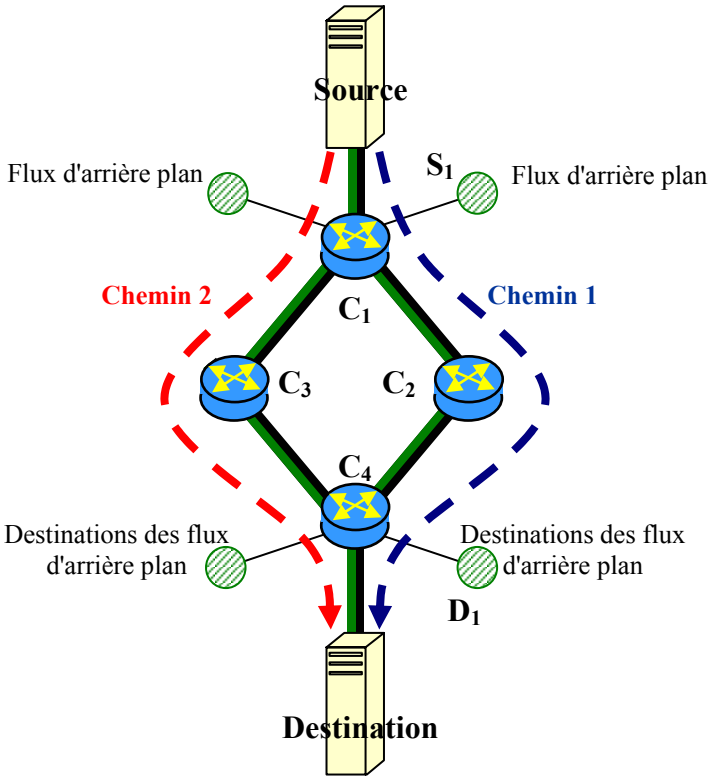


Figure 57: En fonction de la QoS estimée, le chemin 1 ou 2 sera utilisé pour acheminer les paquets du flux source.

Le Tableau 57 résume les caractéristiques des différents liens dans le réseau de la Figure 57. Les liens C_1 - C_2, C_1 - C_3, C_2 - C_4 et C_3 - C_4 sont les goulots d'étranglement dans ce réseau.

Tableau 57: Les caractéristiques des liens utilisés dans le schéma de validation.

	Débit du lien (Mbit/sec)	Délai introduit par le lien (ms)
$S - C_1$	10	10
$C_1 - C_2$	2	10
$C_1 - C_3$	2	10
$C_2 - C_4$	2	10
$C_3 - C_4$	2	10
$C_4 - D$	10	10
$S_1 - C_1$	20	10
$D_1 - C_4$	20	10

Les caractéristiques statistiques des différents flux sources de trafic sont résumées dans le Tableau 58:

Tableau 58: Les caractéristiques des différents flux.

Source	Chemin suivi	Trafic de type	Taille moyenne des paquets (en octets)	Débit de la source (kbits/sec)
t_1	$S \rightarrow D$	Constant	500	20
t_2	$S \rightarrow D$	Constant	500	20
S	$S \rightarrow D$	Exponentiel	500	100
S_1	$S_1 \rightarrow D_1$ *(suivant le chemin 1)*	Exponentiel	500	300
S_2	$C_1 \rightarrow C_4$ *(suivant le chemin 1)*	Markovien	4250	300
S_3	$C_1 \rightarrow C_4$ *(suivant le chemin 2)*	Markovien	750	1300

Le trafic noté *Exponentiel* admet un temps inter paquet qui suit une distribution exponentiellement distribuée et une taille des paquets constante (Nous utiliserons les formules de file d'attente M/D/1 avec ce type de trafic).

Dans le cas d'une notation *Markovienne*, le temps inter arrivé des paquets suit une distribution exponentiellement distribuée et la taille des paquets suit aussi une distribution exponentiellement distribuée de valeur moyenne 4250 octets pour le flux S_2 et 750 octets pour le flux émis S_3.

S_1, S_2 et S_3 sont tous les trois des flux d'arrière-plan seront présents au fur et à mesure que la simulation avance. Ils chargeront les chemins de C_1 à C_4. Les deux premiers flux passent par la route 1, alors que le troisième charge le deuxième chemin.

Initialement, seul le flux émis par S_1 existe: il charge le chemin 1 afin que la route C_1-C_3-C_4 soit mieux servie que la route C_1-C_2-C_4.

A l'instant t= 50,6 secondes, le flux S_3 s'active sur le lien C_1-C_3-C_4. Cela entraînera un délai moyen plus important sur la partie C_1-C_3-C_4, que sur la route passant par C_1-C_2-C_4.

Ainsi, le mécanisme mis en place modifiera le chemin afin de prendre la route la moins chargée, en l'occurrence C_1-C_2-C_4.

A l'instant t= 125 secondes, le trafic S_1 est arrêté tandis que S_2 est activé, générant des paquets de grandes tailles. La probabilité que le délai dépasse τ (le pourcentage des paquets arrivant à destination ayant subi un délai supérieur à τ) devienne élevée, ce qui entraîne que la QoS de la connexion S-D n'est plus vérifié. Cette dernière requiert que la probabilité du délai supérieur à τ soit inférieure à ε. Un changement de chemin est effectué: le flux observé repasse par C_1-C_3-C_4. Il faut savoir que le délai introduit par S_2 sur le tronçon C_1-C_2-C_4 est égal à celui introduit par S_3 sur le tronçon C_1-C_3-C_4.

La Figure 58 illustre temporellement le déroulement de la simulation.

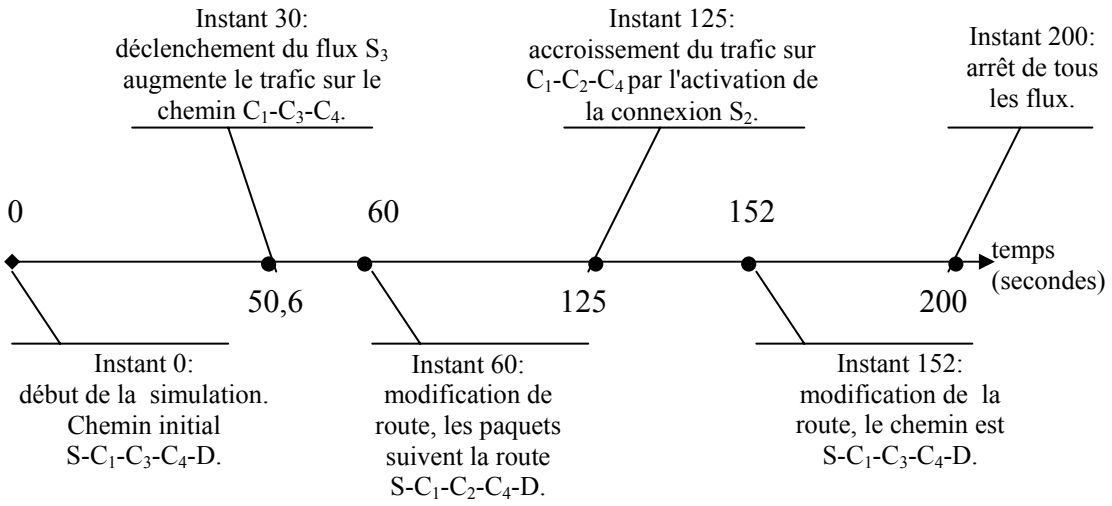


Figure 58: Déroulement temporel de l'experimentation.

Il est à noter que les flux de mesure tomographique et le flux principal devront avoir la même distribution de la taille des paquets (au moins la même taille moyenne). Cette condition garantit que le délai subi sur les liens et dans les routeurs par ces différents flux est approximativement le même, et que la différence provient uniquement de l'attente de l'écoulement des paquets circulants sur les liaisons.

4.4.2. Comparaison des mesures et des estimations

Dans les différentes courbes ci-dessous, l'expression "***tronçon 1***" désigne la connexion C_1-C_2-C_4, et "***tronçon 2***" désigne C_1-C_3-C_4. Pour une meilleure clarté des courbes, l'estimation de ces deux tronçons sera uniquement illustrée. Les délais des liens **Source** - C_1 et C_4 -**Destination** n'évoluent pas vu la charge constante qu'ils transitent tout au long de la simulation.

En appliquant le délai moyen d'une file M/D/1:

$$E[Y] = E[X] + \frac{\rho . E[X]}{2(1-\rho)}$$

En utilisant les valeurs numériques, nous obtenons:

$$E[Y] = \frac{500 \times 8}{10000000} + \frac{\frac{140}{1000} \times \frac{500 \times 8}{10000000}}{2(1 - \frac{140}{1000})} = 0{,}403 \text{ msecondes}$$

Il faudra ici ajouter le délai de transit du lien, qui est de 10 msecondes.

La Figure 59 montre les différentes fonctions de répartition du délai sur les deux tronçons étudiés à t= 48 secondes. A cet instant, les connexions actives sont les deux flux de la tomographie, le flux utile S-D et le flux d'arrière-plan S_1. C'est l'état du système de l'instant t= 0 seconde à l'instant t= 50,6 secondes. La faible différence entre les courbes justifie bien l'utilisation de ces algorithmes pour l'estimation des probabilités cumulatives du délai.

La source S_1 qui émet un flux de débit 300kbits/sec influe peu sur le délai observé: elle introduit une charge de ρ=0,15 au système.

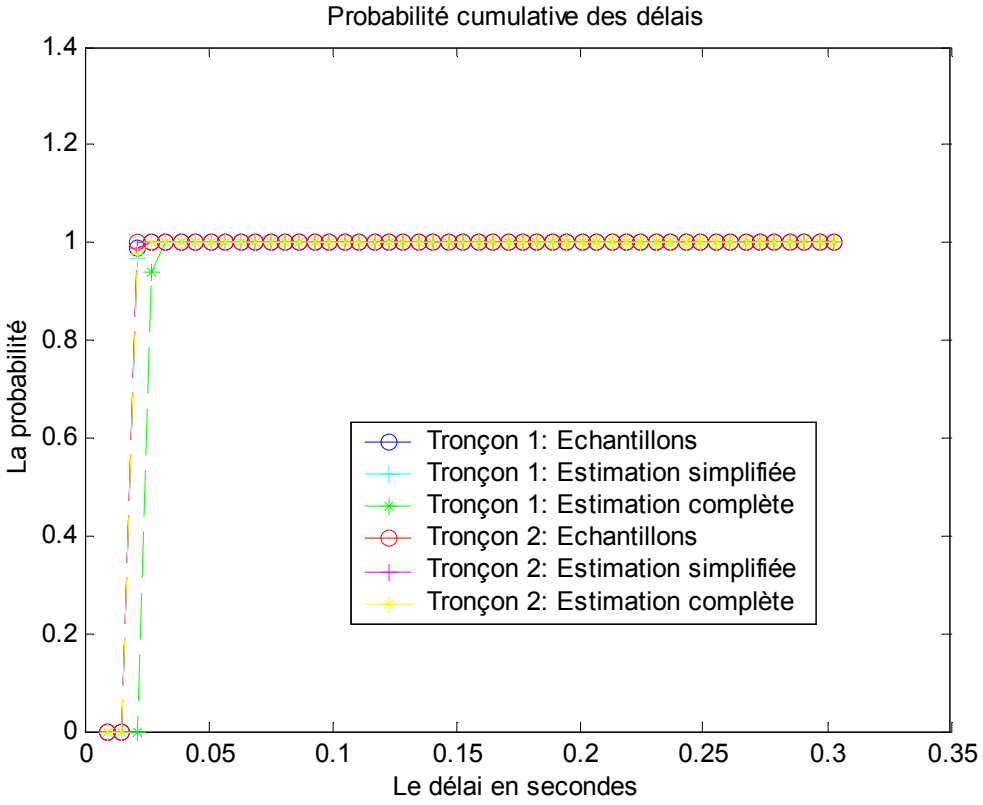


Figure 59: Les différentes fonctions de répartition du délai à l'instant t= 48 secondes.

A l'instant t= 50,6 secondes, le flux de la source S_3 est déclenché. Il ajoute un débit utile de 1300 kbits/sec. La charge du tronçon 2 devient donc:

$$\rho_2 = \frac{\sum \text{débits des connexions qui transitent sur le tronçon 2}}{\text{Le débit du lien } C_1 - C_3 - C_4}$$

$$\rho_2 = \frac{t_1 + S + S_3}{\text{Le débit du lien } C_1 - C_3 - C_4} = \frac{20 + 100 + 1300}{2\,000} = 0.71$$

Quant à la charge sur le tronçon 1, elle est toujours de $\rho_1 = \frac{320}{2000} = 0{,}16$.

Le flux S_3 markovien est majoritaire sur le tronçon 2. En supposant que ce schéma obéisse à un réseau de Jackson, et en appliquant les formules de la file M/M/1 pour chaque

lien (les flux déterministes de t_1 et t_2 sont minoritaires), nous obtenons le temps d'attente moyen des paquets S-D sur chacun des deux liens C_1-C_3 et C_3-C_4 : $E[Y] = \frac{E[X]}{(1-\rho)}$.

Les valeurs numériques utilisées donnent le délai sur C_1-C_3:

$$E[Y]_{C_1-C_3} = E[Y]_{C_3-C_4} = \frac{\frac{500*8}{2000000}}{(1-0{,}71)} = 6{,}9 \text{ msec}$$

Sur le tronçon 1, ce sont les paquets de S_1 qui sont majoritaires. Le délai sur chaque lien suit les formules d'une file M/D/1: $E[Y] = E[X] + \frac{\rho.E[X]}{2(1-\rho)}$

$$E[Y]_{C_1-C_2} = E[Y]_{C_2-C_4} = \frac{500\times 8}{2000000} + \frac{\frac{320}{2000}.\frac{500\times 8}{2000000}}{2(1-\frac{320}{2000})} = 2{,}2 \text{ msec}$$

Les deux liens S- C_1 et C_4-D peuvent être supposés des files M/D/1 (le flux utile S-D est majoritaire). Le délai s'écrit:

$$E[Y]_{S-C_1} = E[Y]_{C_4-D} = \frac{500\times 8}{10000000} + \frac{\frac{140}{10000}.\frac{500\times 8}{10000000}}{2(1-\frac{140}{10000})} = 0{,}4 \text{ msec}$$

Tableau 59: Délais moyens à partir de l'instant t= 50,6 secondes

Chemin 1		Chemin 2	
Lien	Délai calculé	Lien	Délai calculé
S - C_1	10.4	S - C_1	10.4
C_1- C_2	12.2	C_1 - C_3	16.9
C_2 - C_4	12.2	C_3 - C_4	16.9
C_4 - D	10.4	C_4 - D	10.4
Total	45.2	Total	54.6

A partir de l'instant t= 50,6 sec, le délai total prévu par la théorie tout au long du chemin 2 est de 54,6 msec. Sur le chemin 1, le délai à l'instant t= 50,6 secondes est toujours de 45,2 msec. Les valeurs estimées sont respectivement de 57,7 msec et 45,1 msec.

Comme le délai sur le chemin 1 est au moins inférieure à 25% du délai sur le chemin 2, le mécanisme déclenche la procédure de changement de route, qui est achevée à t= 60 secondes.

La Figure 60 présente les fonctions de répartition à l'instant t = 60 secondes, juste avant le changement de route. Nous observons la différence de la QoS sur les deux chemins.

La Figure 61 présente les fonctions de répartition à l'instant t = 124 secondes. La seule différence entre la Figure 60 et la Figure 61 est que le flux utile S-D transite sur le chemin 2 dans la première et sur le chemin 1 dans la seconde. Cette modification n'altère pas de façon conséquente les caractéristiques statistiques de la liaison du fait que d'une part, le débit du flux est relativement réduit, et d'autre part, la charge acheminée sur le chemin 1 est faible.

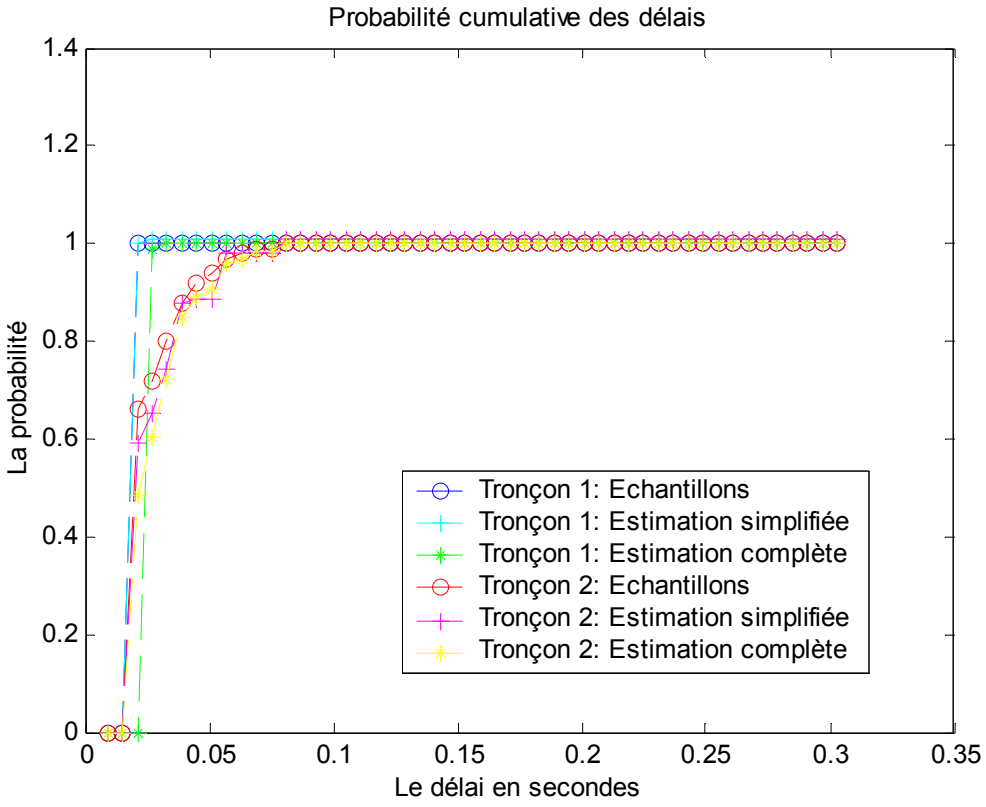


Figure 60: Les différentes fonctions de répartition du délai à l'instant t= 60 secondes.

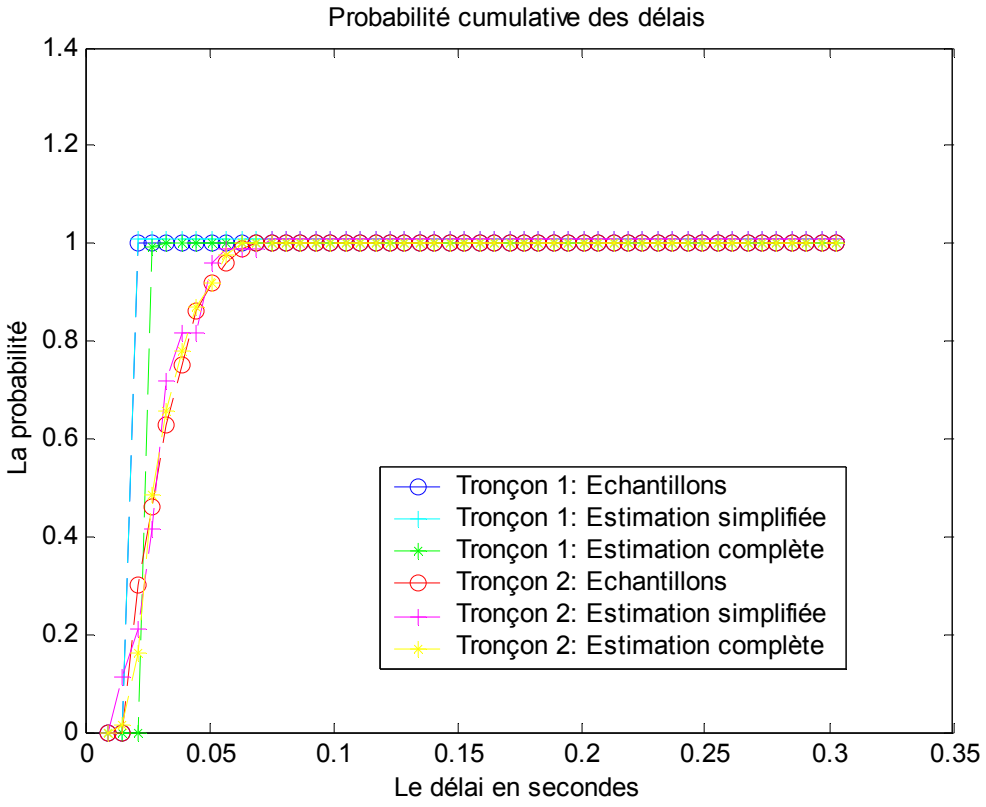


Figure 61: Les différentes fonctions de répartition du délai à l'instant t=124 secondes.

A l'instant t= 125 secondes, la source S_2 entame l'émission des paquets. Chacun, d'une taille moyenne de 4250 octets, nécessite un temps de traitement moyen sur les liens C_1-C_2 et C_2-C_4 du tronçon 1:

$$E[X] = \frac{4250 \times 8}{2000000} = 17 \text{ msec.}$$

Le débit de la source S_2 est de 300Kb/sec. Ce qui donne une charge sur le tronçon de $\rho_1 = \frac{300 + 100 + 20}{2000} = 0{,}21$. Cependant, le délai moyen observé sur le tronçon 1 est sensiblement le même que celui observé sur le tronçon 2 avec une charge de $\rho_2 = \frac{1300 + 20}{2000} = 0{,}66$. La fonction de répartition du délai (cf. Figure 62) du tronçon 1 n'est pas nulle pour les grandes valeurs du délai. Les paquets courts de la connexion S-D ainsi que t_1 et t_2 sont pris au piège derrière les gros paquets de S_2.

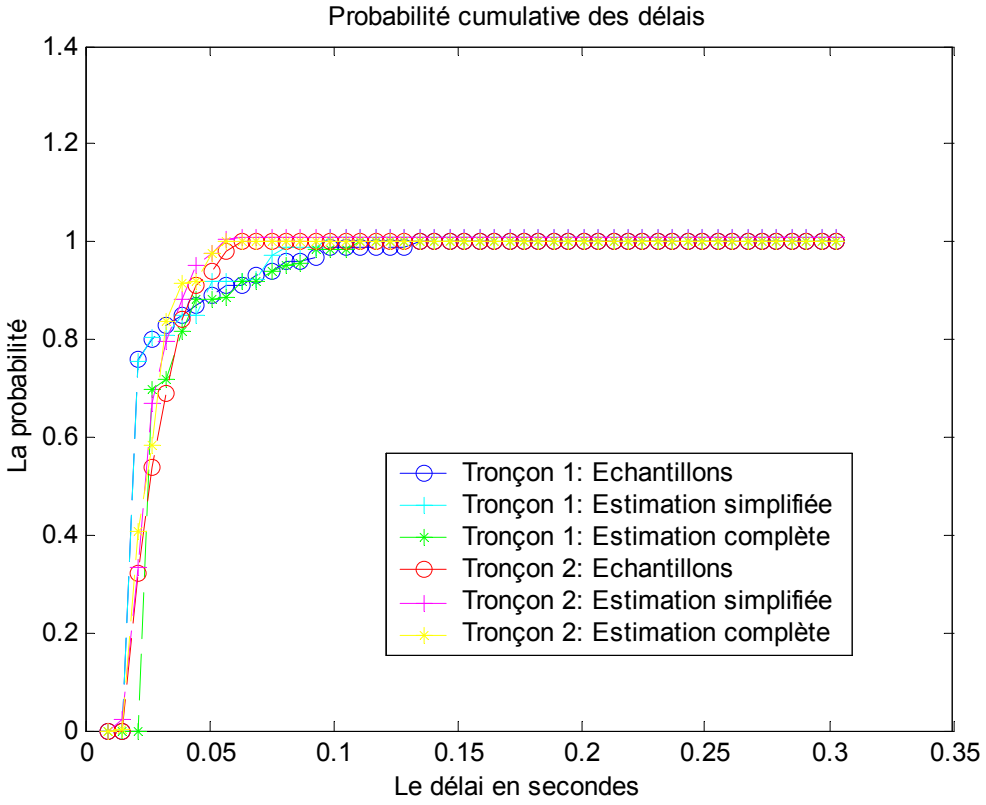


Figure 62: Les différentes fonctions de répartition du délai à l'instant t=148 secondes

A l'instant t= 150 secondes, la probabilité du dépassement du seuil τ= 0,1 secondes est supérieure au 1% tolérable (cf. Figure 63). Le mécanisme effectue alors un changement du chemin utilisé dans le but de garantir la QoS demandée. Dans une configuration où les deux chemins admettent des dépassements de seuil, la route qui admet la valeur la plus faible de cette probabilité sera retenue. Nous appliquerons alors le même principe que celui du délai moyen.

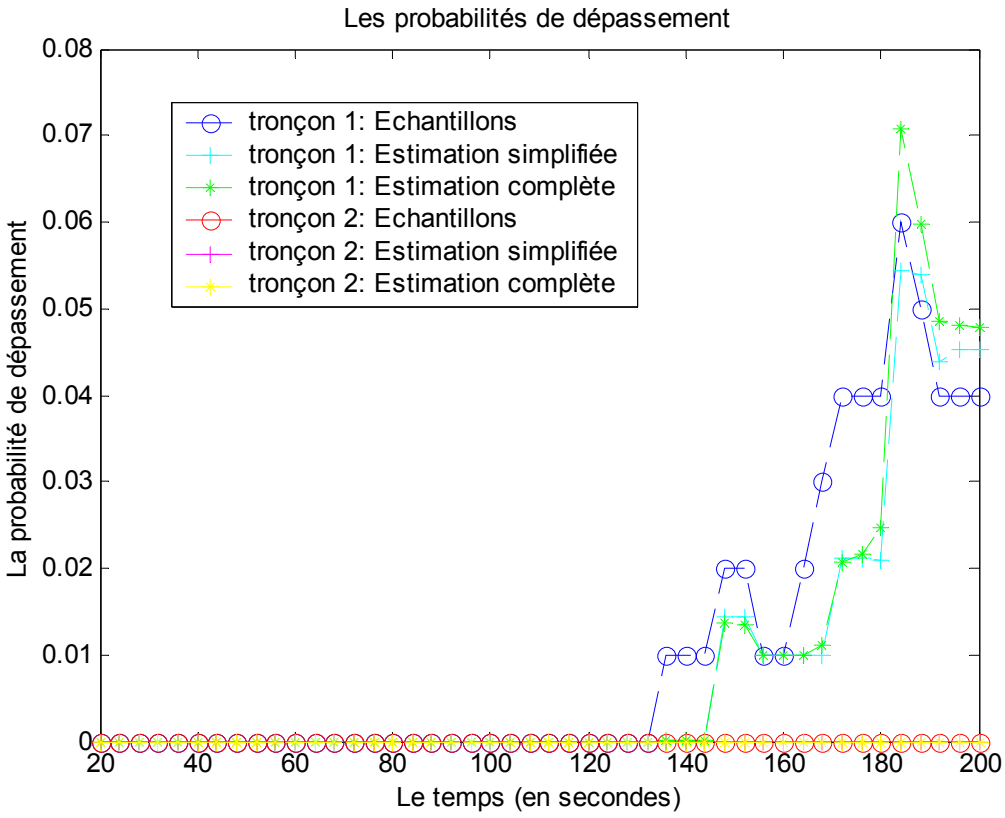


Figure 63: Les probabilités de dépassement sur chaque tronçon en fonction du temps et de l'algorithme d'estimation.

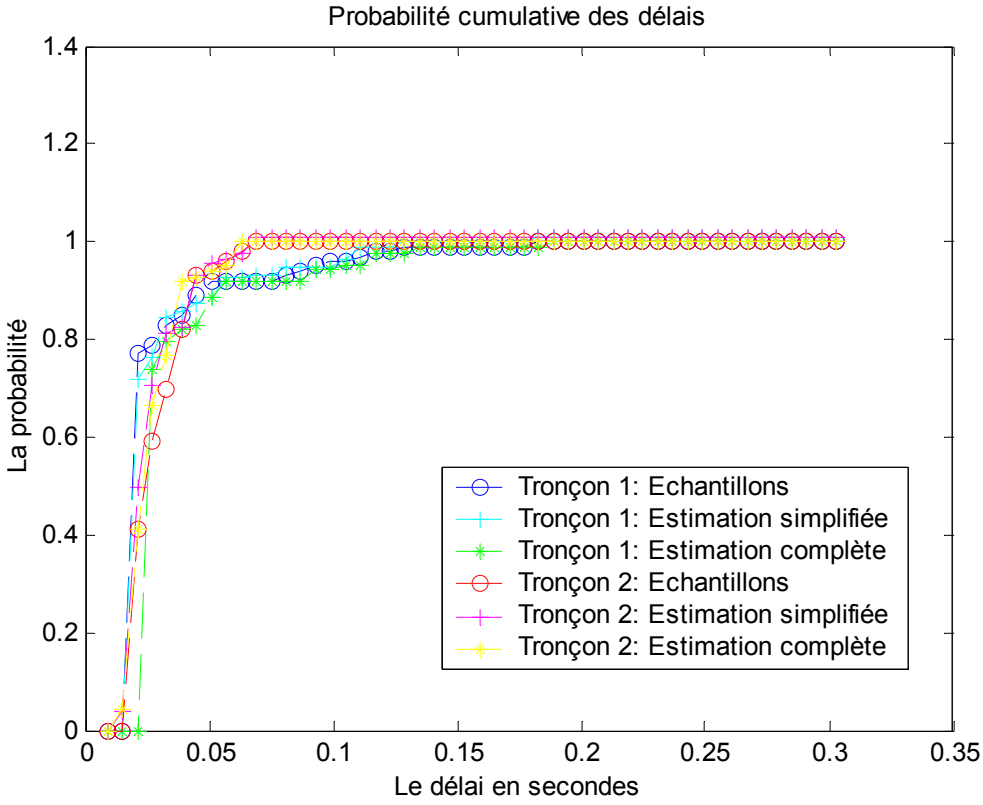


Figure 64: Les différentes fonctions de répartition du délai à l'instant t= 200 secondes.

La Figure 64 présente les différentes fonctions de répartition à l'instant t= 200 secondes. Nous rencontrons toujours une probabilité élevée d'obtention de longs délais d'attentes des paquets.

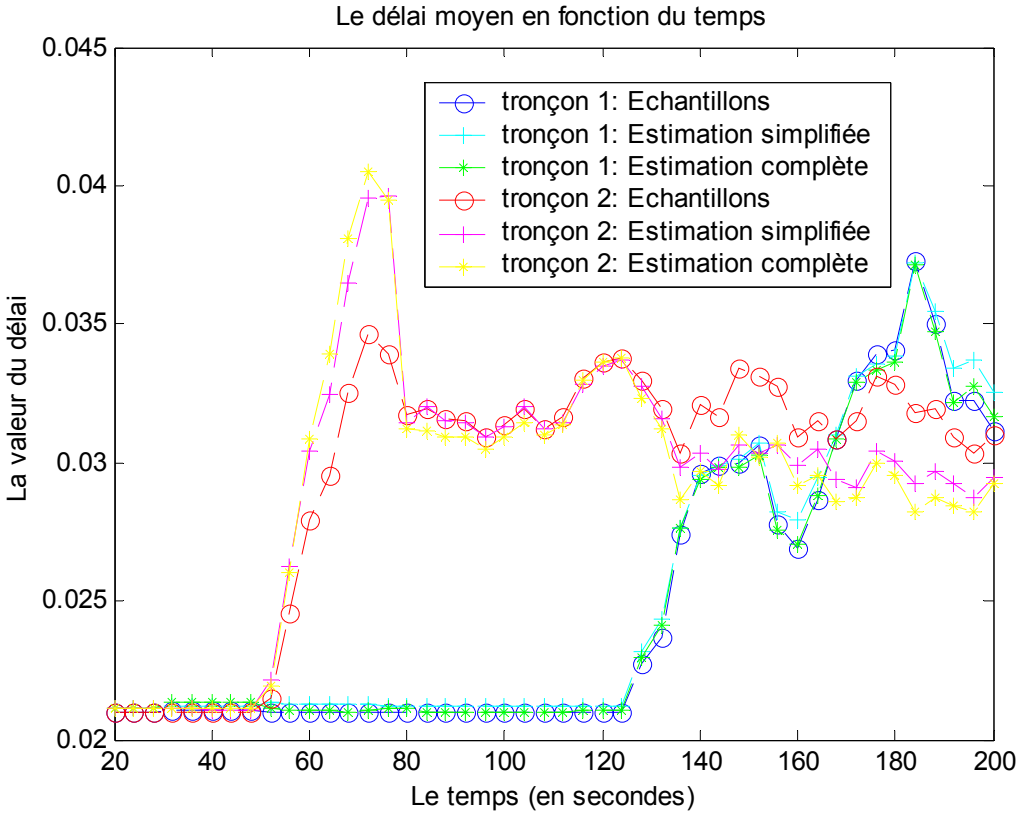


Figure 65: La variation du délai moyen sur les tronçons 1 et 2 en fonction du temps et de la méthode d'estimation.

L'évolution du délai moyen réel et estimé sur les deux tronçons 1 et 2 est illustrée sur la Figure 65. Nous pouvons voir l'accroissement du délai moyen aux instants t= 50 secondes pour le tronçon 2 et t = 125 secondes pour le tronçon 1. Cette évolution relativement douce de la pente réside dans le choix des valeurs de l'estimation: la fenêtre de largeur N = 100 échantillons inclue les échantillons qui n'avaient pas subi le délai dû aux paquets des connexions S_2 et S_3. Cette caractéristique donne à l'algorithme un aspect temporisateur face à des rafales brusques et momentanées qui peuvent perturber le réseau.

4.4.3. Le gain acquis par la source

Les figures ci-dessous montrent le gain du mécanisme mis en place. Les tracés du "***chemin 1***" (respectivement ***"chemin 2"***) sont les mesures effectuées sur le flux Source – Destination s'il suit tout au long de l'expérimentation le ***chemin 1*** (respectivement ***chemin 2***). Le "***chemin optimal***" est celui dicté par le mécanisme que nous avons proposé plus haut.

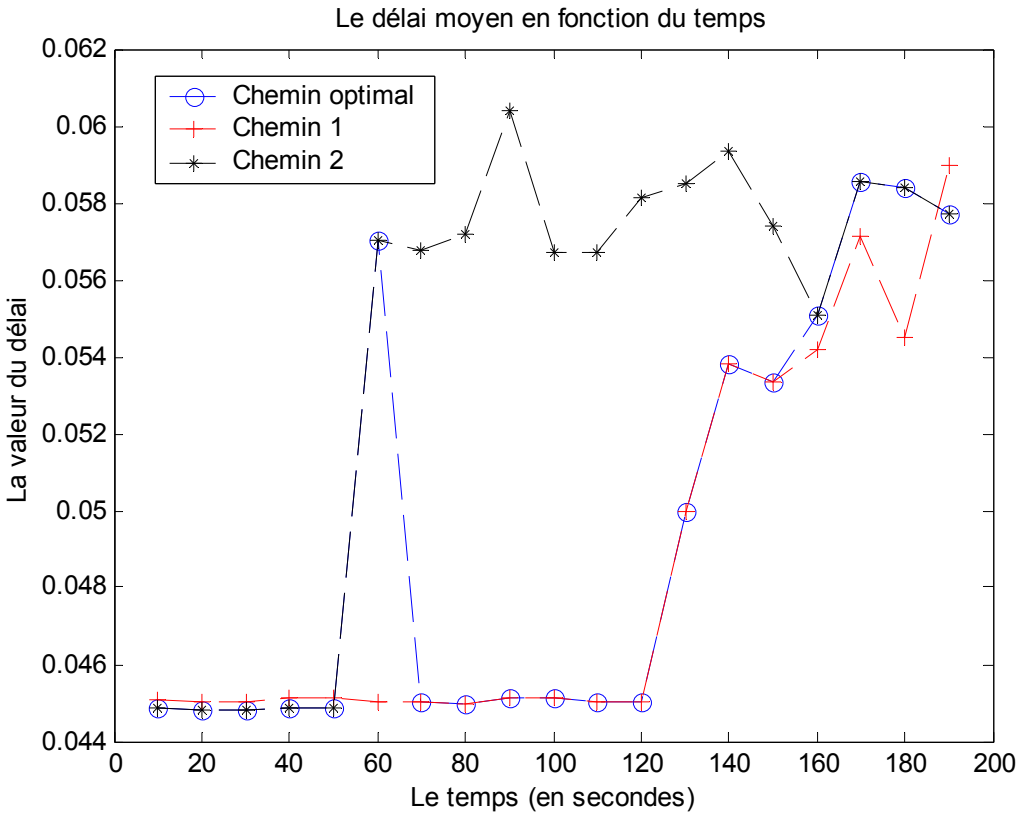


Figure 66: Comparaison du délai moyen du flux util en fonction du chemin suivi.

La Figure 66 présente l'évolution du délai moyen des paquets S-D en fonction du chemin utilisé. Nous observons le gain obtenu grâce au mécanisme utilisé en adaptant la route en fonction du délai estimé. La largeur du pic rencontré à l'instant t = 60 secondes dépend fortement de la valeur de la fenêtre d'estimation et de la fréquence à laquelle l'algorithme s'exécute. Plus cette fréquence sera grande, plus la durée du pic sera faible. Cependant, le mécanisme deviendra plus vulnérable à la variabilité des trafics présents. Une deuxième contrainte opérationnelle limite la fréquence de réalisation de l'algorithme: il s'agit du temps d'exécution de l'algorithme. Ainsi, la valeur de cet intervalle sera fixée en fonction de ces différents paramètres.

La probabilité de dépassement du seuil $\tau = 0{,}1$ secondes sur chaque chemin est présentée sur la Figure 67. Nous remarquons bien sur le chemin 2 une probabilité toujours inférieure à 2%, tandis que celle du chemin 1 s'élève à 5% avec la mise en service du flux S_2.

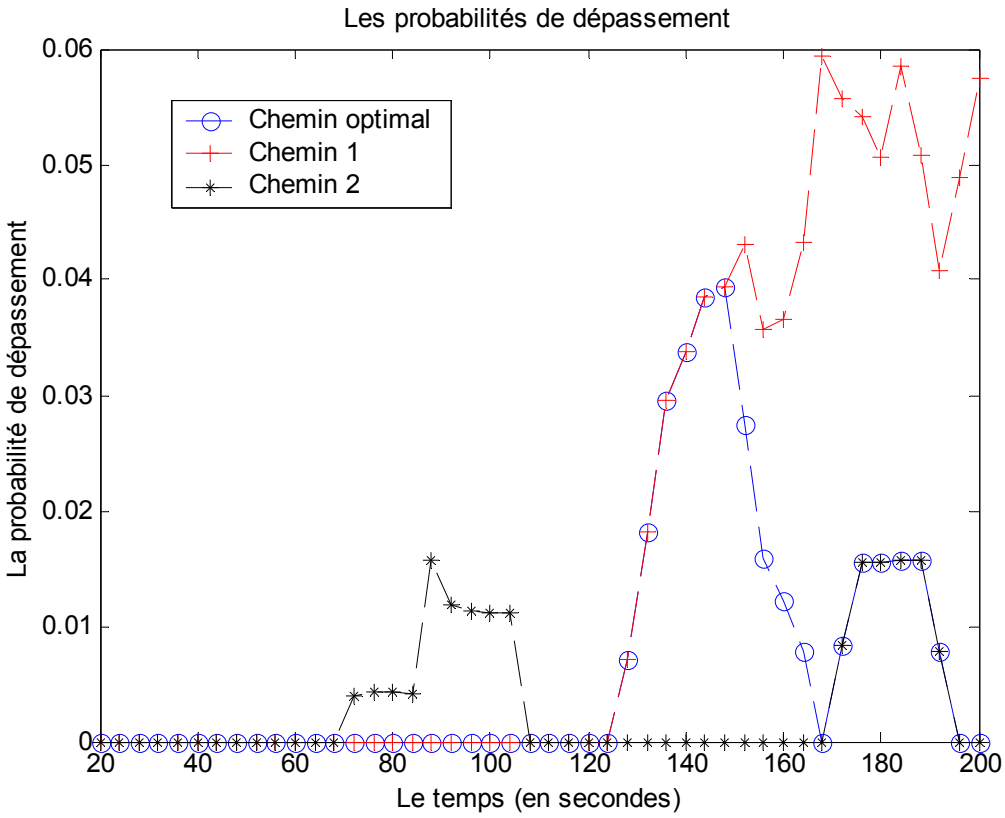


Figure 67: La probabilité du dépassement du seuil 0,1 seconde par le flux utile en fontion du chemin suivi.

4.5. Ouvertures

Plusieurs pistes peuvent être développées en tant que nouvelles orientations. Dans notre modèle de validation, nous avons modifié le chemin à l'instant de détection par le mécanisme d'un gain de délai supérieure à 25% d'une nouvelle route. Une optimisation de cet aspect pourra être de prévoir la charge introduite sur ce nouveau chemin et de calculer son effet sur le délai. Si ce dernier reste inférieur au délai du chemin actuel, l'adaptation de la route est autorisée.

Un deuxième aspect qui pourra être développé est une plus grande réactivité vis-à-vis des modifications de la charge véhiculée dans le réseau en donnant une plus grande importance aux délais des paquets récents qu'à ceux des paquets plus anciens. Une des méthodes d'application de cette discrimination est de dupliquer les valeurs récentes afin que leurs poids deviennent plus importants dans le calcul de l'estimation. Ainsi l'ensemble des délais qui sera introduit à l'algorithme sera formé d'une combinaison des délais réels. L'insertion d'un nouveau délai nécessitera alors de:

- Dupliquer q fois la nouvelle valeur, $y = \{y_1(k_T + 1), y_2(k_T + 1)\}$
- Extraire la dernière valeur $y = \{y_1(k_1 + 1), y_2(k_1 + 1)\}$
- Réduire de 1 la duplication des paquets $y = \{y_1(k_T - 1), y_2(k_T - 1)\}$, $y = \{y_1(k_T - 2), y_2(k_T - 2)\} \ldots y = \{y_1(k_T - q), y_2(k_T - q)\}$

En supposant que les paquets de mesure sont émis d'une source CBR, il faudra maintenir le nombre total d'échantillons constant tout au long de l'estimation.

Il restera alors de bien choisir la valeur de q ainsi que la méthode de duplication des *(q-1)* autres mesures. Ce choix sera fonction de la vitesse désirée de la réactivité et de la capacité d'absorption des rafales brusques.

Il est intéressant d'observer dans le cas d'un déploiement de ce mécanisme sur plusieurs extrémités dans un même réseau de voir quel serait l'effet du changement d'un flux sur le comportement et la QoS des autres flux existant dans le réseau. Ou encore de constater si une oscillation entre chemins pourra être créée entre deux ou plusieurs flux.

Le seul élément de réponse actuel que nous avons est la possibilité d'existence de la convergence vers un routage stable, tant que le nombre de chemin existant est en nombre fini (ils sont tous déjà définis lors de la création des LSP). Dans le cas où il n'existe pas une solution satisfaisant à la fois le routage et la QoS pour chaque connexion, nous ne pourrons pas garantir le bon fonctionnement du mécanisme. D'ailleurs, dans ce dernier cas, il sera impossible à un réseau de garantir une QoS si la charge totale du réseau est dépassée. La seule solution pour un tel système est de mettre en place au niveau des extrémités du réseau une fonction de contrôle d'admission. Elle assurera la limitation de la charge totale du réseau à une valeur inférieure à la quantité admissible par chaque type de QoS.

Une option des réseaux basés sur MPLS est la possibilité d'introduire un équilibrage de charge entre les liens ce qui pourra décharger certains liens tout en entraînant une meilleure QoS dans le réseau. Mais, pour le cas des connexions temps réels, cela risque de créer un problème de (dé) séquencement de paquets ainsi qu'une création de gigue, qui pourront détériorer la QoS souhaitée. Dans le cas où aucun des chemins ne garantit la QoS souhaitée, le partage sera l'ultime solution à déployer pour faire passer les flux.

4.6. Conclusion

En se basant sur divers algorithmes de tomographie des réseaux, nous avons proposé un nouveau mécanisme d'adaptation dynamique du chemin en fonction du délai observé sur les différents tronçons formant le chemin de bout en bout. Notre mécanisme sollicite une modification de l'itinéraire utilisé dans le cas d'une réduction soit au niveau du débit moyen, soit au niveau de la probabilité de dépassement d'un certain délai maximal relatif à la QoS des flux transportés.

Comme nous l'avons vu dans différents résultats obtenus, notre proposition entraîne un gain non négligeable au niveau de la QoS observée par le flux protégé. Bien que nous n'ayons utilisé ce mécanisme que sur un réseau simulé, nous sommes confiants qu'un déploiement dans un réseau réel portera des avantages.

Il est à noter que dans les différentes normes de MPLS, les méthodes de détermination des critères de QoS utilisées pour la détermination des chemins basés contraintes (Constrained Based), ne sont pas spécifiées et sont gardées pour les constructeurs. Notre méthode développée ci-dessus peut être utilisée à cette fin pour, par exemple, déterminer le délai moyen de bout en bout.

Conclusions

1. Conclusions générales

Nous présentons ici les différentes conclusions et contributions de nos travaux illustrés tout au long de ce manuscrit de thèse.

Dans un premier temps, nous avons étudié le protocole MPLS, qui permet entre autre de créer des chemins pour chaque type de classes de service. D'ailleurs, ce protocole implémente plusieurs autres capabilités d'ingénierie de trafics et de répartition de charge sur différents chemins. Ensuite, nous avons exposé les avantages de DiffServ, une des techniques de différenciation de service implémentable sur chaque routeur afin de favoriser des classes de trafics par rapport à d'autres. En associant ensemble MPLS et DiffServ il sera possible de créer des chemins différents en fonction de la classe de trafic. Cette possibilité permettra alors de dissocier les chemins de différents flux en fonction de la QoS disponible sur les liens et requise par les classes. Un des moyens de concrétiser la différenciation de service au niveau de chaque nœud est d'utiliser des ordonnanceurs de flux. En fonction de la politique d'ordonnancement, les paquets qui seront traités avant d'autres pourront gagner des fractions de secondes très importantes pour être acheminés plus tôt à destination. Nous avons ensuite comparé 2 familles d'ordonnanceurs: le WRR comme membre de la famille RR et WF^2Q comme membre de la famille GPS. Il s'est avéré que l'ordonnanceur WRR fournit de meilleurs résultats que l'ordonnanceur WF^2Q pour une distribution d'inter arrivée des paquets suivant la loi de Poisson et une distribution de la taille des paquets suivant la loi exponentielle.

Dans un second axe de recherche et en notant le manque dans la littérature scientifique d'un modèle commun à une agrégation de flux sortant d'un réseau mobile, nous nous sommes penchés sur ce sujet par plusieurs points. Nous avons premièrement étudié un réseau UTRAN, le réseau d'accès à l'UMTS, puis, nous avons prolongé notre étude en second lieu sur un réseau IEEE 802.11. Nous avons aussi étudié le comportement de l'agrégat en fonction du type des flux émis. Ainsi, l'agrégation de flux voix, modélisée chacune par un processus On/Off, est bien approximée par un processus MMPP 2 pour le réseau 802.11. Par contre ces flux sont mieux approximés par une loi d'inter arrivée Gaussienne dans un réseau UTRAN. D'ailleurs, la loi Gaussienne modélise bien les flux Web, FTP et le multiplexage des 3 flux à la sortie de la couche MAC UTRAN. Dans le réseau WiFi, la loi Gaussienne modélise les flux FTP quand le nombre de source voix est faible. La modélisation de l'agrégat devient mieux approximé par un processus MMPP 2 quand le nombre de sources voix devient élevé. L'annexe 1 comporte les diverses études statistiques que nous avons utilisées avant d'aboutir aux algorithmes en question.

Nous avons entamé notre troisième axe de recherche par l'étude d'un modèle analytique pour l'ordonnanceur GPS à trois classes. Bien que les équations soient linéaires et pas très nombreuses (pour trois classes, il existe 27 équations de Kolmogorov), la solution devient intractable du fait du nombre élevé de paramètres. Les solutions intermédiaires sont tellement volumineuses en espace mémoire que les ordinateurs que nous avons utilisés n'ont pas pu résoudre. Nous avons alors entamé une analyse par simulation sur un nœud implémentant WRR. Avec plusieurs poids associés à chaque classes ainsi que deux

distributions de la taille des paquets, en l'occurrence une distribution exponentielle et une distribution suivant la loi de Pareto, nous avons tracé la variation du délai en fonction de la charge introduite par chacune des trois classes de services. Nous avons conclu que la meilleure implémentation consistera à attribuer une portion aux flux voix supérieure à la valeur de la bande passante moyenne de ces flux. Cela permettra de leur accorder une QoS supérieure. Plus la valeur du poids est grande, plus les flux seront prioritaires. Bien entendu, le gain d'une classe se fait au détriment d'une autre. Alors, pour valider notre schéma, nous avons mesuré les performances d'un réseau avec les différents mécanismes de QoS présentés dans le chapitre 1. La priorité absolue garantit la meilleure QoS aux flux voix. Mais ce gain défavorise le trafic de données qui nécessite un taux de perte réduit. En utilisant l'ordonnanceur WRR avec la pondération adéquate, nous avons pu maintenir un délai réduit aux flux de classe 1 tout en réduisant le taux de perte des données de la classe 2. Nous avons obtenu les meilleures performances en utilisant l'équilibrage de charge.

Comme nous avons constaté que la QoS d'un flux peut être détériorée à cause d'une charge élevée sur au moins un des nœuds qui forment le chemin de bout en bout, nous avons conçu dans notre dernier axe de recherche une méthode d'adaptation dynamique du chemin en fonction de la QoS disponible sur les liens et celle désirée par le flux. Cette méthode repose sur la notion de "tomographie des réseaux" et elle est facilement déployée grâce au protocole MPLS. Notre contribution réside dans le domaine de la détermination de la variation du délai en fonction de l'espace (sur les différents liens et nœuds) et du temps. La modification de la route utilisée est déclanchée par deux facteurs. Soit que le délai moyen sur le chemin utilisé actuellement dépasse par une valeur ε critique et pendant un laps de temps donné le délai moyen sur une autre route. Soit que la probabilité de dépassement d'une valeur reste supérieure à un seuil pour une durée τ. Les gains que nous obtenons justifient bien l'utilité de cette implémentation par rapport à un réseau standard.

2. Ouvertures et perspectives

Notre travail ouvre la voie à plusieurs perspectives intéressantes. Nous les présenterons ci-dessous comme suite aux travaux relatifs aux différents chapitres de cette thèse:

(1) Dans l'architecture des réseaux Internet du futur, une nouvelle composante pourra s'introduire: la gestion de la QoS par un protocole basé politique e.g. COPS. Ce type de protocole pourra non seulement automatiser et mieux contrôler la gestion de la QoS, mais il permettra aussi une administration de la QoS plus personnalisée et mieux ciblée. Ainsi, le réseau interagira différemment avec chaque type de clients/connexion en fonction des divers droits octroyés au moment de la création du profil. C'est alors cette manière qui fixera la priorité des paquets, ou proposera tel chemin ou tel autre chemin…

(2) Une fois les lois d'agrégations déterminées, une étude sur la variation des paramètres σ et λ pour le processus MMPP et la moyenne et la variance pour la loi Gaussienne peuvent alors démarrer. Elle analysera aussi l'effet d'un ajout ou d'un retrait d'un certain nombre d'utilisateurs sur les valeurs de ces paramètres.

Evidement, notre travail peut lancer d'autres études sur les différentes couches MAC du protocole 802.11x ou avec d'autres techniques CDMA que celle que nous avons utilisé dans le réseau UMTS.

(3) Dans la partie de différenciation de service, nous n'avons pas pu résoudre analytiquement le modèle que nous avons proposé. La solution que nous avons présentée est basée sur une simulation d'un réseau implémentant les caractéristiques voulues. Il se peut qu'une analyse numérique puisse donner des résultats à cette question qui reste irrésolue malgré les efforts et les différents travaux déjà existants dans ce domaine.

Nous avons, par ailleurs, étudié l'effet du partage de charge avec MPLS. Cependant, ce protocole permet beaucoup plus de possibilités de gestion de QoS comme nous l'avons cité dans le Chapitre 1. Pour ne mentionner qu'un des volets qui méritent d'être analysés nous évoquons l'aspect de la réservation de la bande passante sur les liens/LSP crées. Cet aspect englobe aussi bien l'effet de la réservation des ressources sur la QoS observée, que les méthodes et les algorithmes d'estimation de cette bande passante qu'il faudra épargner aux flux.

(4) Comme nous l'avons noté dans le Chapitre 4, nos travaux sur l'adaptation dynamique du routage peuvent être complétés par une étude du facteur d'échelle de l'implémentation de ce mécanisme dans un réseau réel. Il faudra s'assurer que d'une part, un changement de chemin d'un flux n'entraînera pas une cascade d'autres modifications, et d'autre part, que le nombre de paquets de tomographie introduit dans le système reste faible et ne perturbe pas les caractéristiques statistiques du réseau observé. Une autre piste plausible est d'affiner les paramètres de réactivité choisie afin de répondre aux exigences temporelles des réseaux. Les différentes valeurs des seuils et du nombre d'échantillons pris en compte seront alors étudiées et leur impact sera mesuré sur la QoS reçue par les flux.

Enfin, le contrôle d'admission, bien qu'il ne soit pas étudié dans cette thèse, reste toujours un élément primordial de contrôle de la QoS: en définissant et utilisant une fonction CAC (Call Admission Control) adéquate, nous pourrons garantir la qualité de service demandée de tous les flux admis au réseau. Mais cette fonction nécessite une compréhension et un contrôle total des différents comportements qui peuvent exister dans chaque composant appartenant au réseau. Malheureusement, les réseaux IP sont loin de ce scénario là. Des travaux dans ce sens sont donc nécessaires pour pouvoir contrôler efficacement les utilisateurs et les flux traversant les liens.

ANNEXE 1 :
ESTIMATION MMPP

A partir d'un fichier trace généré par simulation, les paramètres du trafic, supposés MMPP, seront estimés. L'estimation consiste à trouver les valeurs des paramètres qui approximent le mieux ce processus.

Deux familles de méthodes d'estimations de paramètres des processus MMPP sont possibles: les méthodes basées sur la comparaison de moments et les méthodes basées sur la maximisation de vraisemblance (MLE Maximum Likelihood Estimation). Dans **[56]**, l'auteur effectue une comparaison entres ces méthodes pour le cas MMPP 2. Les meilleurs sont celles basées sur le MLE.

Pour simplifier les notations, nous utiliserons la notation d'un modèle MMPP 2 qui admet comme paramètres:

$$Q = \begin{pmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{pmatrix} \quad \text{et} \quad \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

Dans les parties qui suivent, nous présentons les méthodes nécessaires afin de pouvoir modéliser les trafics. Dans les deux premières parties, nous introduisons les MLE et les Modèles de Markov Caché (HMM Hidden Markov Model). Ce dernier modèle nous est indispensable pour trouver les paramètres recherchés. La troisième partie décrit les modifications apportées à MLE quand il s'agit de traiter des HMM. Ensuite, dans la quatrième partie, nous introduisons l'algorithme de maximisation de prévision (EM Expectation Maximisation). C'est cet algorithme qui est le mieux adapté pour notre situation **[55]**. Dans la cinquième partie, et à partir des différents éléments déjà cités, nous décrivons l'algorithme utilisé pour déterminer les paramètres recherchés. Pour diminuer le temps de convergence de l'algorithme, nous avons utilisé une méthode d'estimation des paramètres décrite dans **[51]**. Dans nos différentes validations, cette dernière méthode donnait des valeurs proches, de l'ordre de 20%-50% des valeurs réelles. Cet écart n'est pas acceptable pour adopter le résultat de cet algorithme comme les valeurs finales, mais, nous l'utiliserons afin d'avoir des valeurs initiales nécessaires à notre algorithme itératif.

1. MLE: Maximum Likelihood Estimation

Les méthodes MLE utilisent les échantillons pour estimer les valeurs des paramètres pour lesquels les échantillons obtenus sont les plus probables.

A partir d'un échantillon d'interarrivée $\{\tau_i\}$ d'un processus de fonction de répartition *f(t)*, la probabilité d'avoir une observation τ est:

$$P(\tau) = f(\tau)$$

En prenant N le nombre d'échantillons et θ l'ensemble des paramètres, la fonction $l(\theta)$ de maximum de vraisemblance s'écrit:

$$l(\theta) = \prod_{i=1}^{N} P(\tau_i) = \prod_{i=1}^{N} f(\tau_i)$$

L'ensemble θ des paramètres les plus probables est celui qui maximise la fonction de vraisemblance $l(\theta)$. Pour trouver les paramètres, il faudra donc trouver les valeurs pour lesquels $l(\theta)$ est maximale. La meilleure méthode est de dériver par rapport aux paramètres en question. Pour simplifier et comme le logarithme est une fonction monotone strictement croissante, maximiser $L = ln\, l(\theta)$ revient à maximiser $l(\theta)$.

Pour illustrer par un exemple, nous appliquons ci-dessous cette méthode à un échantillon de variables aléatoires indépendantes et identiquement distribuées suivant une loi exponentielle.

La densité de temps d'interarrivées s'écrit pour la loi exponentielle:

$$f(\tau) = \lambda e^{-\lambda\tau}$$

Le maximum de vraisemblance s'écrit alors:

$$l(\theta) = \prod_{i=1}^{N} f(\tau_i) = \prod_{i=1}^{N} \lambda e^{-\lambda\tau_i}$$

En prenant $L(\theta)=ln\, l(\theta)$:

$$L(\theta) = \ln(\prod_{i=1}^{N} \lambda e^{-\lambda\tau_i}) = \sum_{i=1}^{N} \ln(\lambda e^{-\lambda\tau_i}) = \sum_{i=1}^{N} \ln(\lambda) + \sum_{i=1}^{N} \ln(e^{-\lambda\tau_i}) = N\ln(\lambda) - \sum_{i=1}^{N} \lambda\tau_i$$

$$L(\theta) = N\ln(\lambda) - \lambda\sum_{i=1}^{N} \tau_i$$

Pour trouver la valeur qui maximise L, il suffit alors de dériver par rapport au paramètre recherché et d'égaliser le résultat à 0:

$$\frac{\partial L(\theta)}{\partial\lambda} = \frac{\partial\left(N\ln(\lambda) - \lambda\sum_{i=1}^{N}\tau_i\right)}{\partial\lambda} = \frac{\partial N\ln(\lambda)}{\partial\lambda} - \frac{\partial\lambda\sum_{i=1}^{N}\tau_i}{\partial\lambda} = N\frac{\partial\ln(\lambda)}{\partial\lambda} - \frac{\partial\lambda}{\partial\lambda}\sum_{i=1}^{N}\tau_i$$

$$\frac{\partial L(\theta)}{\partial\lambda} = N\frac{1}{\lambda} - \sum_{i=1}^{N}\tau_i$$

Donc

$$\frac{\partial L(\theta)}{\partial\lambda} = 0 \Rightarrow N\frac{1}{\lambda} - \sum_{i=1}^{N}\tau_i = 0 \Rightarrow N\frac{1}{\lambda} = \sum_{i=1}^{N}\tau_i$$

$$\Rightarrow \frac{1}{\lambda} = \frac{\sum_{i=1}^{N}\tau_i}{N} = \sum_{i=1}^{N}\frac{\tau_i}{N}$$

Ce qui correspond à la moyenne des temps d'interarrivées des échantillons.

Dans le cas d'un processus MMPP-2, cette méthode, telle qu'elle est, ne s'applique pas: les équations des dérivées de la fonction d'inter arrivée ne sont pas solubles analytiquement. Pour résoudre ce problème, il faudra trouver un algorithme pour maximiser numériquement la fonction de vraisemblance.

2. Chaînes de Markov Cachées

a. Définition

Une chaîne de Markov cachée est un processus bivariant à temps discret $\{(X_n, Y_n)\}$ tel que :

i. La chaîne $\{X_n\}$ des états du processus est une chaîne de Markov à état fini.
$$P\{X_k / X_{k-1},...X_0\} = P\{X_k / X_{k-1}\} \qquad \forall k$$

ii. Sachant $\{X_n\}$, $\{Y_n\}$ est une séquence de variables aléatoires conditionnellement indépendantes
$$P\{(Y_k / \{X_k\}) \cap (Y_{k-i} / \{X_k\})\} = 0 \qquad \forall k, i \neq 0$$

iii. La distribution conditionnelle de Y_k dépend de $\{X_n\}$ seulement par X_k.
$$P\{Y_k / X_k,...X_0\} = P\{Y_k / X_k\}$$

$\{X_k\}$ est appelé le *régime*.

En d'autres termes: le modèle HMM est un processus doublement stochastique dont l'un de ces processus n'est pas observable et sera noté "caché". Ce dernier peut être déterminé par un autre processus stochastique qui produit la séquence des observations. Les deux exemples ci-dessous illustrent mieux cette définition.

b. Premier exemple de HMM: Pile ou Face

Imaginons que nous sommes dans une salle divisée en deux par une barrière: nous sommes dans une partie, une autre personne est dans l'autre. Cette personne effectue un jeu de pile ou face avec une ou plusieurs pièces. Elle ne nous dit pas avec quelle pièce elle a obtenu le résultat, mais elle nous communique sa valeur (Pile ou Face). L'observation est donc :

$$O=O_1\ O_2\ O_3\ \ldots\ O_T$$
$$=P\ P\ F\ \ldots\ F$$

Il faudra déterminer:

- Quels états dans le modèle correspondent à quelles observations.
- Le nombre d'états dans le modèle.

a) Si nous décidons qu'une seule pièce est utilisée alors le système sera une chaîne de Markov simple avec p et (1-p) comme probabilité des états respectifs pile et face.

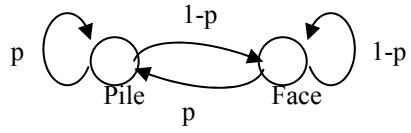


Figure 68: Premier scénario

b) Si nous décidons que deux pièces ont été utilisées et que chaque pièce admet sa propre probabilité de pile ou face, alors, en plus de la chaîne de Markov de chaque jeton, une matrice de passage entre ces deux pièces existe aussi.

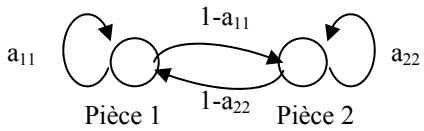


Figure 69: Second scénario

c. Deuxième exemple de HMM: le modèle des balles dans l'urne

Considérons un modèle de N urnes contenant des balles de M couleurs. Un procédé de tirage des différentes urnes est effectué et uniquement la couleur nous est communiquée.

Le processus résultant est une série de couleurs :

O = {Bleu, Blanc, Rouge, Vert, …, Marron}

Le processus HMM est un processus qui associe à chaque urne un état ainsi qu'une matrice de transition entre les urnes. A chaque urne, une probabilité de tirage de chaque couleur est définie.

d. Les éléments d'une HMM

Cinq éléments sont définis pour les HMM:

1. N_i : le nombre d'états du modèle, i.e. le nombre de pièce ou d'urnes.
$$S = \{S_1, S_2, \ldots, S_N\}.$$
2. M : le nombre de symboles e.g. le résultat du processus, i.e. pile/face, la couleur de la balle $V = \{V_1, V_2, \ldots, V_M\}$
3. La matrice des probabilités de transition entre les états.
$$A_{ij} = P[q_{t+1} = S_j / q_t = S_i] \quad 1 \le i \le N, \ 1 \le j \le N$$
4. La distribution de la probabilité d'observation d'un symbole dans chaque état j:
$$B = \{b_j(k)\} \text{ où } b_j(k) = P[V_k \text{ à } t / q_t = S_j] \quad 1 \le j \le N \text{ et } 1 \le k \le M$$
5. La distribution initiale des états $\Pi = \{\Pi_j\}$
$$\Pi_j = P[q_1 = S_i] \quad 1 \le i \le N$$

Les HMM sont donc une bonne méthode pour trouver les paramètres d'un MMPP.

3. MLE pour les HMM

Pour le cas particulier des HMM, un facteur L_c, appelé *complet Likelihood* (vraisemblance complète) est défini pour mesurer la vraisemblance du résultat à celui de l'échantillon.

La vraisemblance $l(\theta)$ sachant les différentes observations:

$$l(\theta; y_1, \ldots, y_n) = \pi(\theta) \left\{ \prod_{i=1}^{n} f(y_i; \theta) \right\} \mathbf{1}$$

où $\pi(\theta)$ est la probabilité stationnaire des états de la chaîne,

1 est un vecteur colonne où tous les éléments sont des 1.

La raison de définir la vraisemblance complète est de simplifier les calculs. Elle est définie comme suit:

$$l_c(\theta; y_1, \ldots, y_n; x_0, \ldots, x_n) = \prod_{i=1}^{n} f_{x_{i-1} x_i}(y_i; \theta)$$

Elle est complète vu que les échantillons sont connus et les états de la chaîne cachée sont eux aussi connus. Cette nouvelle vraisemblance est plus facile à calculer: c'est une maximisation d'un produit de scalaire, dépendant des paramètres à trouver. Dans le cas précédent i.e. la vraisemblance, le produit est un produit de matrice, de dimension égale au nombre d'état de la chaîne.

4. L'algorithme EM (Expectation Maximization)

Cet algorithme [16] est utilisé pour la résolution des chaînes de Markov cachées. Il se décompose en deux étapes :

- Etape E: Expectation: Estimer / prévoir la valeur des paramètres rechercher puis calculer les valeurs cachées en se basant sur la prévision. C'est l'état de la chaîne.
- Etape M: Maximisation: En fonction de cette chaîne de transition des états et des échantillons obtenus, nous ré estimons les valeurs des paramètres en maximisant le MLE. Tant que la précision de la convergence n'a pas été atteinte, revenir à l'étape E.

Pour illustrer cet algorithme, nous développons l'exemple ci-dessous :

Soit *n* observations catégorisées en 4 groupes $y=(y_1,y_2,y_3,y_4)$= (125, 18, 20, 34).
La probabilité qu'un échantillon soit dans un groupe est donnée par :

$$\left(\frac{1}{2}+\frac{1}{4}\pi,\frac{1}{4}(1-\pi),\frac{1}{2}(1-\pi),\frac{1}{4}\pi\right) \quad 0\le\pi\le 1$$

Quelle est la valeur de π qui maximise la vraisemblance ?

La fonction de vraisemblance est :

$$l(\pi)\approx\left(\frac{1}{2}+\frac{1}{4}\pi\right)^{y_1}\left(\frac{1}{4}(1-\pi)\right)^{y_2}\left(\frac{1}{4}(1-\pi)\right)^{y_3}\left(\frac{1}{4}\pi\right)^{y_4}$$

Pour maximiser $l(\pi)$, il suffit de maximiser $L(\pi)=\log l(\pi)$:

$$L(\pi)=\log l(\pi)\approx y_1\log\left(\frac{1}{2}+\frac{1}{4}\pi\right)+y_2\log\left(\frac{1}{4}(1-\pi)\right)+y_3\log\left(\frac{1}{4}(1-\pi)\right)+y_4\log\left(\frac{1}{4}\pi\right)$$

En dérivant par rapport à π :

$$\frac{dL(\pi)}{d\pi}=\frac{y_1}{2+\pi}-\frac{y_2+y_3}{1-\pi}+\frac{y_4}{\pi}=0\Rightarrow\pi=0.627$$

En réalité, nous avons 5 catégories (une catégorie cachée). Les données complètes sont donc :

$$x=(x_1,x_2,x_3,x_4,x_5)$$

En prenant comme probabilités correspondantes :

$$\left(\frac{1}{2},\frac{1}{4}\pi,\frac{1}{4}(1-\pi),\frac{1}{2}(1-\pi),\frac{1}{4}\pi\right) \qquad 0\le\pi\le 1$$

En d'autres termes, nous avons divisé y_1 en deux variables x_1+x_2.

$$l(\pi)\approx\left(\frac{1}{2}\right)^{x_1}\left(\frac{1}{4}\pi\right)^{x_2}\left(\frac{1}{4}(1-\pi)\right)^{x_3}\left(\frac{1}{4}(1-\pi)\right)^{x_4}\left(\frac{1}{4}\pi\right)^{x_5}$$

avec $x_1+x_2=y_1=125$; $x_3=y_2=18$; $x_4=y_3=20$; $x_5=y_4=34$.

$$L(\pi)=\log l(\pi)\approx x_1\log\left(\frac{1}{2}\right)+x_2\log\left(\frac{1}{4}\pi\right)+x_3\log\left(\frac{1}{4}(1-\pi)\right)+x_4\log\left(\frac{1}{4}(1-\pi)\right)+x_5\log\left(\frac{1}{4}\pi\right)$$

$$\frac{dL(\pi)}{d\pi}=\frac{x_2+x_5}{\pi}-\frac{x_3+x_4}{1-\pi}=0$$

$$\frac{dL(\pi)}{d\pi}=\frac{x_2+34}{\pi}-\frac{38}{1-\pi}=0$$

- Etape E:
  Estimation de x en ayant les données observées y:

$$x_1^{(i)} = y_1 \frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}\pi^{(i)}} \text{ et } x_2^{(i)} = y_1 \frac{\frac{1}{4}\pi^{(i)}}{\frac{1}{2}+\frac{1}{4}\pi^{(i)}}$$

En supposant π=0,5 (prévision), nous trouvons:

$$x_1^{(1)} = y_1 \frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}\pi^{(0)}} = 125\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}0{,}5} = 100 \text{ et } x_2^{(1)} = y_1 \frac{\frac{1}{4}\pi^{(0)}}{\frac{1}{2}+\frac{1}{4}\pi^{(0)}} = 125\frac{\frac{1}{4}0{,}5}{\frac{1}{2}+\frac{1}{4}0{,}5} = 25$$

- Etape M:
  Prendre les données complètes estimées et calculer le MLE comme d'habitude
  $\left(x_1^{(i)}, x_2^{(i)}, y_2, y_3, y_4\right)$

$$\frac{dL(\pi)}{d\pi} = \frac{x_2+34}{\pi} - \frac{38}{1-\pi} = 0$$
$$\frac{dL(\pi)}{d\pi} = \frac{25+34}{\pi} - \frac{38}{1-\pi} = 0$$

  Ce qui donne $\pi^{(1)} = 0{,}608$

  Nous recommençons:

$$x_1^{(2)} = y_1 \frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}\pi^{(1)}} = 125\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{4}0{,}608} = 95{,}86\text{ , } x_2^{(2)} = y_1 \frac{\frac{1}{4}\pi^{(1)}}{\frac{1}{2}+\frac{1}{4}\pi^{(1)}} = 125\frac{\frac{1}{4}0{,}608}{\frac{1}{2}+\frac{1}{4}0{,}608} = 29{,}14$$

  Ce qui donne

$$\frac{dL(\pi)}{d\pi} = \frac{x_2+34}{\pi} - \frac{38}{1-\pi} = 0$$
$$\frac{dL(\pi)}{d\pi} = \frac{29{,}14+34}{\pi} - \frac{38}{1-\pi} = 0$$
$$\pi^{(2)} = 0{,}624$$

Cette procédure est répétée jusqu'à ce que la convergence soit atteinte, c'est à dire, $\left|\frac{\pi^{(n)} - \pi^{(n-1)}}{\pi^{(n)}}\right| < tol$, où *tol* est la tolérance (en générale égale à 0,001=0,1%).

Dans l'exemple ci-dessous, la convergence est atteinte au bout de la quatrième itération : π=0,627. Les valeurs des x_1 et x_2 sont respectivement x_1= 95,2 et x_2 =29,8.

La convergence de cet algorithme vers les vraies valeurs des paramètres est démontrée dans **[16]**.

5. Solution adoptée: utiliser l'algorithme EM pour MLE

a. L'algorithme proprement dit

En s'inspirant de **[55]**, la méthode utilisée est l'algorithme EM pour le MLE. L'algorithme se résume ainsi : (pour plus de détail, se référer à **[55]**).

π_0 étant l'estimation initiale.

n le nombre d'échantillons.

Δt_k l'intervalle de temps entre les paquets formant l'échantillon k.

r=2, le nombre d'état du MMPP.
La matrice de densité de transition est donnée par:

$$f(x) = \exp\{(Q-\Lambda)x\}\Lambda$$

La matrice de probabilité de transition de la chaîne $\{X_k\}$ est :

$$P = \int_0^{+\infty} f(y)dy = (\Lambda - Q)^{-1}\Lambda$$

Comme dans **[36]**, la fonction $\overline{F}(t)$ définit une matrice r x r. L'élément *(i,j)* de cette fonction est la probabilité que le processus MMPP démarrant de l'état *i* soit à l'état *j* à l'instant t et qu'aucun évènement ne se produise entre les instants 0 et t.

$$\overline{F}(t) = \exp\{(Q-\Lambda)t\}$$

Chaque itération de l'algorithme est comme suit :

(1) Soit $L(0) = \pi^0$, et pour k=1,..,n, soit $L(k)=L(k-1)f(\Delta t_k)$
(2) Soit $R(n+1) = \mathbf{1}$, et pour k= n,..,1, soit $R(k)=f(\Delta t_k)R(k+1)$
(3) Pour $i,j = 1..r$, soit $A_{ij} = 0$, $B_i=0$.
(4) Pour k=1,..,n, soit

$$A_{ij} \to A_{ij} + L(k-1)\int_{t_{k-1}}^{t_k} \overline{F}(t-t_{k-1})1_i 1_j^T f(t_k - t)dtR(k+1)$$

(5) Pour k=1,..,n, soit

$$B_i \to B_i + L_i(k)R_i(k+1)$$

(6) Ce qui donne comme estimation :

$$\hat{q}_{ij} = q_{ij}^0 \frac{A_{ij}}{A_{ii}}, i, j = 1..r, i \neq j \text{ et } \hat{\lambda}_i = \frac{B_i}{A_{ii}}, i = 1..r.$$

Le temps de calcul de l'intégrale, le point (4) de l'algorithme, est extrêmement long. Pour résoudre ce problème, la méthode de décomposition de f(t) et $\overline{F}(t)$ décrite dans **[55]** est utilisée. Elle se résume ainsi:

Q-Λ peut être diagonalisé, i.e. il existe au moins une matrice S tel que $Q-\Lambda=SDS^{-1}$, avec $D=diag(d_1, d_2,..., d_r)$, la matrice des valeurs propres de *Q-Λ*, alors, *f(t)* peut s'écrire $f(t)=Sexp(Dt)S^{-1}\Lambda^0$ et $\overline{F}(t) = Sexp(Dt)S^{-1}$. De plus, après plusieurs manipulations, l'élément *pq* de l'intégrale peut s'écrire:

$$\sum_{u=1}^{r}\sum_{v=1}^{r} S_{pu}S_{ui}^{-1}S_{jv}S_{vq}^{-1}\lambda_q^0\zeta(d_u, d_v)$$

avec

$$\zeta(d_u, d_v) = e^{d_v\Delta t_k}\int_0^{\Delta t_k} e^{(d_u - d_v)t}dt = \begin{cases} \Delta t_k e^{d_u\Delta t_k} & \text{si } d_u = d_v \\ \dfrac{e^{d_u\Delta t_k} - e^{d_v\Delta t_k}}{d_u - d_v} & \text{si } d_u \neq d_v \end{cases}$$

Le résultat $\Phi^1=(Q^1,\Lambda^1)$ de cet algorithme devra être réinjecté pour aboutir à la convergence $\Phi^{i+1} - \Phi^i <$ erreur.

Cette méthode s'applique quelque soit N, le nombre d'état du processus MMPP. Nous l'avons implémenté pour N=2 et 4 états. De plus, il n'y a pas de condition sur la matrice de transition (processus de naissance et de mort).

b. Les valeurs initiales

La méthode décrite ci-dessus converge vers les valeurs réelles, mais, la vitesse de convergence (le nombre d'itération avant que la convergence soit atteinte) dépend fortement des valeurs initiales: si les valeurs initiales sont très lointaines des valeurs finales, le temps de convergence devient très long.

Par contre, la méthode décrite ci-dessous [51] ne s'applique qu'au MMPP-2. L'idée principale est de faire correspondre la distribution d'interarrivées des paquets à celle d'une fonction MMPP-2 et de tirer les paramètres les plus proches.

La distribution complémentaire du temps d'interarrivée X_i qui suit un processus MMPP 2 peut être tiré de [58] et est donnée par :

$$\begin{aligned} F_c(x) &\equiv \Pr(X_i > x) \\ &= pe^{(Q-\Lambda)x}(\Lambda - Q)^{-1}\Lambda \cdot 1 \\ &= qe^{-u_1 x} + (1-q)e^{-u_2 x} \end{aligned}$$

avec $0<q<1$, $\mathbf{1}$ la matrice colonne de un et:

$u_1 = \frac{\lambda_1 + \lambda_2 + \sigma_1 + \sigma_2 - \delta}{2}$, $u_2 = \frac{\lambda_1 + \lambda_2 + \sigma_1 + \sigma_2 + \delta}{2}$, $q = \frac{\lambda_2^2\sigma_1 + \lambda_1^2\sigma_2}{(\lambda_2\sigma_1 + \lambda_1\sigma_2)(u_1 - u_2)} - \frac{u_2}{u_1 - u_2}$ et

$\delta = \sqrt{(\lambda_1 - \lambda_2 + \sigma_1 - \sigma_2)^2 + 4\sigma_1\sigma_2}$

Les paramètres u_1 et u_2 sont les valeurs propres de $(\Lambda - Q)$.

La fonction d'autocovariance $C[k]$ est donnée pour $k \geq 1$ par :

$$\begin{aligned} C[k] &= E[(X_1 - E[X_1])(X_{k+1} - E[X_{k+1}])] \\ &= p(\Lambda - Q)^2 \Lambda [[(\Lambda - Q)^{-1}\Lambda]^{k-1} - 1p](\Lambda - Q)^{-2}\Lambda\, 1 \\ &= A\rho^k \end{aligned}$$

avec $\rho = \frac{\lambda_1\lambda_2}{\lambda_1\lambda_2 + \lambda_1\sigma_2 + \lambda_2\sigma_1}$ et $A = \frac{(\lambda_1 - \lambda_2)^2\sigma_1\sigma_2}{(\lambda_1\sigma_2 + \lambda_2\sigma_1)^2(\lambda_1\lambda_2 + \lambda_1\sigma_2 + \lambda_2\sigma_1)}$

Le paramètre ρ est une des valeurs propres de $(\Lambda - Q)^{-1}\Lambda$. La deuxième est égale à 1 du fait que la matrice est stochastique.

Si les paramètres u_1, u_2, q et ρ sont données, nous pouvons alors déterminer les quatre paramètres du processus MMPP-2 qui sont : λ_1, λ_2, σ_1 et σ_2.

Après quelques étapes de calculs, les équations suivantes peuvent être déduites:

$$\lambda_1 = \frac{1}{2}\left[q(1-\rho)(u_1 - u_2) + \rho u_1 + u_2 + \sqrt{\xi}\right]$$

$$\lambda_2 = \frac{u_1 u_2 \{\lambda_1 - q(u_1 - u_2) - u_2\}}{\lambda_1 u_1 - \lambda_1 q(u_1 - u_2) - u_1 u_2}$$

$$\sigma_1 = \frac{(u_1 - \lambda_1)(u_2 - \lambda_1)}{(\lambda_1 - \lambda_2)}$$

$$\sigma_2 = \frac{(\lambda_2 - u_1)(\lambda_1 + \sigma_1 - u_1)}{(u_1 - \lambda_1)}$$

avec

$$\zeta = [q(1-\sigma)(u_1 - u_2) + \sigma u_1 + u_2]^2 - 4\sigma u_1 u_2$$

Le temps interarrivée moyen est donc : $m \equiv E[X] = \frac{(1-q)u_1 + qu_2}{u_1 u_2}$

La pente initiale de $F_c(x)$ est :

$$\begin{aligned} s_0 &\equiv -\left.\frac{d \log F_c(x)}{dx}\right|_{x=0^+} \\ &= -\frac{1}{1 - F(x)}\left.\frac{d(1 - F(x))}{dx}\right|_{x=0^+} \\ &= f(0^+) \end{aligned}$$

$$= qu_1 + (1-q)u_2$$

Pour les exprimer en fonction de λ_1, λ_2, σ_1 et σ_2, m et s_0 s'expriment par :

$$m = \frac{1}{\pi\Lambda e} = \frac{\sigma_1 + \sigma_2}{\lambda_2\sigma_1 + \lambda_1\sigma_2} \text{ et } s_0 = p\Lambda e = \frac{\lambda_2^2\sigma_1 + \lambda_1^1\sigma_2}{\lambda_2\sigma_1 + \lambda_1\sigma_2}$$

Le paramètre u_1 détermine la pente de $F_c(x)$ pour les $x \to \infty$, alors que u_2 détermine la pente pour les faibles valeurs de x.

Nous considérons les paramètres m, s_0, u_2 et σ pour déterminer $F_c(x)$. Les équations ci-dessus deviennent :

$$\lambda_1 = \frac{s_0 m u_2(1-\rho) + m\rho u_2^2 - s_0 + \sqrt{\kappa}}{2(mu_2 - 1)}$$

$$\lambda_2 = \frac{s_0 - \lambda_1}{1 - \lambda_1 m}$$

$$\sigma_1 = \frac{\lambda_1\lambda_2(1-\rho)(\lambda_1 m - 1)}{\rho(\lambda_1 - \lambda_2)}$$

$$\sigma_2 = \frac{\sigma_1(1 - \lambda_2 m)}{\lambda_1 m - 1}$$

avec

$$\kappa = \{s_0 m u_2(1-\rho) + m\rho u_2^2 - s_0\}^2 - 4\rho u_2(mu_2 - 1)(u_2 - s_0)$$

Pour déterminer ces valeurs, nous procédons comme suit :

Pour la moyenne :

$$m = \sum_{k=1}^{N} xF_c(x)$$

Pour la s_0:

$$s_0 \equiv -\left.\frac{d\log F_c(x)}{dx}\right|_{x=0^+}$$

s_0 est la dérivée en 0^+ du logarithme de la fonction $F_c(x)$. Pour déterminer cette valeur, il suffit de trouver la pente de Fc(x) en 0^+. Donc :

$$s_0 \equiv -\left.\frac{d\log F_c(x)}{dx}\right|_{x=0^+} = -\frac{\Delta\log F_c(x)}{\Delta x} = -\frac{\log F_c(x) - \log F_c(x - \Delta x)}{\Delta x}$$

Il reste à déterminer la valeur de 0^+. Pour minimiser l'influence du hasard, nous avons choisi la valeur 0^+ de telle sorte qu'à 0^+ il y ait 10 échantillons.

En prenant $\Delta(x) = 0 - 0^+ = -x_0^+$, nous pouvons écrire:

$$\Delta\ln(F_c(x)) = \ln(F_c(0)) - \ln(F_c(x_0^+))$$

$$\Delta\ln(F_c(x)) = \ln(1) - \ln(F_c(x_0^+)) = -\ln(F_c(x_0^+))$$

Ce qui donne: $s_0 = \frac{\log F_c(x_0)}{x_0}$

Pour déterminer u_2:

Sachant que $u_1 < u_2$, $F_c(x)$ s'écrit :

$$F_c(x) = qe^{-u_1 x} + (1-q)e^{-u_2 x}$$

$$\frac{\delta F_c(x)}{\delta x} = -u_1 qe^{-u_1 x} - u_2(1-q)e^{-u_2 x}$$

Pour $x \to \infty; \frac{\delta F_c(x)}{\delta x} \to -u_1 qe^{-u_1 x}$

Pour $x \to 0; \frac{\delta F_c(x)}{\delta x} = -u_2(1-q)e^{-u_2 x}$

Alors, pour déterminer u_1, il suffit de déterminer $\frac{\delta F_c(x)}{\delta x}$ pour $x \to \infty$. De même, pour déterminer u_2, il suffit de déterminer $\frac{\delta F_c(x)}{\delta x}$ pour $x \to 0$.

Or, la probabilité de x grande est faible, alors que celle pour les x faibles ne l'est pas. Donc, il est plus prudent de trouver $\frac{\delta F_c(x)}{\delta x}$ pour les faibles valeurs de x et ainsi de trouver u_2.

Pour x_1 et x_2 deux valeurs différentes mais toutes les deux proches de zéro:

$$\frac{\delta F_c(x_1)}{\delta x} = -u_2(1-q)e^{-u_2 x_1} \text{ et } \frac{\delta F_c(x_2)}{\delta x} = -u_2(1-q)e^{-u_2 x_2}$$

Ce qui donne $\frac{\delta F_c(x_1)}{\delta F_c(x_2)} = e^{-u_2(x_1-x_2)}$

Alors u_2 devient:

$$u_2 = \frac{-1}{(x_1-x_2)} \ln\left(\frac{\delta F_c(x_1)}{\delta F_c(x_2)}\right)$$

Or,

$$\left.\frac{\delta F_c(x)}{\delta x}\right|_{x \to x_1} = \frac{F_c(x_1+\delta(x_1)) - F_c(x_1-\delta(x_1))}{(x_1+\delta(x_1)) - (x_1-\delta(x_1))} = \frac{F_c(x_1+\delta(x_1)) - F_c(x_1-\delta(x_1))}{2\delta(x_1)}$$

et

$$\left.\frac{\delta F_c(x)}{\delta x}\right|_{x \to x_2} = \frac{F_c(x_2+\delta(x_2)) - F_c(x_2-\delta(x_2))}{2\delta(x_2)}$$

Pour ρ :

$$C[k] = A\rho^k = E[(X_1 - E(X_1))(X_{k+1} - E(X_{k+1}))]$$

En supposant le régime stationnaire:

$$C[k] = A\rho^k = E[(X_1 - E(X))(X_{k+1} - E(X))] = E[(X_1 - m)(X_{k+1} - m)]$$

Pour trouver ρ nous rappelons que:

$C[k] = A\rho^k$ et $C[k+1] = A\rho^{k+1}$ ce qui implique que $\rho = \frac{C[k+1]}{C[k]}$

Or le calcul des covariances n'est pas très exact. Pour pallier à cela, nous avons effectué une moyenne de ρ pour différents k, c'est-à-dire:

$$\rho_1 = \frac{C[k+1]}{C[k]}, \rho_2 = \sqrt{\frac{C[k+2]}{C[k]}}, \rho_3 = \frac{C[k+2]}{C[k+1]}, \rho_4 = \sqrt[3]{\frac{C[k+3]}{C[k]}}, \rho_5 = \sqrt{\frac{C[k+3]}{C[k+1]}}, \rho_6 = \frac{C[k+3]}{C[k+2]} \ldots$$

et puis la moyenne de ces différentes valeurs.

6. Exemple d'estimation

Pour vérifier l'exactitude de l'implémentation, nous avons procédé à une vérification. En générant une suite de paquets dont le temps inter arrivée suit une loi MMPP 2 de paramètres connus, nous appliquons l'algorithme sur l'échantillon en question et nous obtenons les paramètres estimés.

Deux cas sont illustrés ici bas: ils sont résumés dans le tableau ci-dessous.

Tableau 60: Estimation des paramètres

	λ_1			λ_2			σ_1			σ_2		
	Réelle	Estimé	Erreur %	Réelle	Estimé	Erreur %	Réelle	Estimé	Erreur %	Réelle	Estimé	Erreur %
Cas 1	100	*101,48*	1,48	10	*9,48*	5,2	10	*10,09*	0,9	1	*1,24*	24
Cas 2	10	*10,17*	1,7	1	*0,94*	6	0,5	*0,42*	16	0,2	*0,22*	10

La Figure 70 montre la vitesse de convergence de la méthode proposée pour le cas 1. Après 9 itérations, λ_1, λ_2 et σ_1 sont proches à moins de 6% des valeurs théoriques. Seul σ_2 s'éloigne de la valeur qu'elle doit prendre. L'erreur de σ_2 est de 24%, plausible.

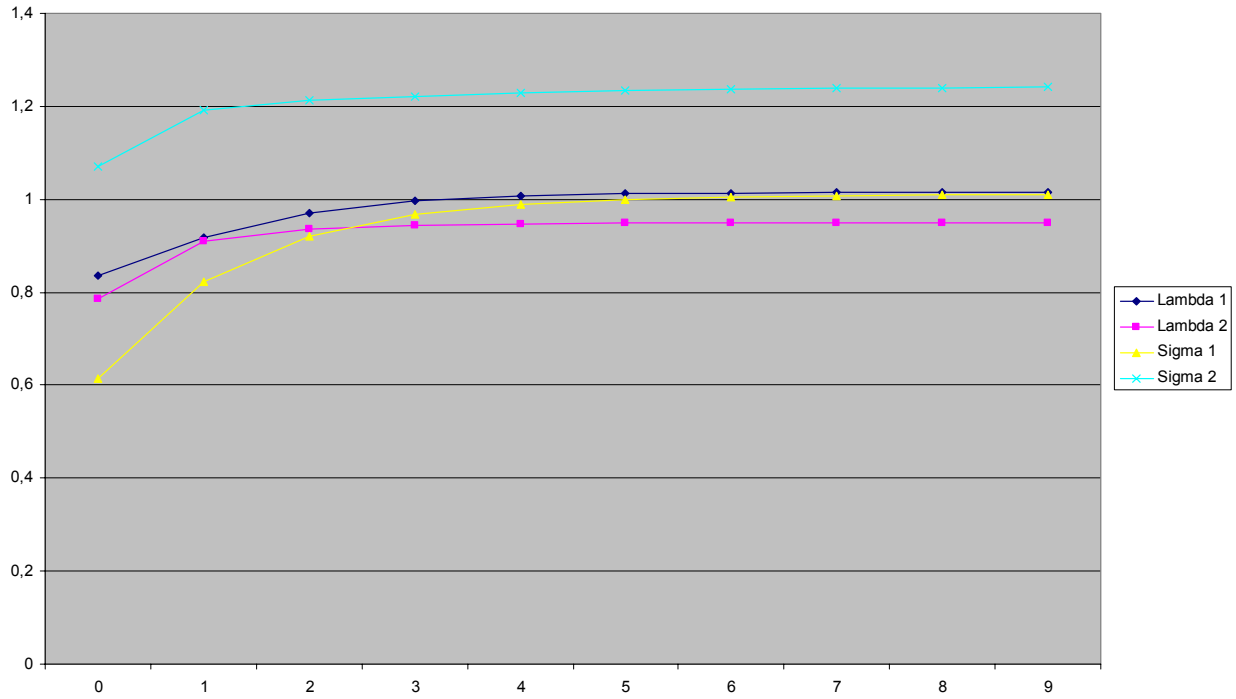


Figure 70: Convergence du cas 1

La Figure 71 montre la vitesse de convergence de la méthode proposée pour le cas 2. Après 7 itérations λ_1, λ_2 et σ_1 et σ_2 sont proches à moins de 15% des valeurs théoriques.

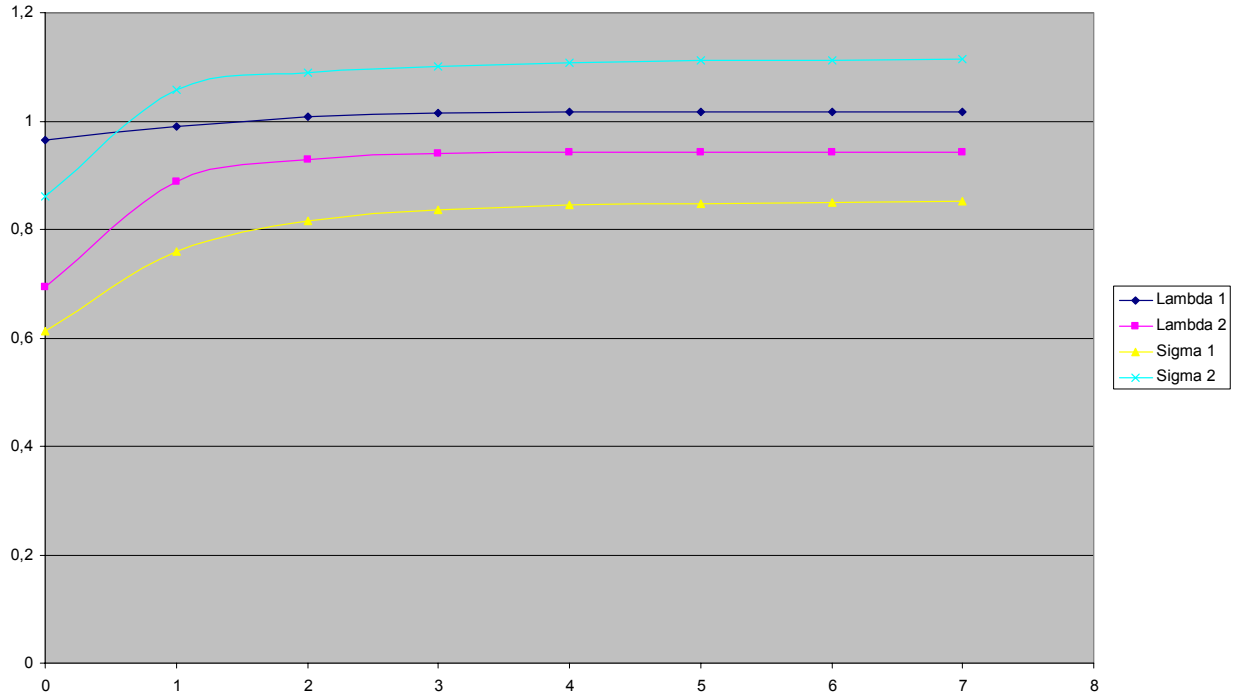


Figure 71: Convergence du cas 2

ANNEXE 2 :
COMPLEXITE DES ALGORITHMES

Nous développons ci-dessous la complexité des deux algorithmes d'estimation de la densité du délai à partir de l'observation du délai de bout en bout.

Comme nous l'avons présenté plus haut, les deux étapes de l'estimation s'écrivent:

Etape E: $\hat{m}_{ij} = \sum_{k=1}^{N} p^{l}(z_i(k) = j / y_1(k), y_2(k))$

Etape M: $\hat{p}_{ij} = \frac{m_{ij}}{\sum_{l=1}^{L} m_{il}}$.

En fixant:

N_ech, le nombre d'échantillons total de l'observation.

N, le nombre d'échantillons pendant chaque fenêtre d'estimation.

L, le nombre des nouveaux échantillons entre deux estimations.

Le nombre de réalisations de chaque algorithme est déduit par: $\frac{N_ech - N}{L} + 1$.

Soient I, le nombre de tronçons présents dans le réseau et J, le nombre des valeurs des délais discrétisés fixés.

1. La méthode complète

Chaque $\hat{m}_{ij}$ nécessite une addition entre N éléments. Donc, pour avoir tous les $\hat{m}_{ij}$, nous avons besoin de $I \times J$ opérations. Ce qui donne pour l'étape E, $I \times J \times N$ additions.

Comme la valeur $\sum_{l=1}^{L} m_{il}$ est la même pour chaque i, nous aurons besoin de $I \times J$ additions et de $I \times J$ divisions pour déterminer les probabilités à l'issu de l'étape M. Puisqu'il s'agit d'un algorithme itératif, les étapes E et M se succèdent jusqu'à aboutir à la convergence avec la précision souhaitée. Cela prendra X itérations.

La complexité totale de la réalisation de l'algorithme est donc:

$C_c = X(I \times J \times N \text{ additions} + I \times J \text{ divisions})$

$C_c = X \times I \times J(N \text{ additions} + 1 \text{ division})$

2. La méthode simple

Chaque $\hat{m}_{ij}$ nécessite uniquement une addition et une soustraction. Donc, pour avoir tous les $\hat{m}_{ij}$, nous avons besoin de $I \times J$ opérations d'addition et $I \times J$ opérations de soustraction. Ce qui donne pour l'étape E, $I \times J \times 2$ additions en supposant qu'une addition et une soustraction sont équivalentes en complexité numérique.

La détermination de la probabilité requiert les mêmes opérations que dans la méthode dite complète. Par contre, dans ce schéma, ces opérations sont réalisées L fois pour aboutir, après chaque intervalle de temps, aux probabilités souhaitées.

La complexité totale de la réalisation de l'algorithme s'écrit alors:

$C_s = L(I \times J \times 2 \text{ additions} + I \times J \text{ divisions})$

$$C_s = L \times I \times J(2\,\text{additions} + 1\,\text{division})$$

Le gain en complexité est donc:

$$\frac{C_c}{C_s} = \frac{X \times I \times J(N\,\text{additions} + 1\,\text{division})}{L \times I \times J(2\,\text{additions} + 1\,\text{division})} = \frac{X(N\,\text{additions} + 1\,\text{division})}{L(2\,\text{additions} + 1\,\text{division})}$$

La valeur de N est relativement élevée: dans les différents schémas nous l'avons pris égale à 100, pour pouvoir tirer une certaine probabilité sur un intervalle significatif. La valeur de L, dépend quant à elle de la rapidité de la réactivité souhaitée face au changement des caractéristiques du lien. Ce choix est celui de l'administrateur du réseau. Nous avons pris la valeur 20 dans nos différentes configurations. La valeur de X n'a pas de règle, mais, l'expérience montre généralement que nous avons besoin d'une trentaine d'itérations pour aboutir à la convergence avec la précision relative de 1%.

Numériquement, le gain de l'algorithme simple s'exprime par:

$$\frac{C_c}{C_s} = \frac{X(N\,\text{additions} + 1\,\text{division})}{L(2\,\text{additions} + 1\,\text{division})} = \frac{30(100\,\text{additions} + 1\,\text{division})}{20(2\,\text{additions} + 1\,\text{division})} = \frac{3}{2}\frac{(100\,\text{additions} + 1\,\text{division})}{(2\,\text{additions} + 1\,\text{division})}$$

$$\frac{C_c - C_s}{C_s} = \frac{X(N\,\text{additions} + 1\,\text{division}) - L(2\,\text{additions} + 1\,\text{division})}{L(2\,\text{additions} + 1\,\text{division})} = \frac{(N \times X - 2 \times L)\text{additions} + (X - L)\text{divisions}}{L(2\,\text{additions} + 1\,\text{division})}$$

$$\frac{C_c - C_s}{C_s} = \frac{30(100\,\text{additions} + 1\,\text{division}) - 20(2\,\text{additions} + 1\,\text{division})}{20(2\,\text{additions} + 1\,\text{division})}$$

$$\frac{C_c - C_s}{C_s} = \frac{2960\text{additions} + 10\text{divisions}}{40\text{additions} + 20\text{divisions}}$$

Soit 75 fois plus d'additions et 1,5 fois plus de divisions.

ANNEXE 3 :
EXEMPLE DE CALCUL DES PROBABILITES STATIONNAIRES POUR L'ORDONNANCEUR WRR

Pour le cas le plus simple, avec les valeurs N_1, N_2 et N_3 égales chacune à 1 (au plus, un seul paquet d'une classe peut attendre dans la file), le système de 27 équations se réduit à un système de 8 équations à 8 inconnus. Ces équations sont exposées dans le Tableau 61.

Tableau 61: Les équations de Kolmogorov pour le système avec N_1=1, N_2=1 et N_3=1.

Classe			Equations d'équilibre
1	2	3	
0	*0*	*0*	$(\lambda_1+\lambda_2+\lambda_3)P(0,0,0)=\mu_0P(1,0,0)+\mu_0P(0,1,0)+\mu_0P(0,0,1)$
0	*0*	N_3	$(\lambda_1+\lambda_2+\mu_0)P(0,0,1)=\mu_{13}P(1,0,1)+\mu_{23}P(0,1,1)+\lambda_3P(0,0,0)$
N_1	*0*	*0*	$(\lambda_2+\lambda_3+\mu_0)P(1,0,0)=\lambda_1P(0,0,0)+\mu_{21}P(1,1,0)+\mu_{31}P(1,0,1)$
N_1	*0*	N_3	$(\lambda_2+\mu_{13}+\mu_{31})P(1,0,1)=\lambda_1P(0,0,1)+\mu_2P(1,1,1)+\lambda_3P(1,0,0)$
0	N_2	*0*	$(\lambda_1+\lambda_3+\mu_0)P(0,1,0)=\mu_{12}P(1,1,0)+\lambda_2P(0,0,0)+\mu_{32}P(0,1,1)$
0	N_2	N_3	$(\lambda_1+\mu_{23}+\mu_{32})P(0,1,1)=\mu_1P(1,1,1)+\lambda_2P(0,0,1)+\lambda_3P(0,1,0)$
N_1	N_2	*0*	$(\lambda_3+\mu_{12}+\mu_{21})P(1,1,0)=\lambda_1P(0,1,0)+\lambda_2P(1,0,0)+\mu_3P(1,1,1)$
N_1	N_2	N_3	$(\mu_1+\mu_2+\mu_3)P(1,1,1)=\lambda_1P(0,1,1)+\lambda_2P(1,0,1)+\lambda_3P(1,1,0)$

Nous résolvons littéralement ces équations sous Maple® 7.0.

P(0,0,0) admet 1906 additions, dont 1570 forment le dénominateur et 336 le numérateur. Chaque élément d'addition au numérateur est une multiplication de 6 paramètres (soit un polynôme de puissance 6) et chaque élément du dénominateur est formé d'une multiplication de 7 paramètres. La différence de puissance entre le numérateur et le dénominateur se trouve dans le fait qu'au numérateur, le paramètre μ_0 est factorisé. Cela nous ramènera à une fraction dont le numérateur et le dénominateur auront la même puissance.

P(0,0,1) admet 640 additions, dont 304 forment le numérateur et 336 le dénominateur. Chaque élément d'addition au numérateur et au dénominateur est une multiplication de 6 paramètres (un polynôme de puissance 6). Le facteur λ_3 au numérateur et μ_0 au dénominateur complètent la fraction.

Chaque probabilité stationnaire peut donc s'écrire:

$$\frac{\sum_{m=0}^{4}\sum_{l=0}^{4}\sum_{k=0}^{4}\sum_{j=0}^{4}\sum_{i=0}^{4}\sum_{h=0}^{4}\sum_{g=0}^{4}\sum_{f=0}^{4}\sum_{e=0}^{4}\sum_{d=0}^{4}\sum_{c=0}^{4}\sum_{b=0}^{4}\sum_{a=0}^{4}\left(\begin{matrix}N(a,b,c,d,e,f,g,h,i,j,k,l,m)\cdot\lambda_1^a\cdot\lambda_2^b\cdot\lambda_3^c\cdot\mu_0^d\\ \cdot\mu_1^e\cdot\mu_2^f\cdot\mu_3^g\cdot\mu_{12}^h\cdot\mu_{13}^i\cdot\mu_{21}^j\cdot\mu_{23}^k\cdot\mu_{31}^l\cdot\mu_{32}^m\end{matrix}\right)}{\sum_{m=0}^{4}\sum_{l=0}^{4}\sum_{k=0}^{4}\sum_{j=0}^{4}\sum_{i=0}^{4}\sum_{h=0}^{4}\sum_{g=0}^{4}\sum_{f=0}^{4}\sum_{e=0}^{4}\sum_{d=0}^{4}\sum_{c=0}^{4}\sum_{b=0}^{4}\sum_{a=0}^{4}\left(\begin{matrix}D(a,b,c,d,e,f,g,h,i,j,k,l,m)\cdot\lambda_1^a\cdot\lambda_2^b\cdot\lambda_3^c\cdot\mu_0^d\\ \cdot\mu_1^e\cdot\mu_2^f\cdot\mu_3^g\cdot\mu_{12}^h\cdot\mu_{13}^i\cdot\mu_{21}^j\cdot\mu_{23}^k\cdot\mu_{31}^l\cdot\mu_{32}^m\end{matrix}\right)}$$

Sachant que nous avons toujours $a+b+c+d+e+f+g+h+i+j+k+l+m=7$ et les constantes N*(a,b,c,d,e,f,g,h,i,j,k,l,m)* et D*(a,b,c,d,e,f,g,h,i,j,k,l,m)* pouvant être nulles pour des valeurs données.

ANNEXE 4 :
RESULTATS COMPLEMENTAIRES POUR L'ORDONNANCEUR WRR

1. Les chaînes incluses

Nous illustrons ici les différentes chaînes incluses issues du modèle analytique d'un ordonnanceur WRR avec des temps inter arrivées et un temps de service tous les deux exponentiels.

Sur les 6 plans:

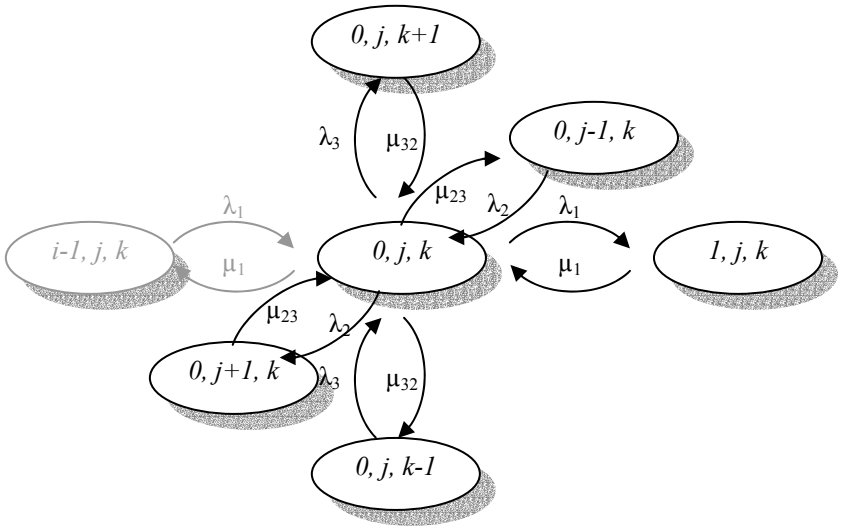


Figure 72: La chaîne incluse dans le plan *i=0, j et k ∈{1..N-1}*.

$$(\lambda_1+\lambda_2+\lambda_3+\mu_{23}+\mu_{32})P(0,j,k)=\mu_1P(1,j,k)+\mu_{23}P(0,j+1,k)+\lambda_2P(0,j-1,k)$$
$$+\mu_{32}P(0,j,k+1)+\lambda_3P(0,j,k-1)$$

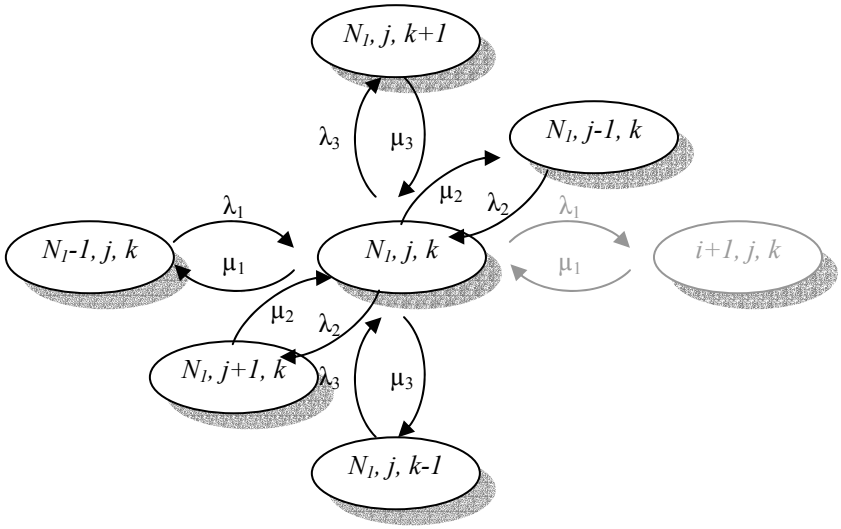


Figure 73: La chaîne incluse dans le plan *i=N1, j et k ∈{1..N-1}*.

$$(\lambda_2+\lambda_3+\mu_1+\mu_2+\mu_3)P(N_1,j,k)=\lambda_1P(N_1-1,j,k)+\mu_2P(N_1,j+1,k)+\lambda_2P(N_1,j-1,k)+\mu_3P(N_1,j,k+1)+\lambda_3P(N_1,j,k-1)$$

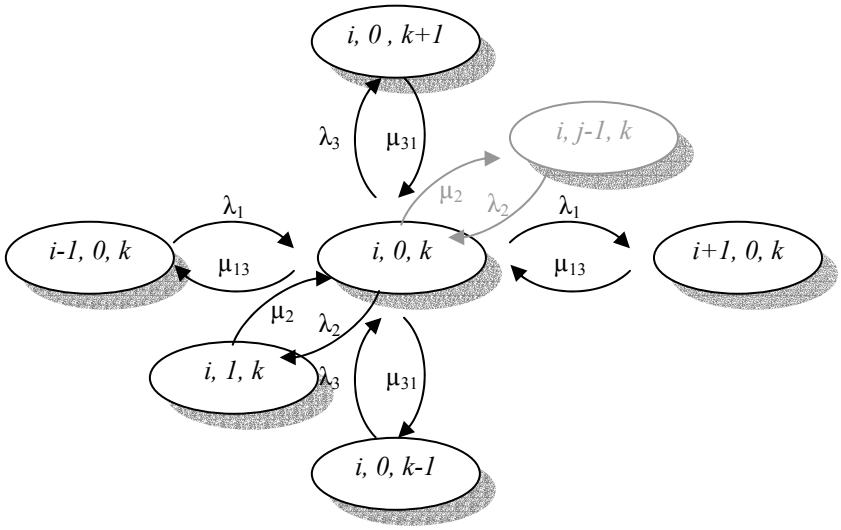


Figure 74: La chaîne incluse dans le plan j=*0, i et k ∈{1..N-1}*.

$$(\lambda_1+\lambda_2+\lambda_3+\mu_{13}+\mu_{31})P(i,0,k)=\mu_{13}P(i+1,0,k)+\lambda_1P(i-1,0,k)+\mu_2P(i,1,k)+\mu_{31}P(i,0,k+1)+\lambda_3P(i,0,k-1)$$

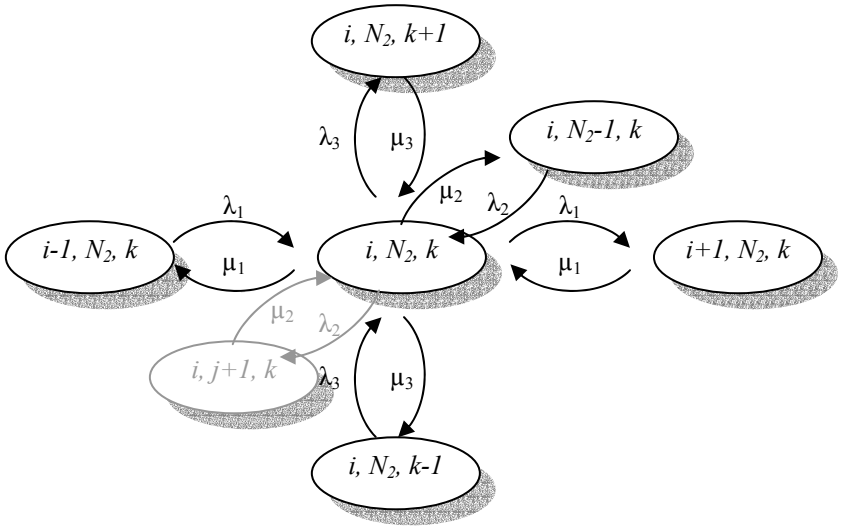


Figure 75: La chaîne incluse dans le plan j=N_2, *i et k* $\in$*{1..N-1}.*

$$(\lambda_1+\lambda_3+\mu_1+\mu_2+\mu_3)P(i,N_2,k)=\mu_1P(i+1,N_2,k)+\lambda_1P(i-1,N_2,k)+\lambda_2P(i,N_2-1,k)+\mu_3P(i,N_2,k+1)+\lambda_3P(i,N_2,k-1)$$

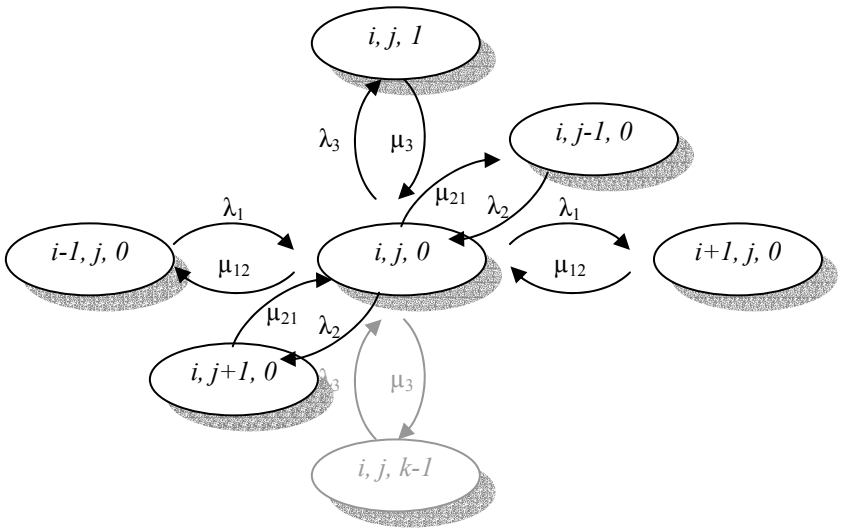


Figure 76: La chaîne incluse dans le plan k=*0, i et j* $\in$*{1..N-1}.*

$$(\lambda_1+\lambda_2+\lambda_3+\mu_{12}+\mu_{21})P(i,j,0)=\mu_{12}P(i+1,j,0)+\lambda_1P(i-1,j,0)+\mu_{21}P(i,j+1,0)+\lambda_2P(i,j-1,0)+\mu_3P(i,j,1)$$

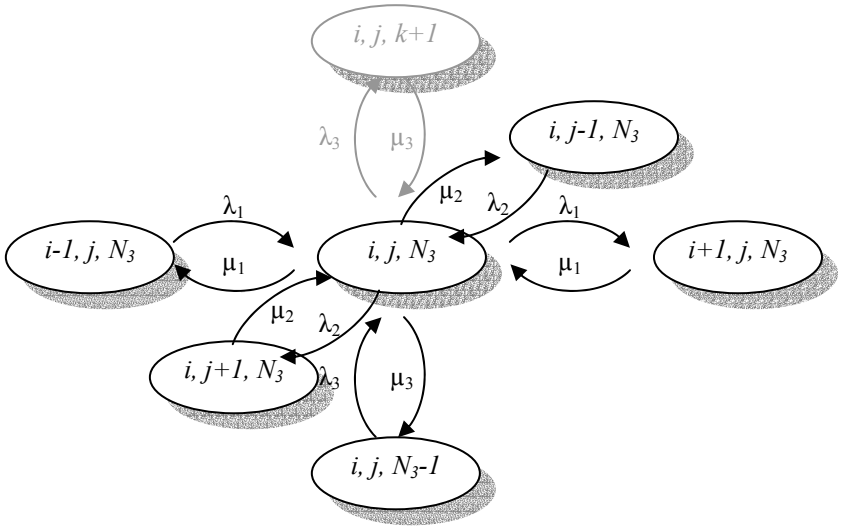


Figure 77: La chaîne incluse dans le plan k=N_3, i et j $\in${1..N-1}.

$$(\lambda_1+\lambda_2+\mu_1+\mu_2+\mu_3)P(i,j,N_3)=\mu_1P(i+1,j,N_3)+\lambda_1P(i-1,j,N_3)+\mu_2P(i,j+1,N_3)+\lambda_2P(i,j-1,N_3)+\lambda_3P(i,j,N_3-1)$$

Sur les 12 lignes

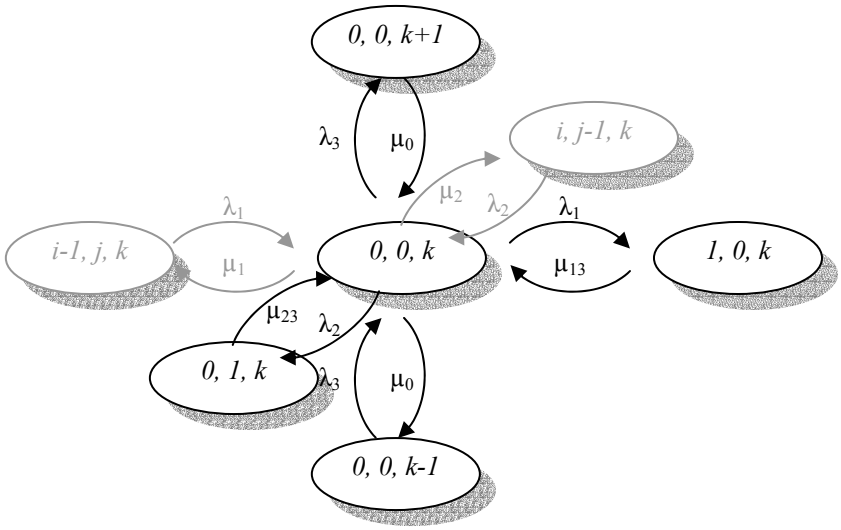


Figure 78: La chaîne incluse sur la droite *i=0, j=0 et k* $\in${1..N_3-1}.

$$(\lambda_1+\lambda_2+\lambda_3+\mu_0)P(0,0,k)=\mu_{13}P(1,0,k)+\mu_{23}P(0,1,k)+\mu_0P(0,0,k+1)+\lambda_3P(0,0,k-1)$$

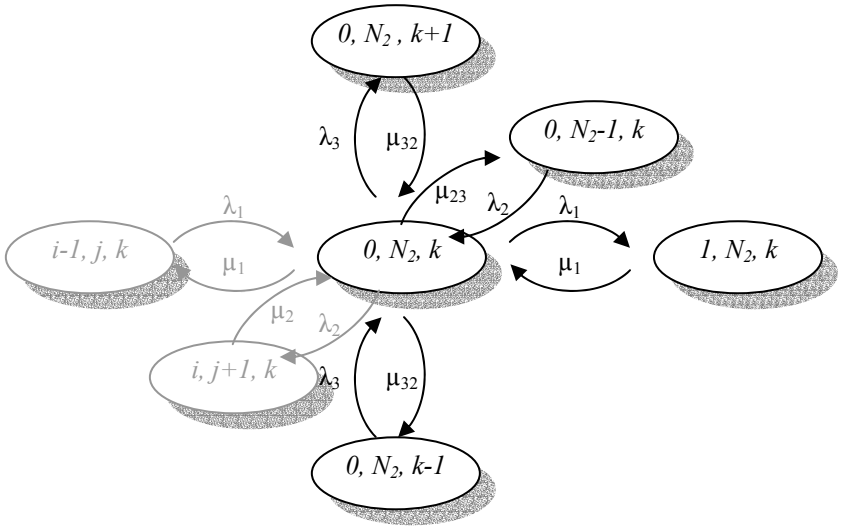


Figure 79: La chaîne incluse sur la droite *i=0, j=N_2 et k ∈{1..N_3-1}*.

$$(\lambda_1 + \lambda_3 + \mu_{23} + \mu_{32})P(0, N_2, k) = \mu_1 P(1, N_2, k) + \lambda_2 P(0, N_2 - 1, k) + \mu_{32} P(0, N_2, k+1) + \lambda_3 P(0, N_2, k-1)$$

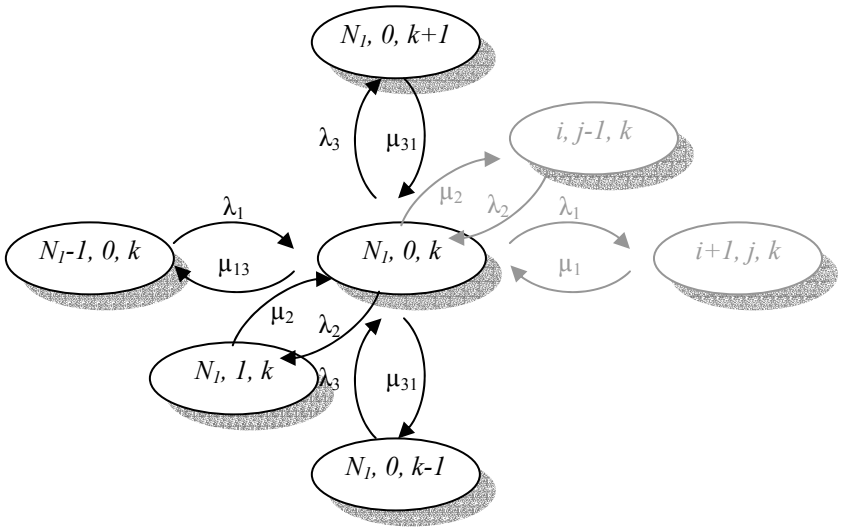


Figure 80: La chaîne incluse sur la droite *i=N_1, j=0 et k ∈{1..N_3-1}*.

$$(\lambda_2 + \lambda_3 + \mu_{13} + \mu_{31})P(N_1, 0, k) = \lambda_1 P(N_1 - 1, 0, k) + \mu_2 P(N_1, 1, k) + \mu_{31} P(N_1, 0, k+1) + \lambda_3 P(N_1, 0, k-1)$$

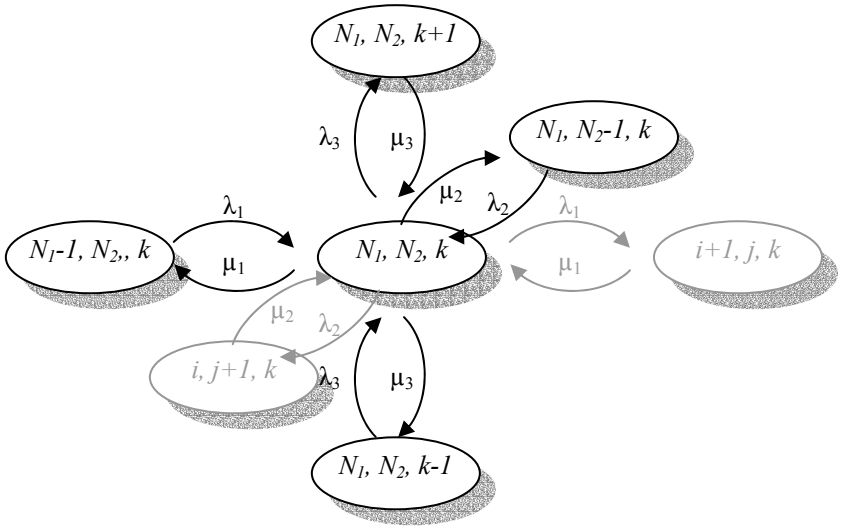


Figure 81: La chaîne incluse sur la droite *i=N1, j=N2 et k ∈{1..N3-1}*.

$$(\lambda_3+\mu_1+\mu_2+\mu_3)P(N_1,N_2,k)=\lambda_1P(N_1-1,j,k)+\lambda_2P(N_1,N_2-1,k)+\mu_3P(N_1,N_2,k+1)+\lambda_3P(N_1,N_2,k-1)$$

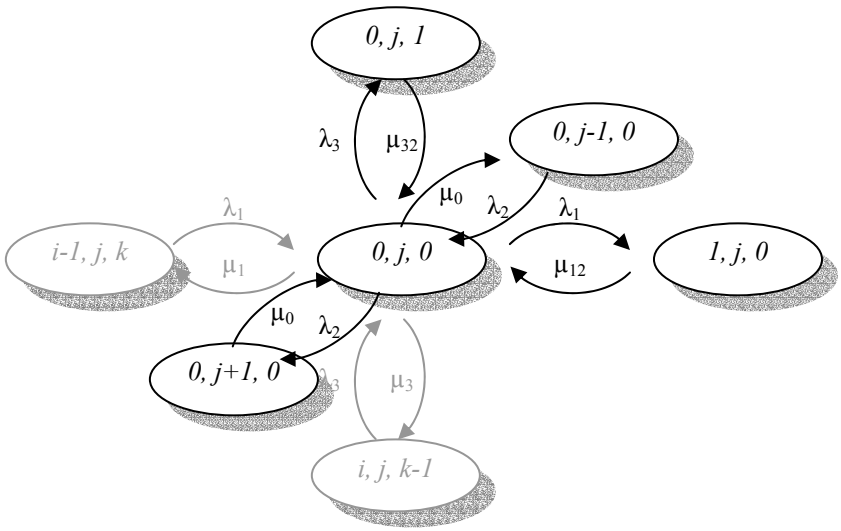


Figure 82: La chaîne incluse sur la droite *i=0, j∈{1..N2-1} et k=0*.

$$(\lambda_1+\lambda_2+\lambda_3+\mu_0)P(0,j,0)=\mu_{12}P(1,j,0)+\mu_0P(0,j+1,0)+\lambda_2P(i,j-1,0)+\mu_{32}P(0,j,1)$$

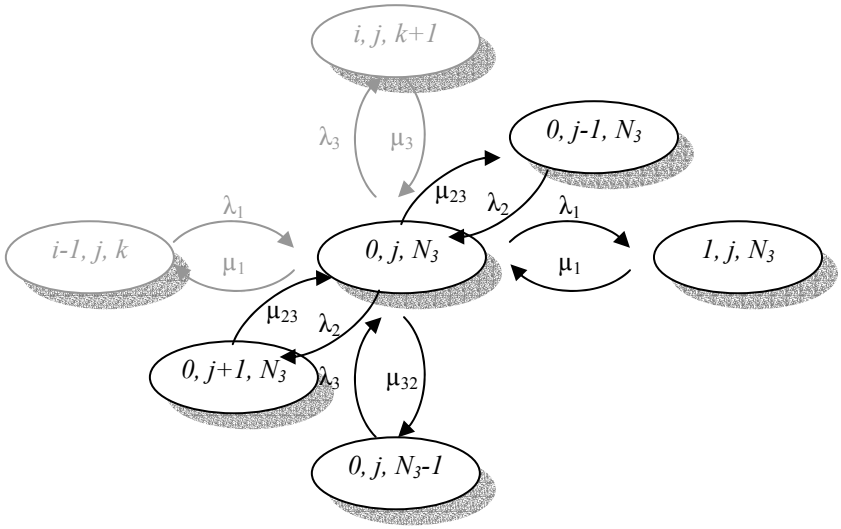


Figure 83: La chaîne incluse sur la droite *i=0, j∈{1..N₂-1} et k=N₃.*

$$(\lambda_1 + \lambda_2 + \mu_{23} + \mu_{32})P(0, j, N_3) = \mu_1 P(1, j, N_3) + \mu_{23} P(0, j+1, N_3) + \lambda_2 P(0, j-1, N_3) + \lambda_3 P(0, j, N_3 - 1)$$

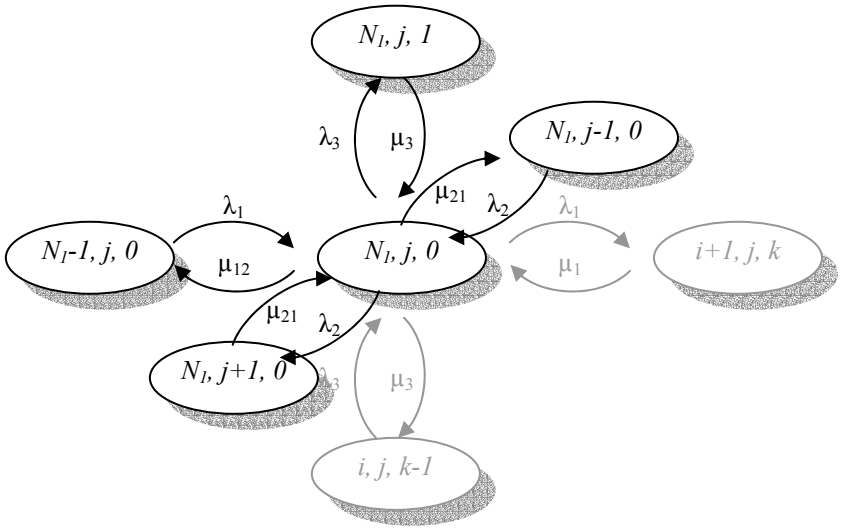


Figure 84: La chaîne incluse sur la droite *i=N₁, j∈{1..N₂-1} et k=0.*

$$(\lambda_2 + \lambda_3 + \mu_{12} + \mu_{21})P(N_1, j, 0) = \lambda_1 P(N_1 - 1, j, 0) + \mu_{21} P(N_1, j+1, 0) + \lambda_2 P(N_1, j-1, 0) + \mu_3 P(N_1, j, 1)$$

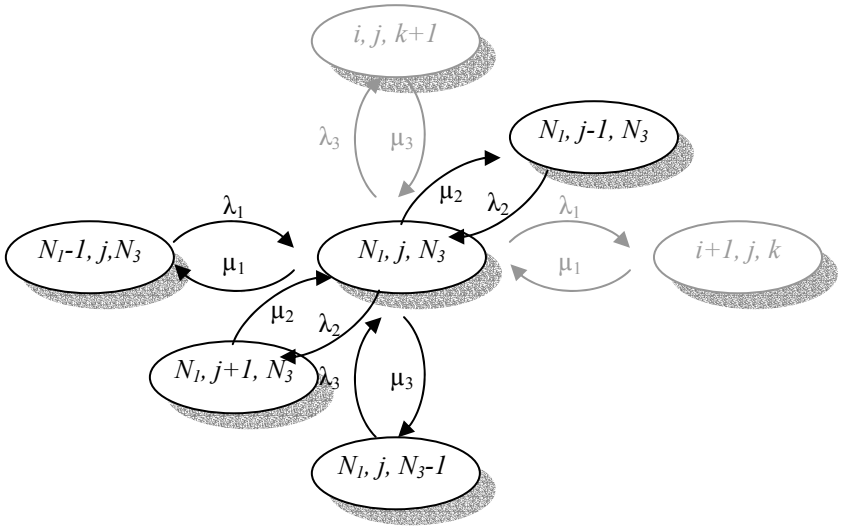


Figure 85: La chaîne incluse sur la droite *i=N1, j∈{1..N2-1} et k=N3*.

$$(\lambda_2+\mu_1+\mu_2+\mu_3)P(N_1,j,N_3)=\lambda_1 P(N_1-1,j,N_3)+\mu_2 P(N_1,j+1,N_3)+\lambda_2 P(N_1,j-1,N_3)+\lambda_3 P(N_1,j,N_3-1)$$

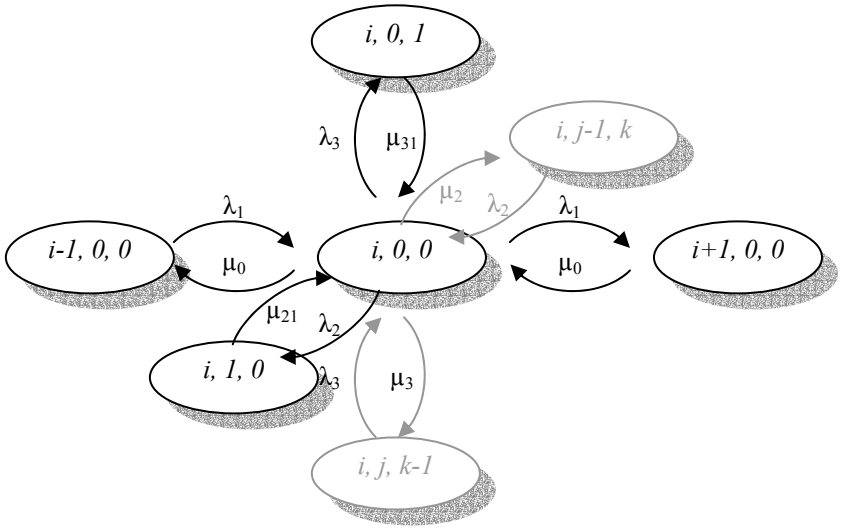


Figure 86: La chaîne incluse sur la droite *i∈{1..N1-1}, j=0 et k=0*.

$$(\lambda_1+\lambda_2+\lambda_3+\mu_0)P(i,0,0)=\mu_0 P(i+1,0,0)+\lambda_1 P(i-1,0,0)+\mu_{21}P(i,1,0)+\mu_{31}P(i,0,1)$$

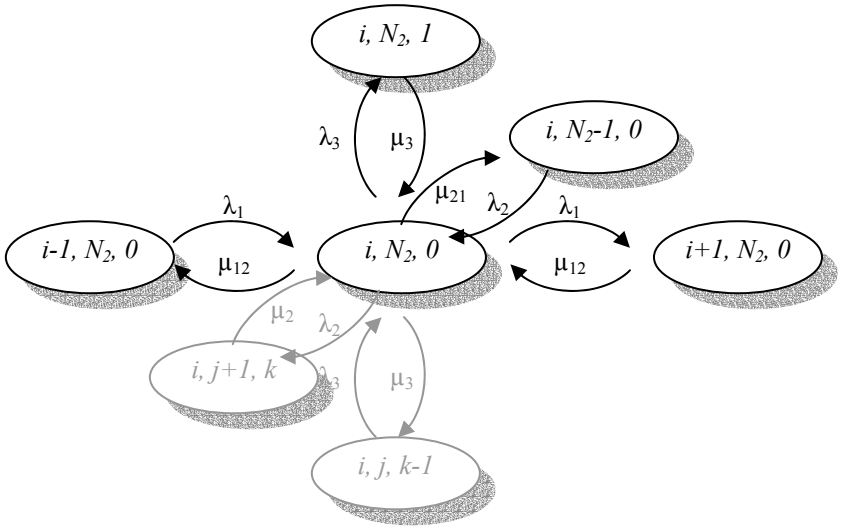


Figure 87: La chaîne incluse sur la droite *i∈{1..N1-1}, j=N2 et k=0.*

$$(\lambda_1+\lambda_3+\mu_{12}+\mu_{21})P(i,N_2,0)=\mu_{12}P(i+1,N_2,0)+\lambda_1P(i-1,N_2,0)+\lambda_2P(i,N_2-1,0)+\mu_3P(i,N_2,1)$$

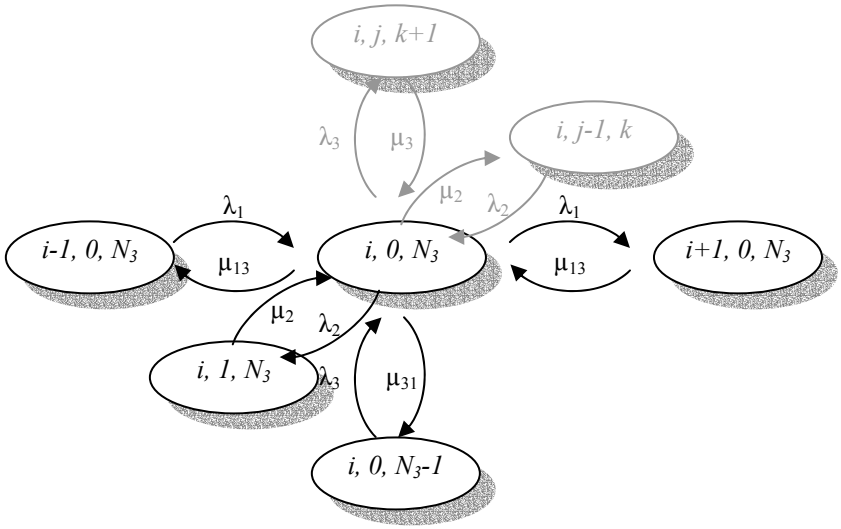


Figure 88: La chaîne incluse sur la droite *i∈{1..N1-1}, j=0 et k=N3.*

$$(\lambda_1+\lambda_2+\mu_{13}+\mu_{31})P(i,0,N_3)=\mu_{13}P(i+1,0,N_3)+\lambda_1P(i-1,0,N_3)+\mu_2P(i,1,k)+\lambda_3P(i,0,N_3-1)$$

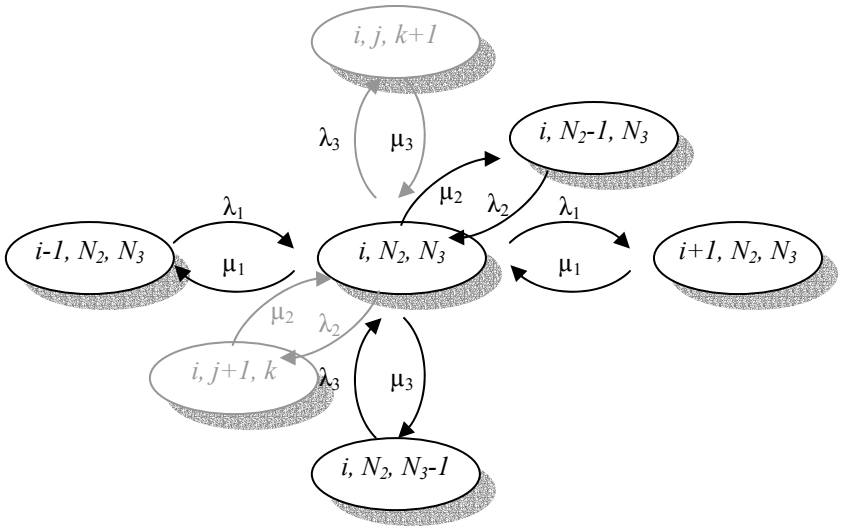


Figure 89: La chaîne incluse sur la droite *i∈{1..N1-1}, j=N2 et k=N3*.

$$(\lambda_1+\mu_1+\mu_2+\mu_3)P(i,N_2,N_3)=\mu_1P(i+1,N_2,N_3)+\lambda_1P(i-1,N_2,N_3)+\lambda_2P(i,N_2-1,N_3)+\lambda_3P(i,N_2,N_3-1)$$

Sur les points

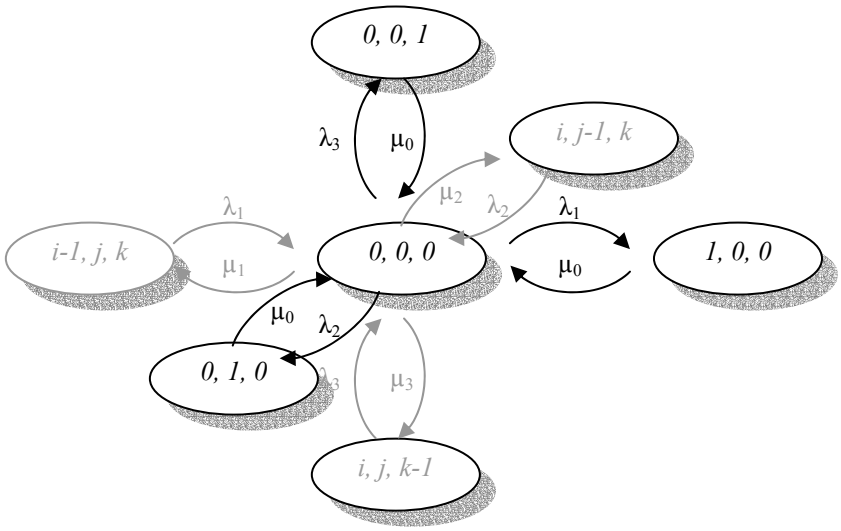


Figure 90: La chaîne incluse au point *i=0, j=0 et k=0.*

$$(\lambda_1+\lambda_2+\lambda_3)P(0,0,0)=\mu_0P(1,0,0)+\mu_0P(0,1,0)+\mu_0P(0,0,1)$$

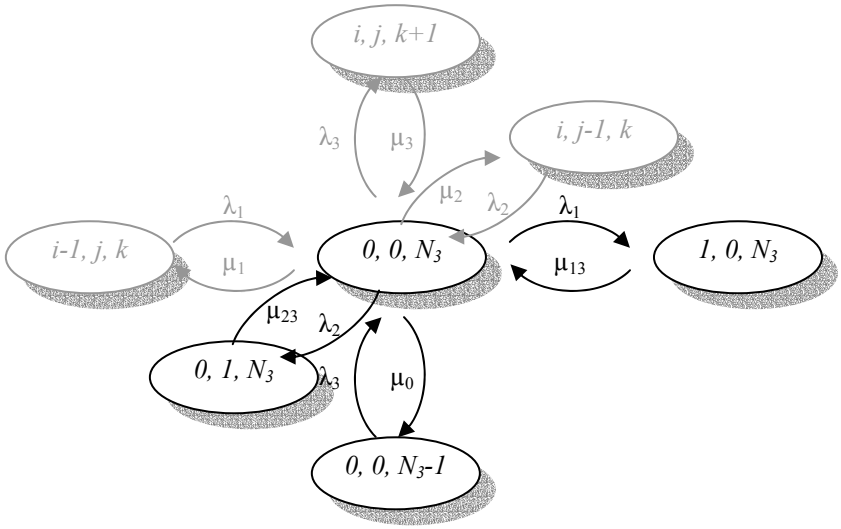


Figure 91: La chaîne incluse au point *i=0, j=0 et k=N_3*.

$$(\lambda_1 + \lambda_2 + \mu_0)P(0,0,N_3) = \mu_{13}P(1,0,N_3) + \mu_{23}P(0,1,N_3) + \lambda_3 P(0,0,N_3 - 1)$$

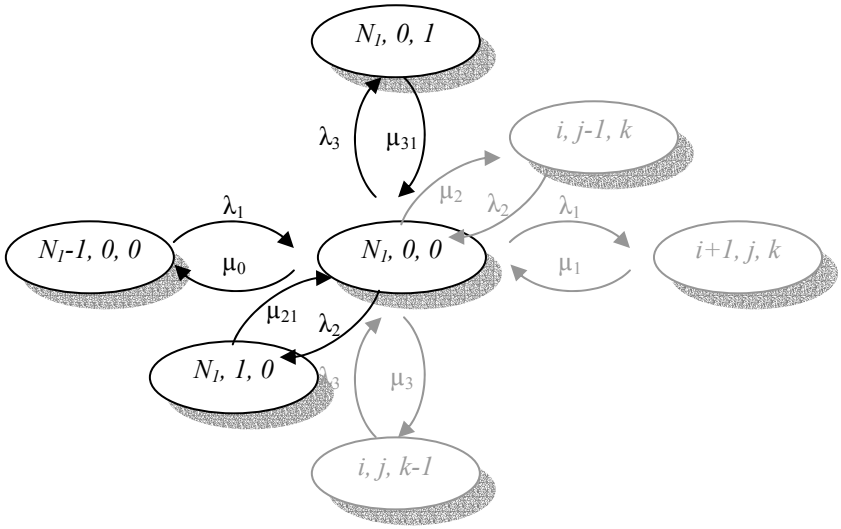


Figure 92: La chaîne incluse au point *i=N_1, j=0 et k=0*.

$$(\lambda_2 + \lambda_3 + \mu_0)P(N_1,0,0) = \lambda_1 P(N_1 - 1,0,0) + \mu_{21}P(N_1,1,0) + \mu_{31}P(N_1,0,1)$$

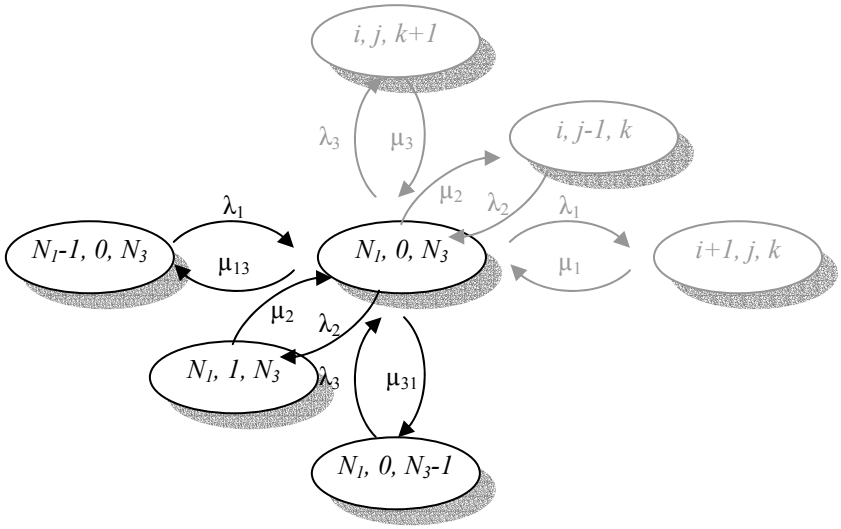


Figure 93: La chaîne incluse au point *i=N1, j=0 et k=N3*.

$$(\lambda_2 + \mu_{13} + \mu_{31})P(N_1,0,N_3) = \lambda_1 P(N_1 - 1,0,N_3) + \mu_2 P(N_1,1,N_3) + \lambda_3 P(N_1,0,N_3 - 1)$$

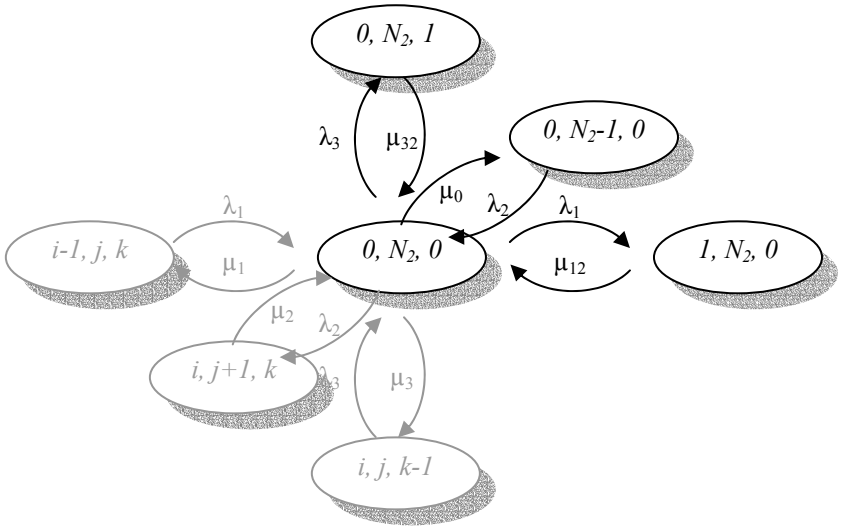


Figure 94: La chaîne incluse au point *i=0, j=N2 et k=0*.

$$(\lambda_1 + \lambda_3 + \mu_0)P(0,N_2,0) = \mu_{12} P(1,N_2,0) + \lambda_2 P(0,N_2 - 1,0) + \mu_{32} P(0,N_2,1)$$

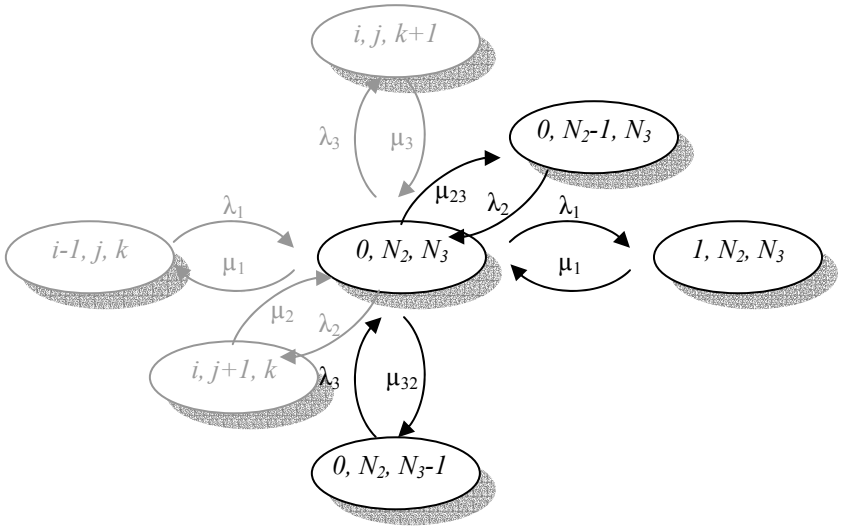


Figure 95: La chaîne incluse au point *i=0, j=N_2 et k=N_3*.

$$(\lambda_1 + \mu_{23} + \mu_{32})P(0, N_2, N_3) = \mu_1 P(1, N_2, N_3) + \lambda_2 P(0, N_2 - 1, N_3) + \lambda_3 P(0, N_2, N_3 - 1)$$

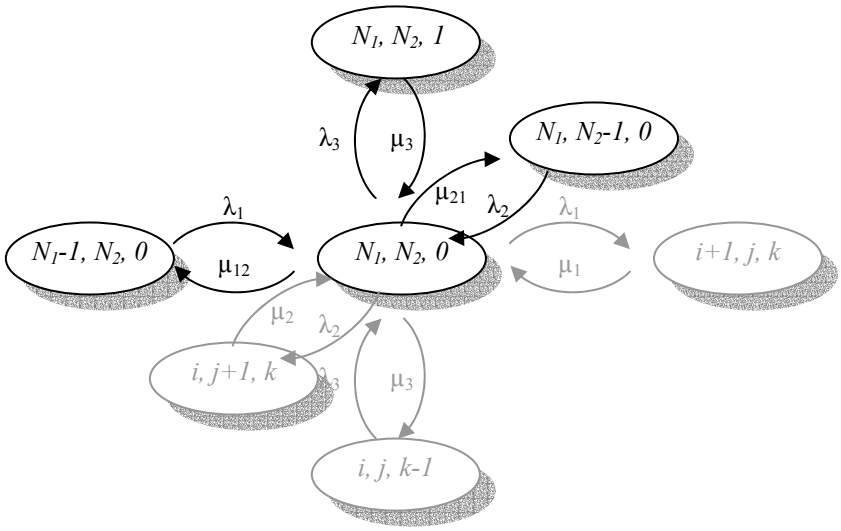


Figure 96: La chaîne incluse au point *i=N_1, j=N_2 et k=0*.

$$(\lambda_3 + \mu_{12} + \mu_{21})P(N_1, N_2, 0) = \lambda_1 P(N_1 - 1, N_2, 0) + \lambda_2 P(N_1, N_2 - 1, 0) + \mu_3 P(N_1, N_2, 1)$$

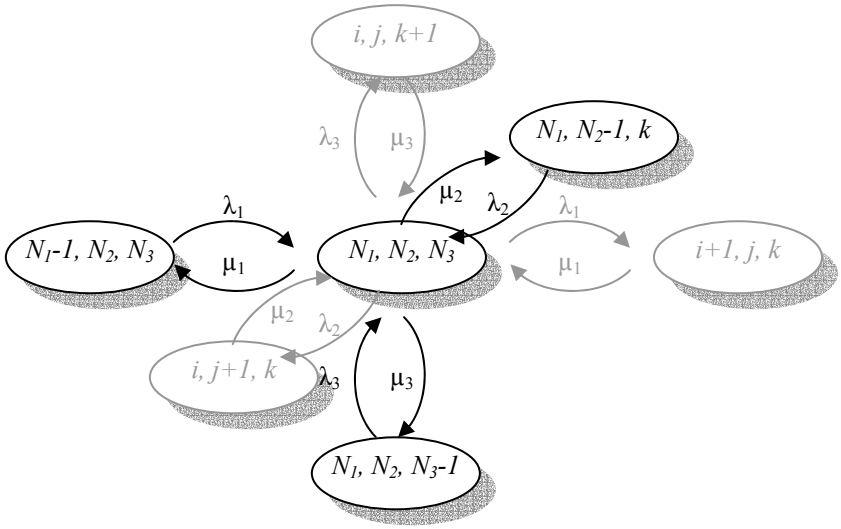


Figure 97: La chaîne incluse au point *i=N1, j=N2 et k=N3*.

$$(\mu_1+\mu_2+\mu_3)P(N_1,N_2,N_3)=\lambda_1 P(N_1-1,N_2,N_3)+\lambda_2 P(N_1,N_2-1,N_3)+\lambda_3 P(N_1,N_2,N_3-1)$$

2. L'évolution du délai

Cette partie regroupe les différentes courbes des délais de l'ordonnanceur WRR avec des poids attribués à chacune des trois classes non présentées dans le chapitre Chapitre 3.

Les premiers cinq tableaux de figures ont été obtenus avec un processus d'inter arrivée des paquets et d'une distribution de la taille des paquets qui suivent une loi exponentielle. Les cinq tableaux de figures qui les suivent représentent la variation du délai moyen avec un processus d'inter arrivée des paquets suivant un Poisson et une distribution de la taille des paquets qui suit la loi de Pareto décrite dans le paragraphe 3.2.2. .

Tableau 62: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Poisson.

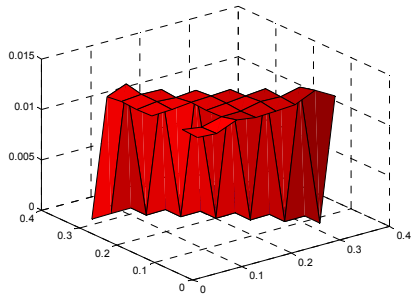
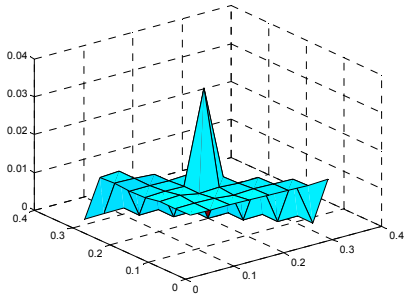
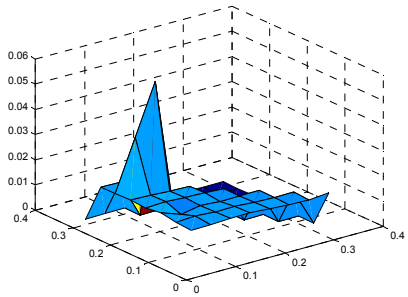
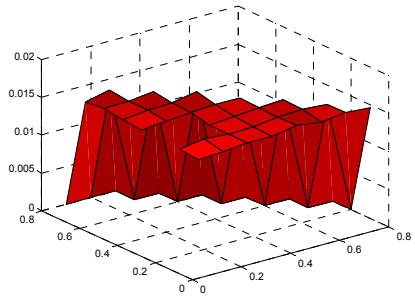
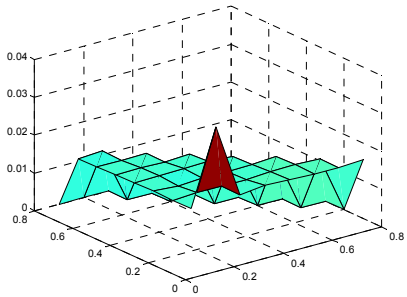
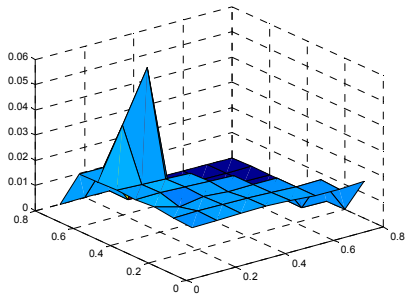
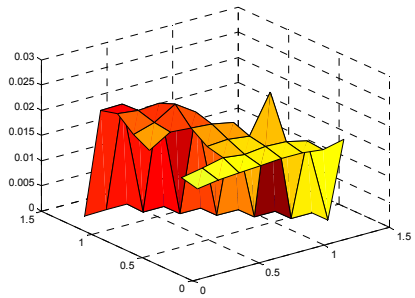
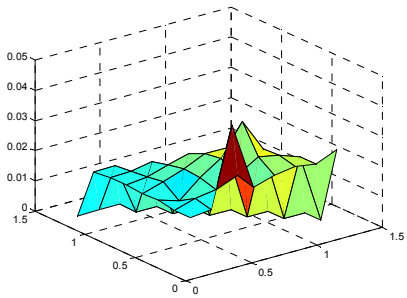
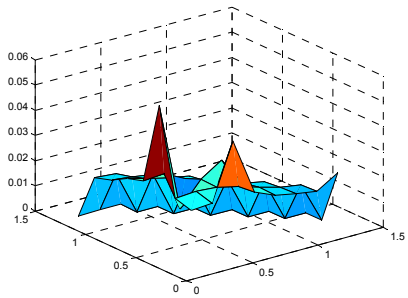
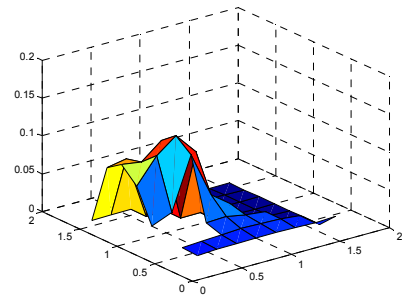
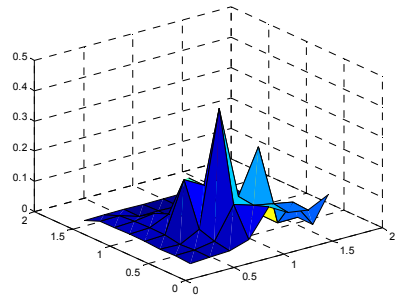
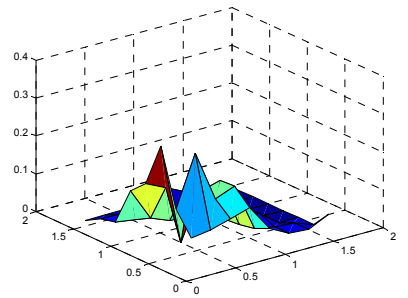
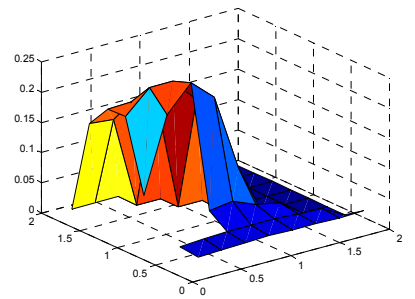
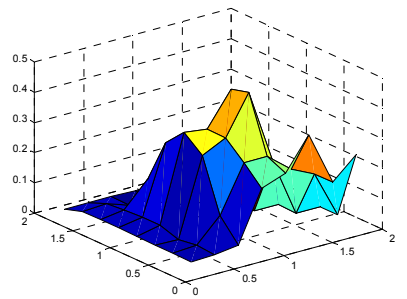
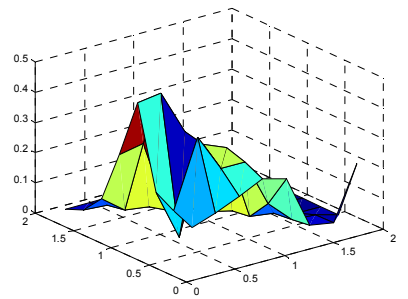
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 63: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Poisson.

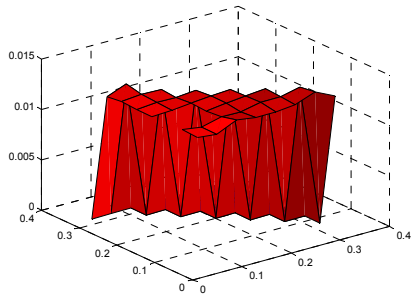
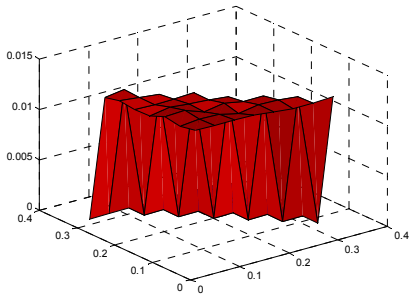
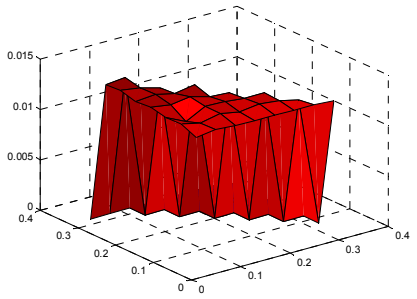
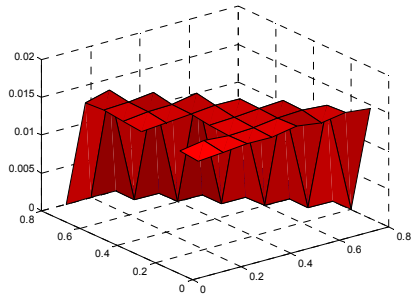
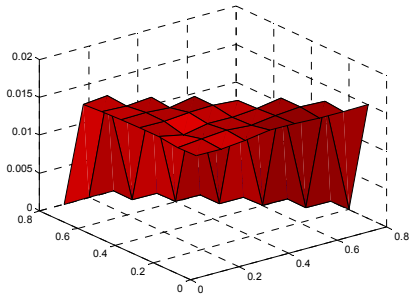
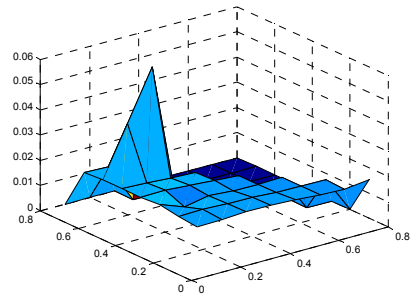
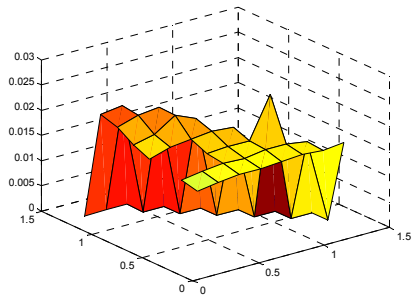
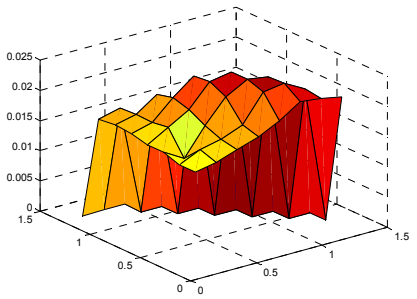
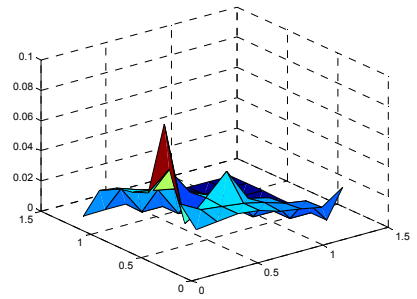
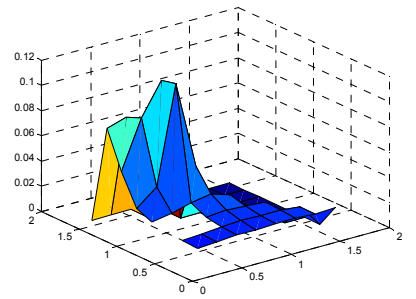
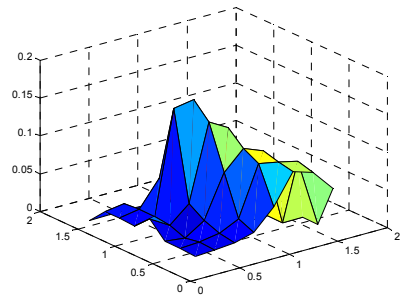
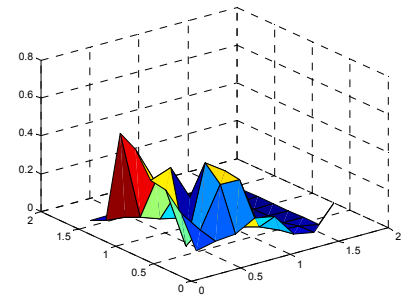
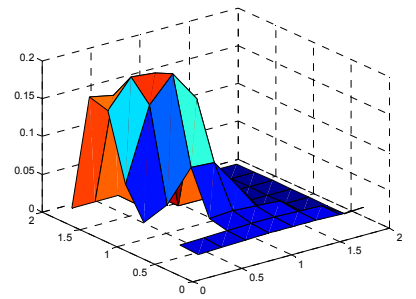
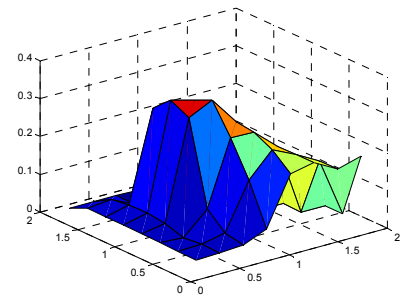
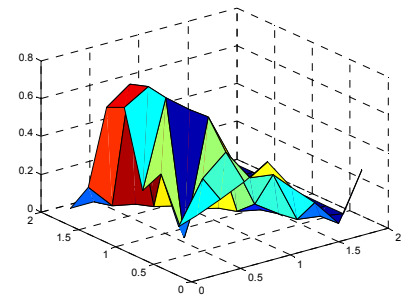
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 64: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Poisson.

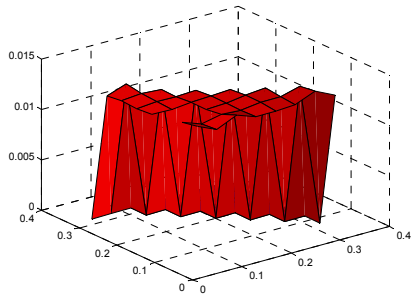
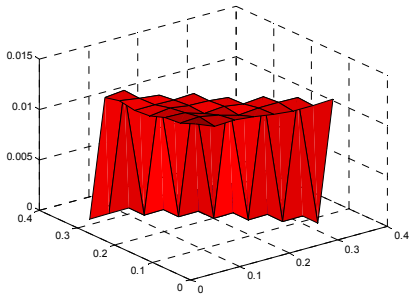
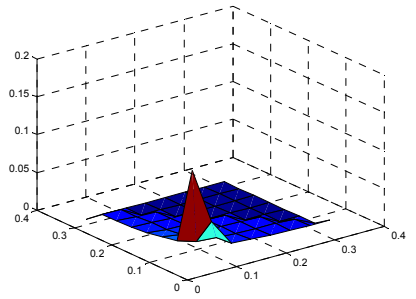
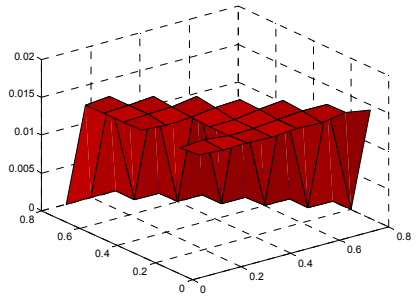
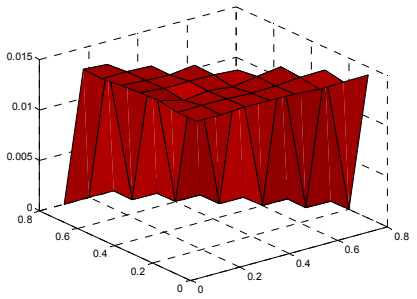
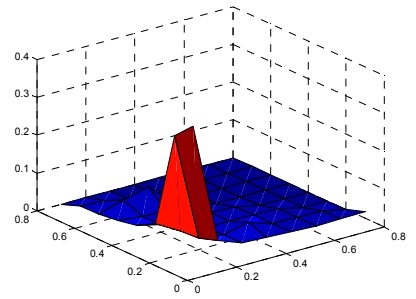
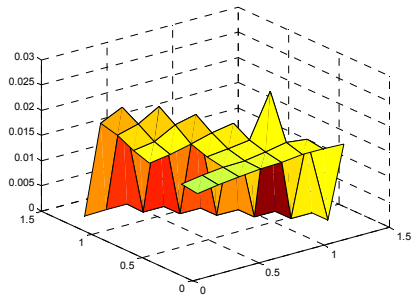
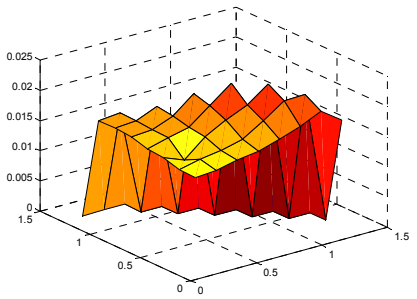
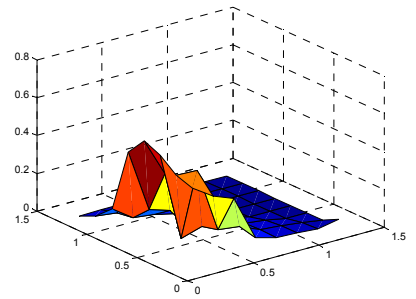
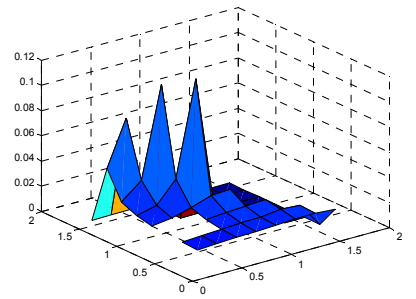
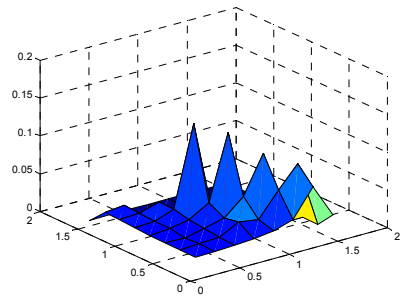
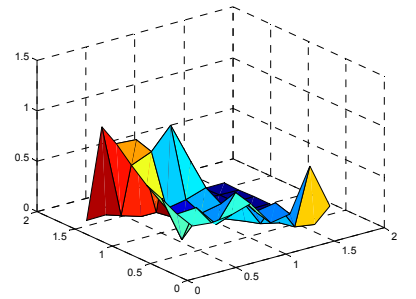
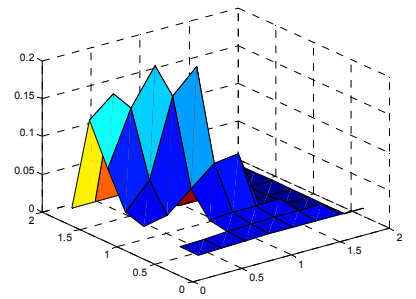
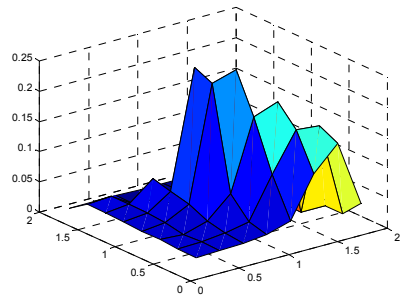
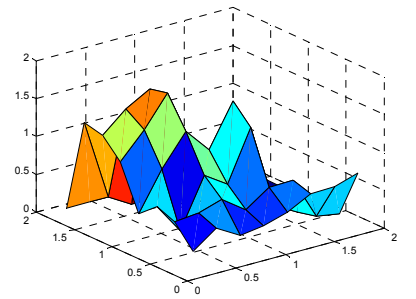
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 65: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Poisson.

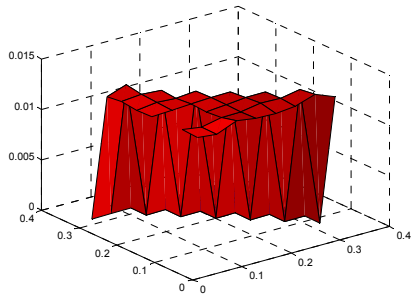
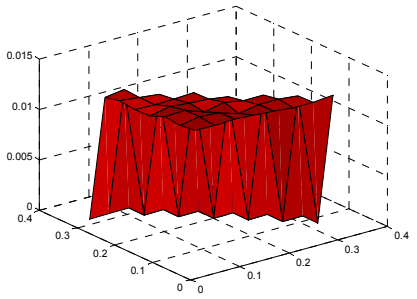
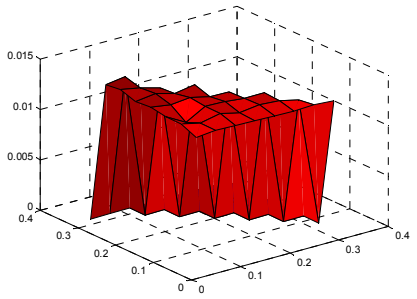
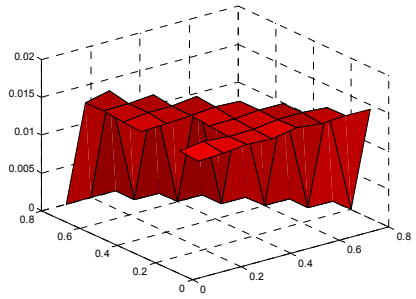
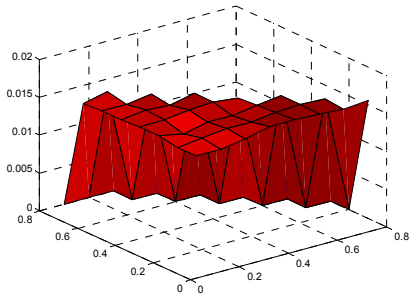
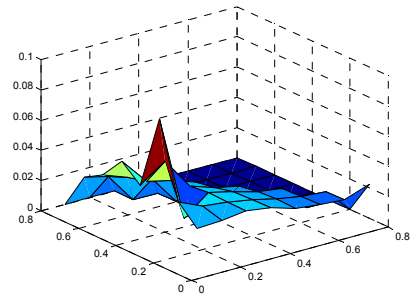
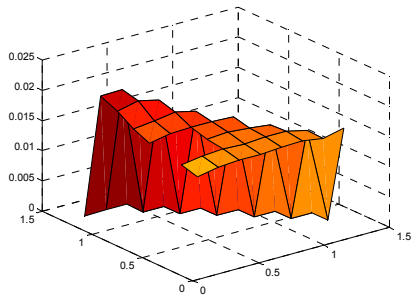
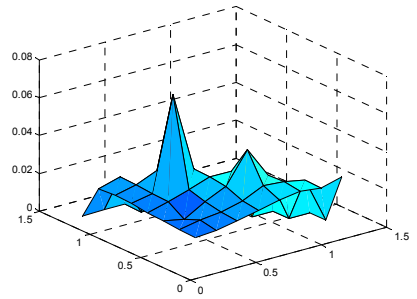
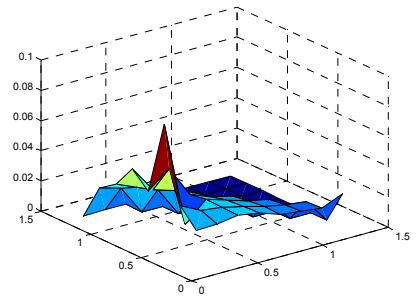
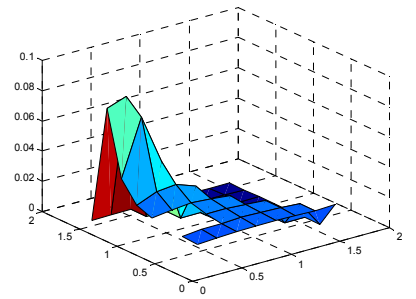
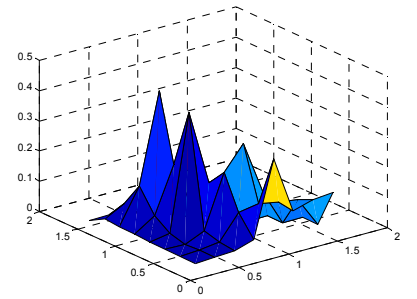
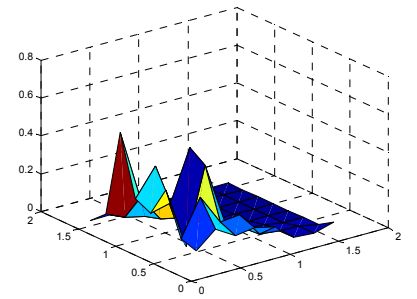
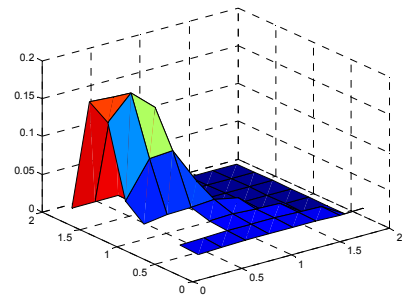
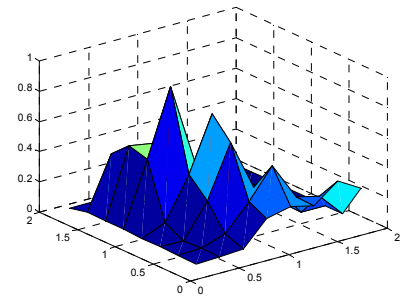
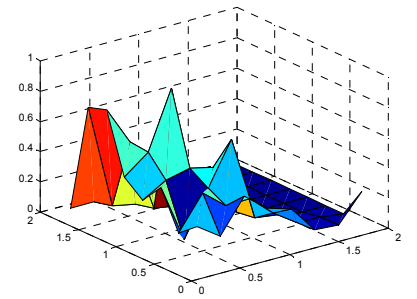
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 66: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Poisson.

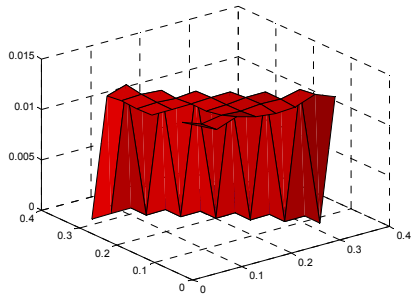
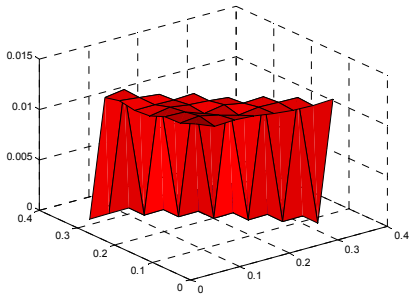
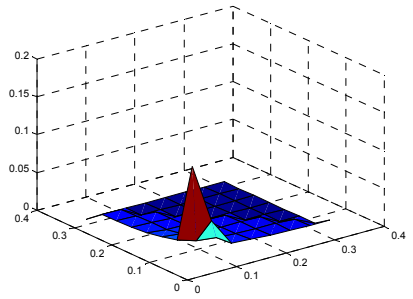
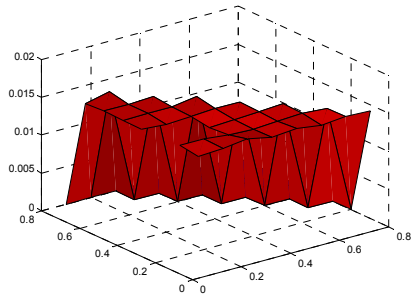
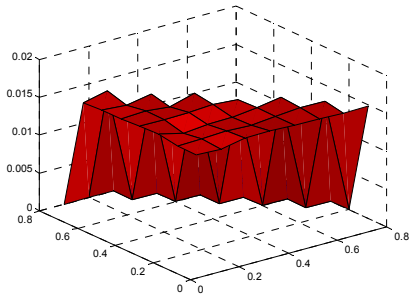
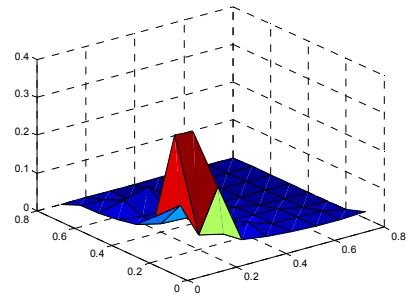
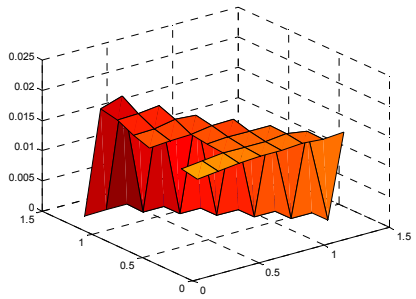
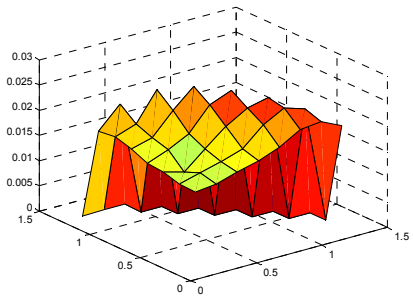
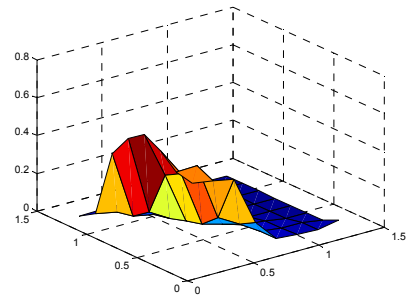
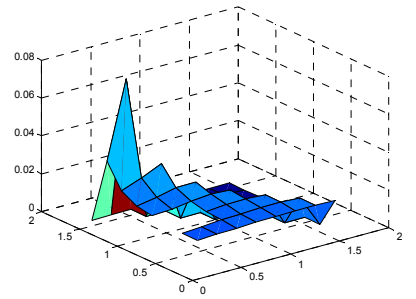
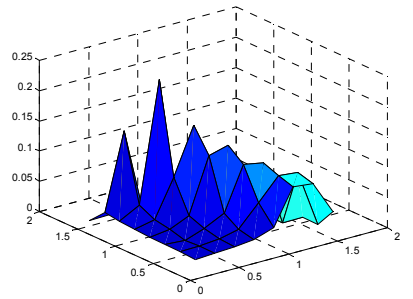
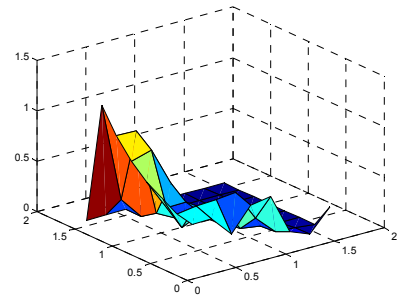
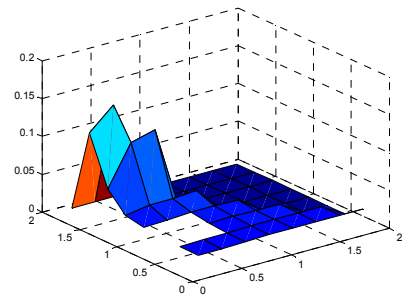
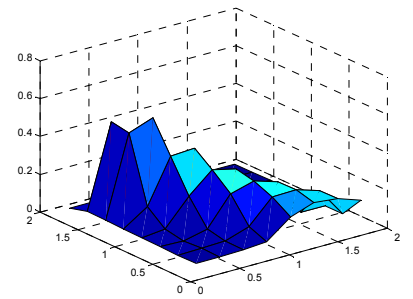
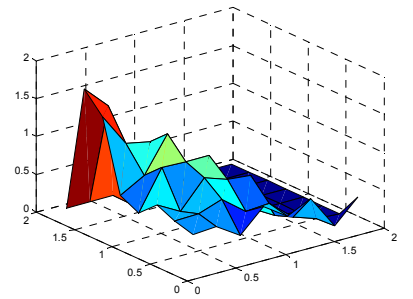
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 67: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Pareto.

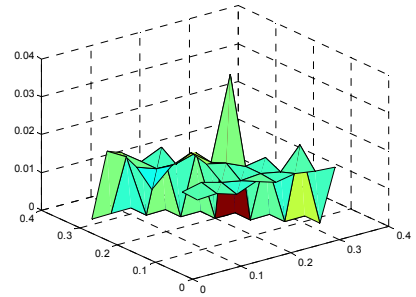
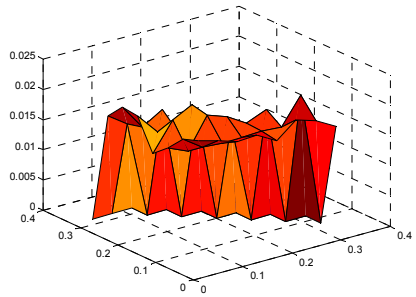
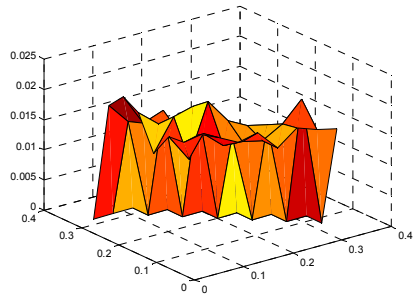
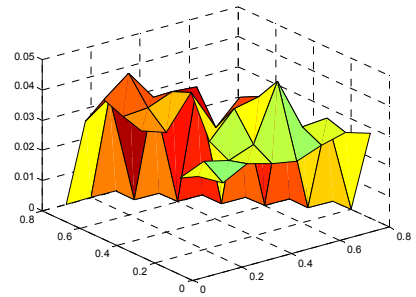
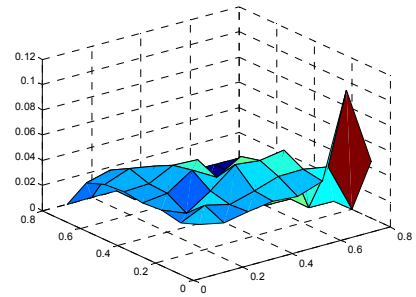
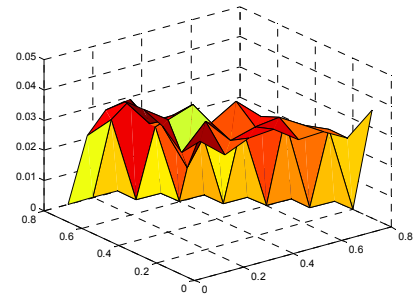
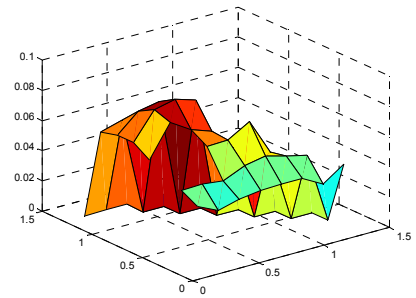
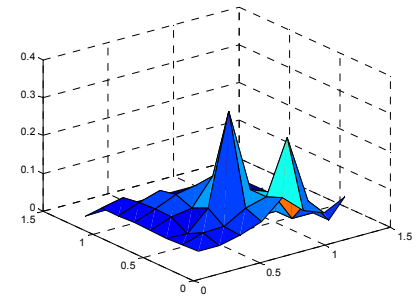
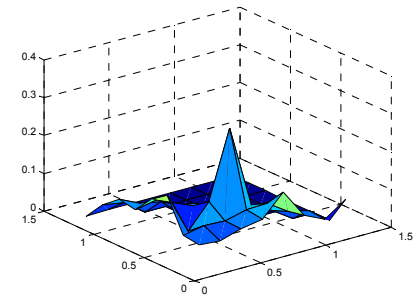
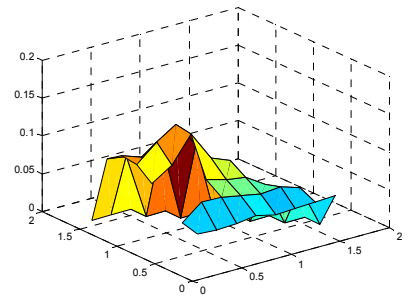
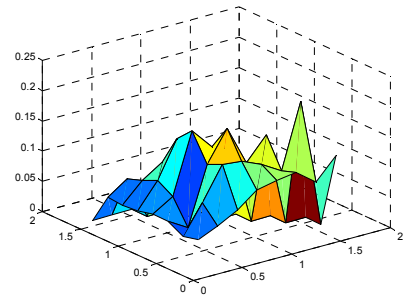
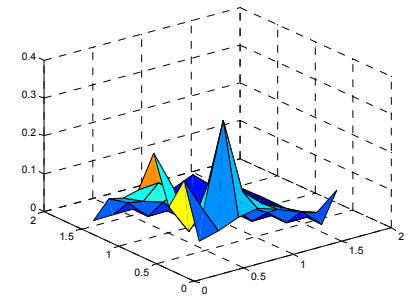
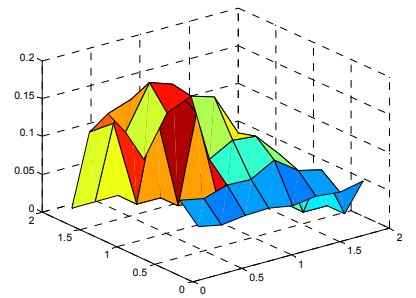
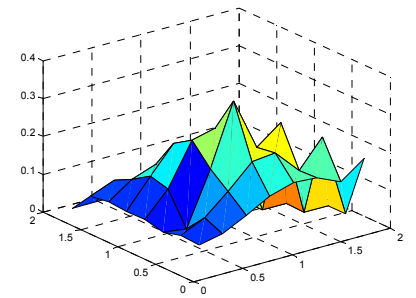
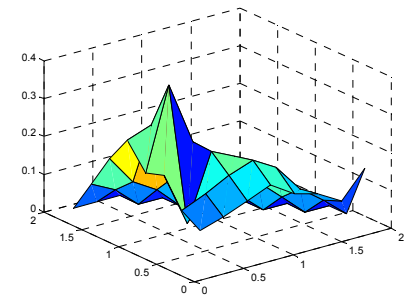
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 68: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Pareto.

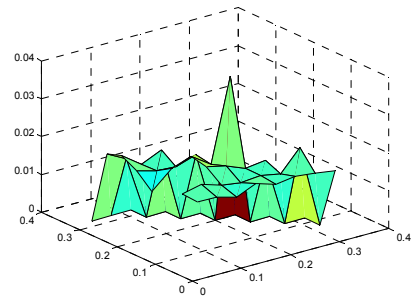
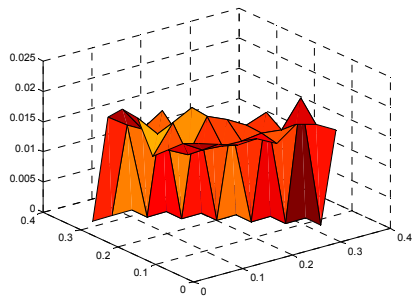
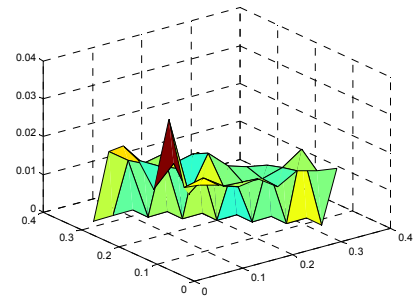
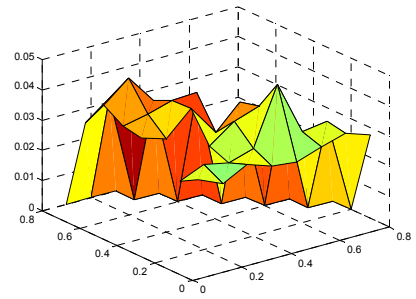
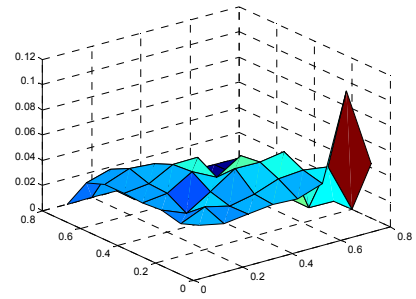
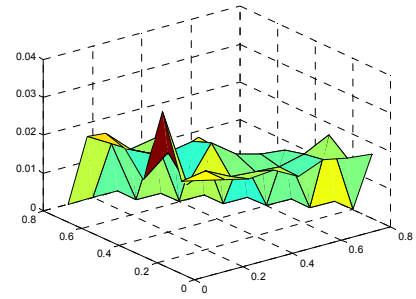
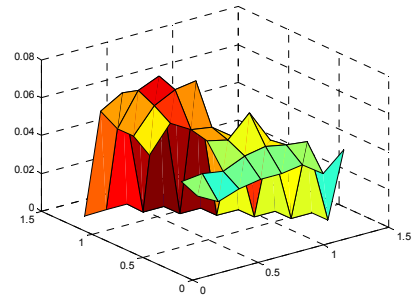
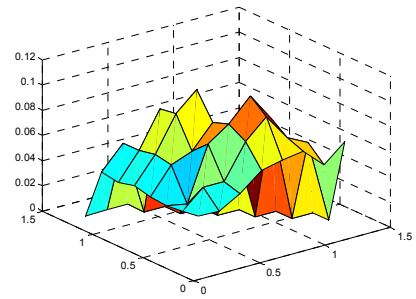
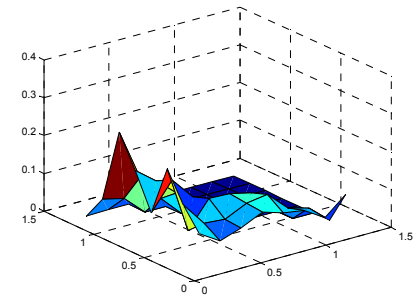
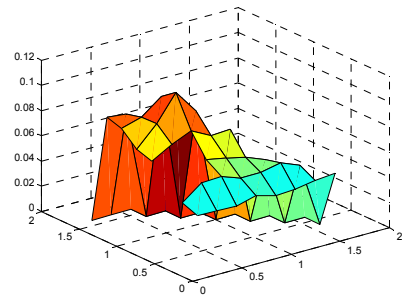
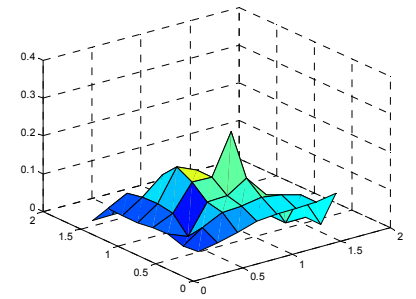
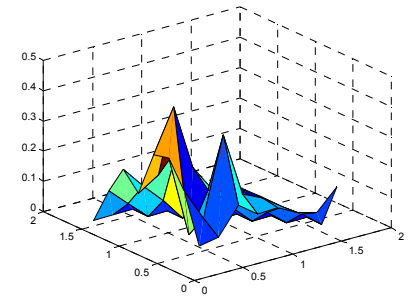
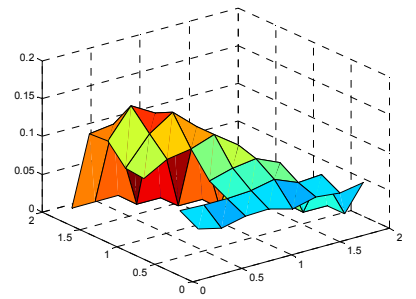
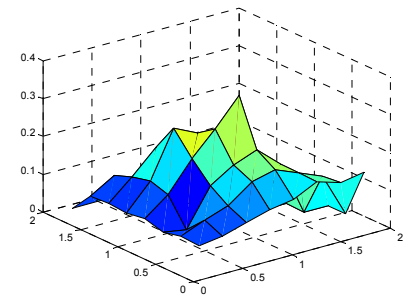
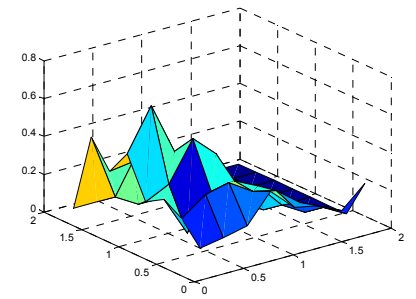
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 69: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Pareto.

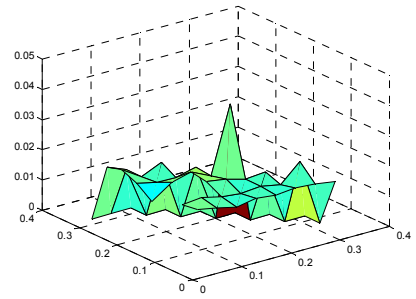
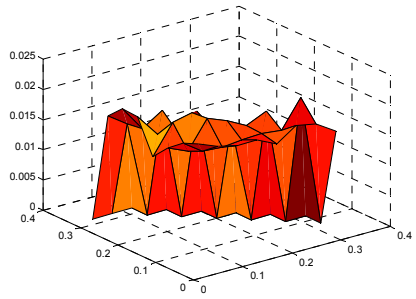
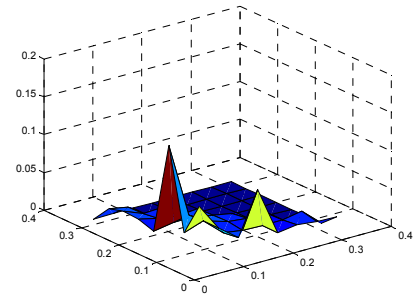
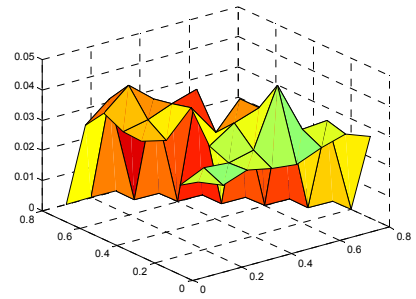
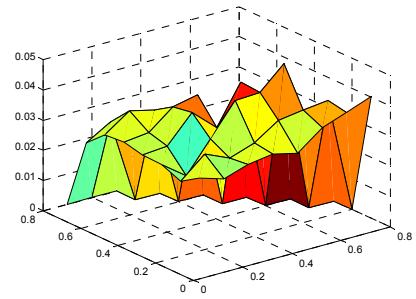
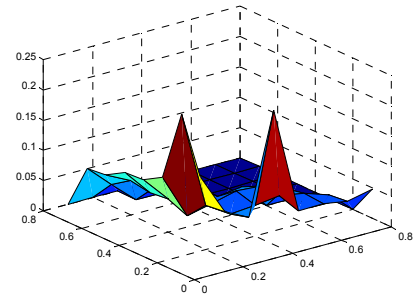
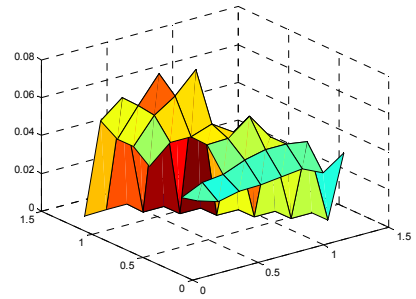
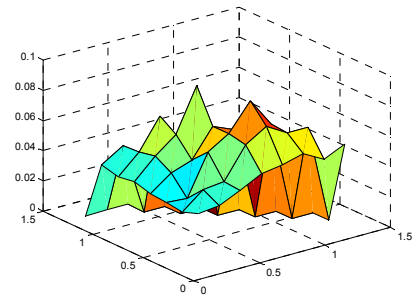
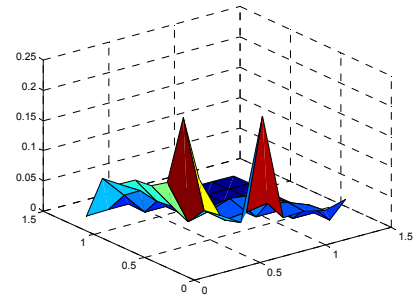
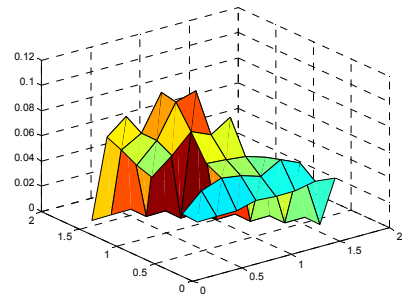
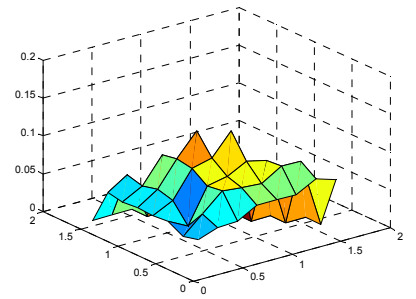
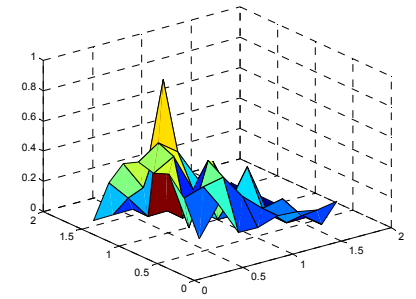
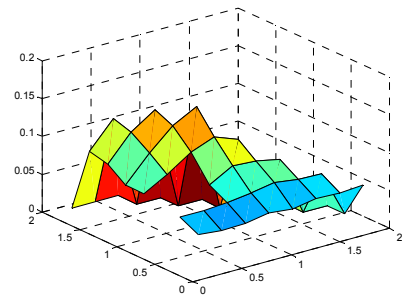
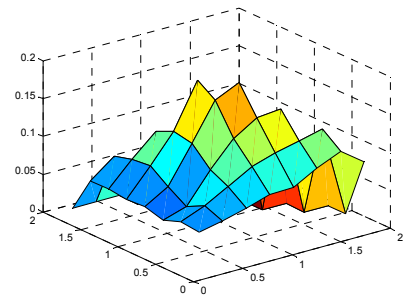
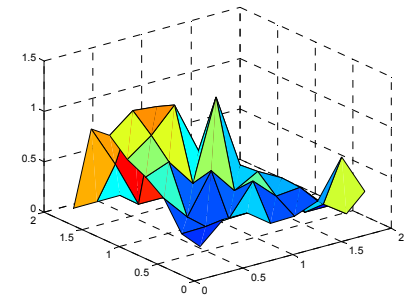
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 70: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Pareto.

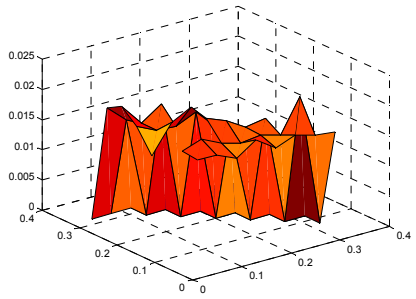
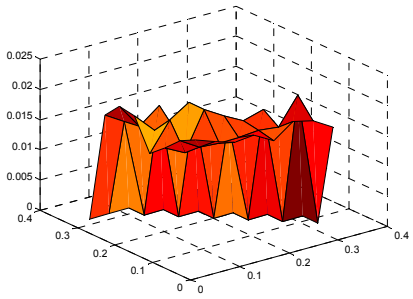
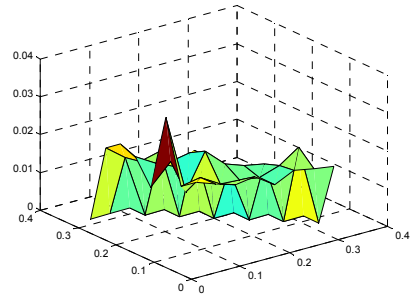
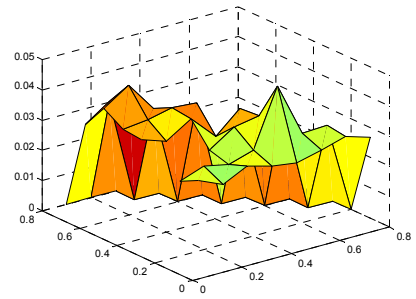
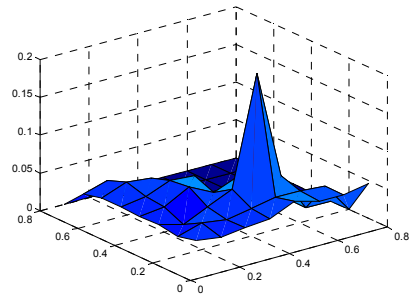
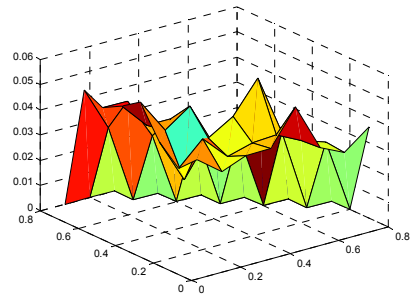
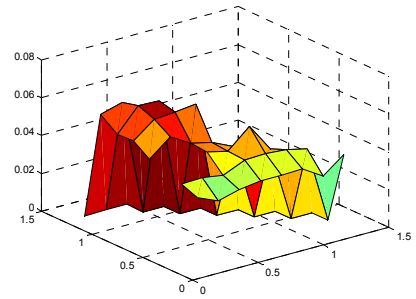
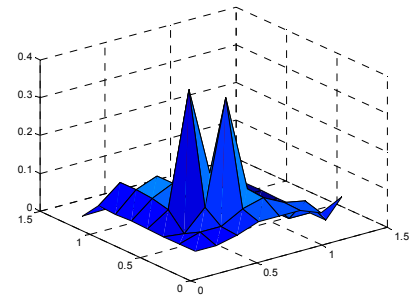
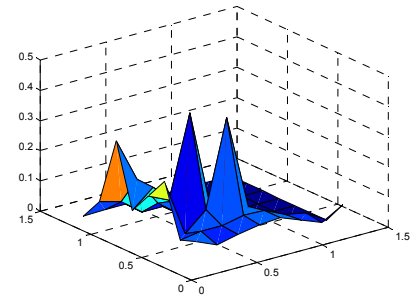
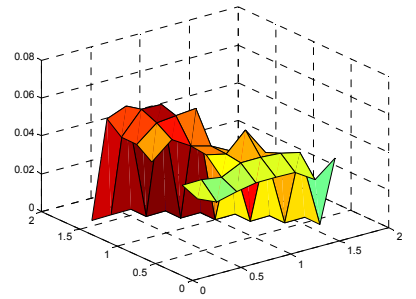
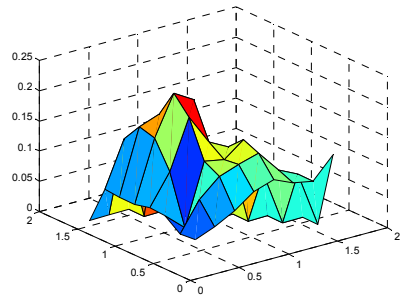
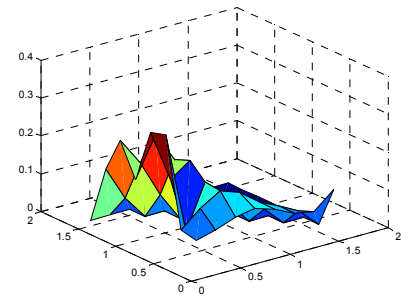
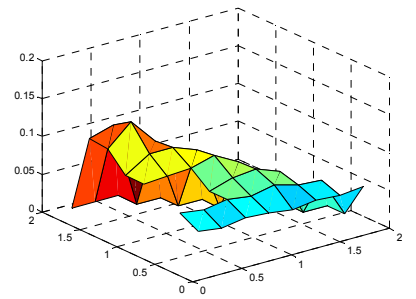
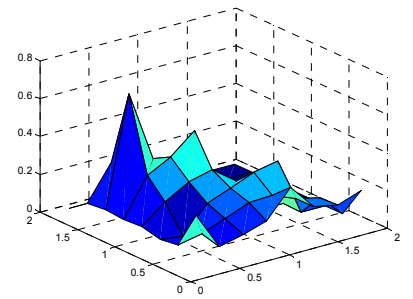
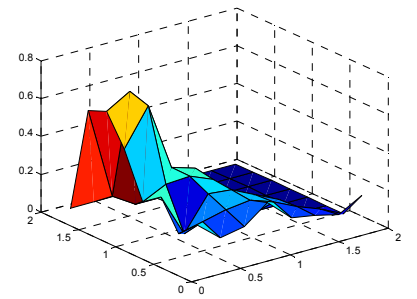
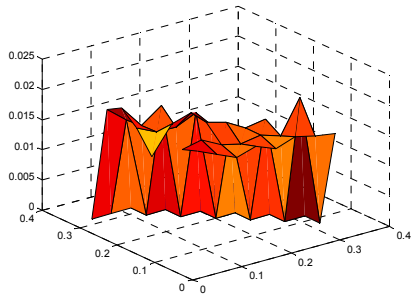
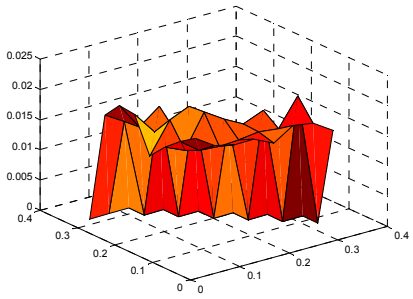
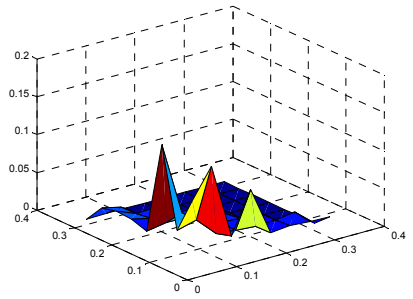
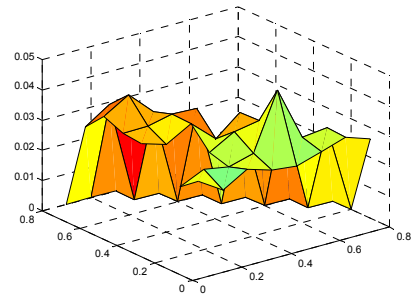
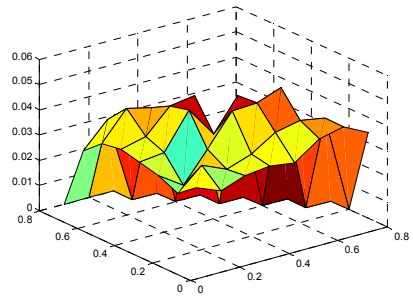
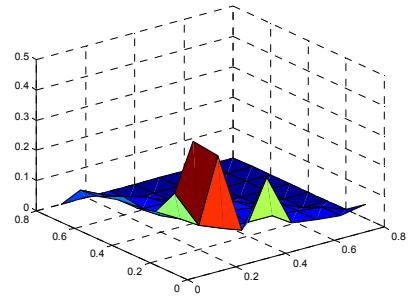
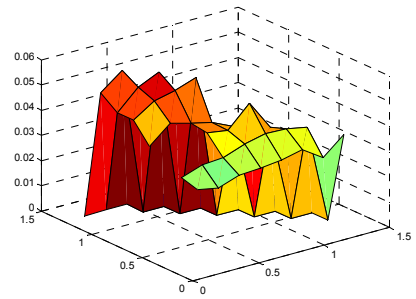
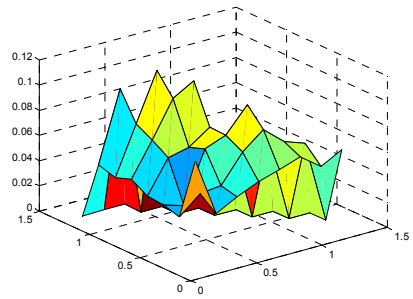
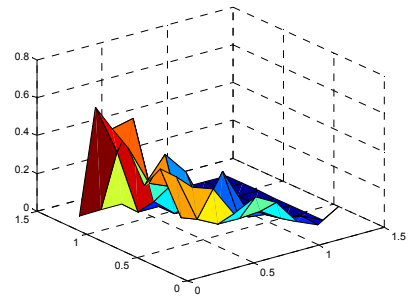
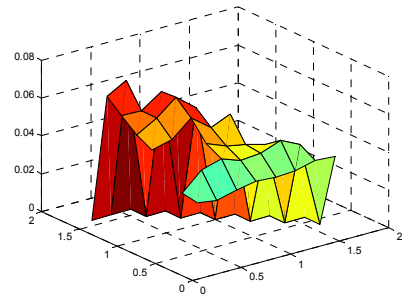
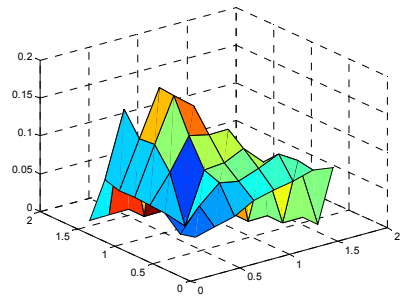
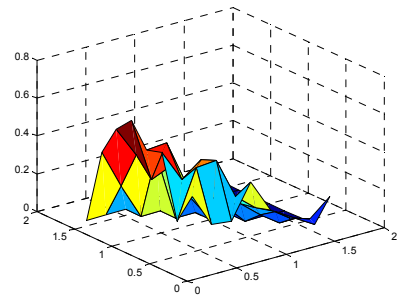
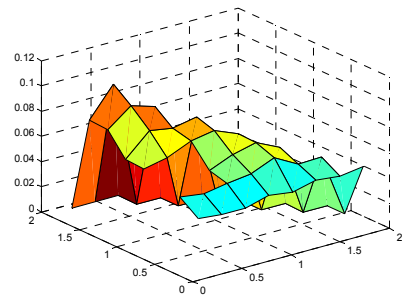
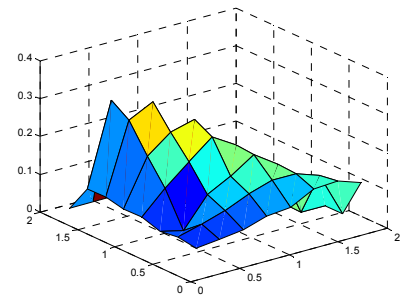
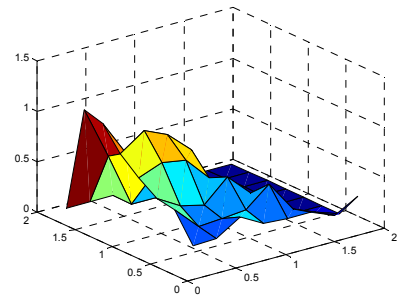
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 71: La variation du délai moyen en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Pareto.

Classe 1	Classe 2	Classe 3
		
		
		
		
		

3. L'évolution du taux de perte

Comme dans le paragraphe précédent, les tableaux de figures suivants sont complémentaires au paragraphe 3.2. Ci-dessous, nous présentons les variations du taux de perte en fonction de la charge et de la pondération utilisée. Les premiers cinq tableaux de figures reflètent les performances d'une file M/M/1. Les cinq tableaux d'après sont obtenus avec une file M/Pareto/1.

Tableau 72: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Poisson.

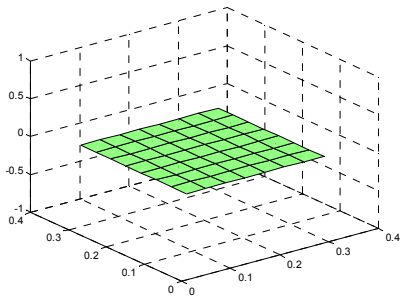
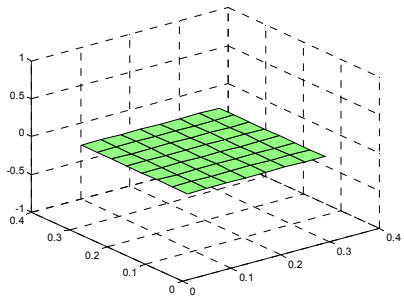
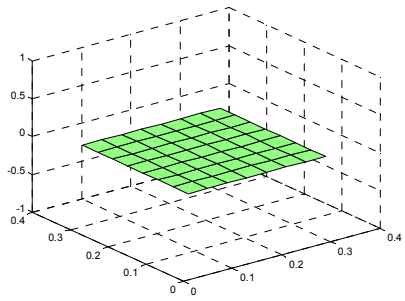
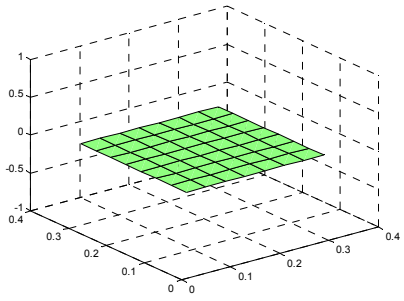
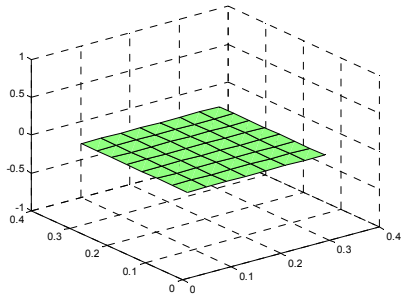
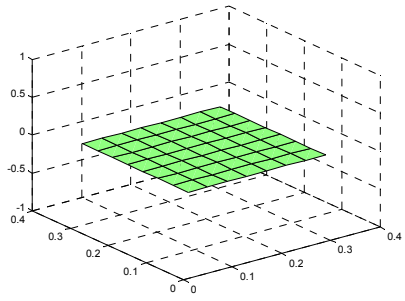
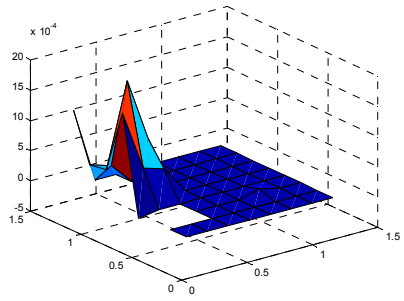
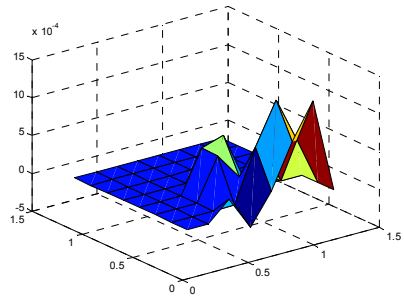
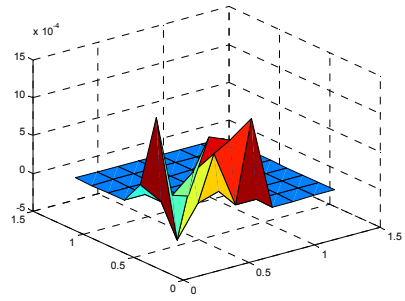
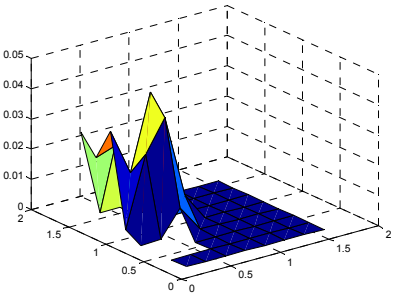
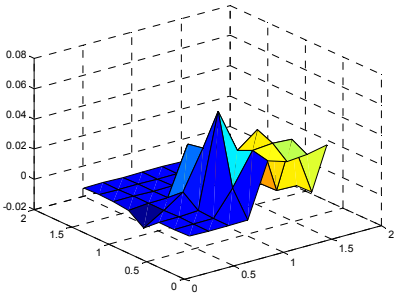
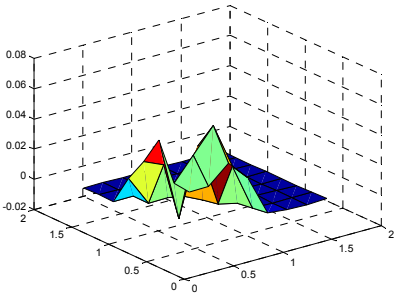
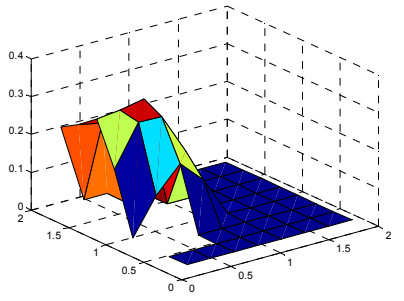
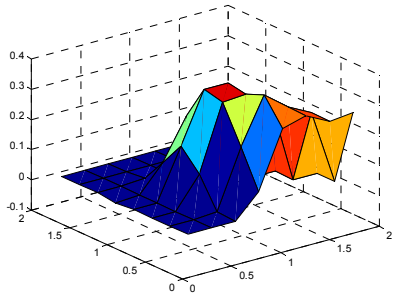
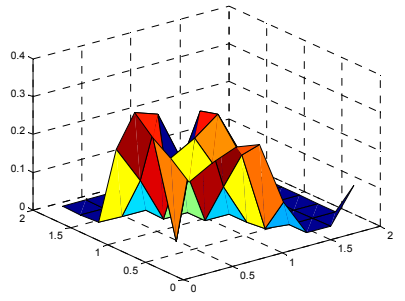
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 73: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2 avec une distribution de la taille des paquets suivant la loi de Poisson.

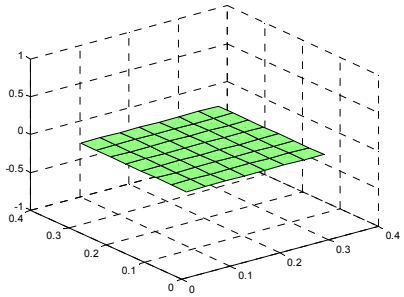
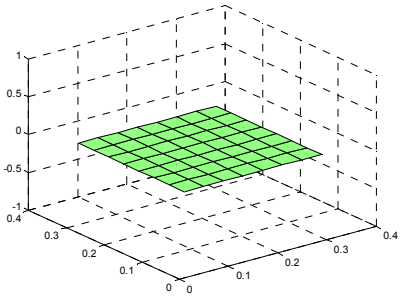
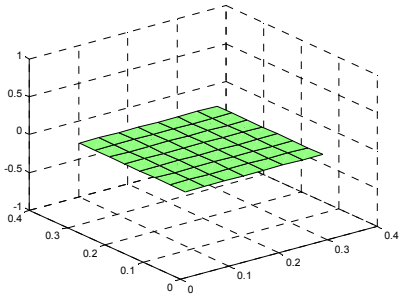
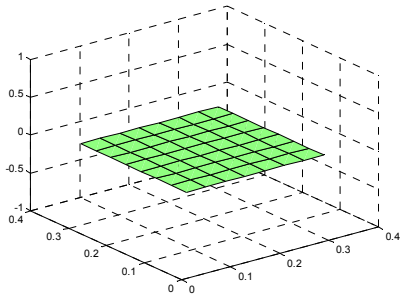
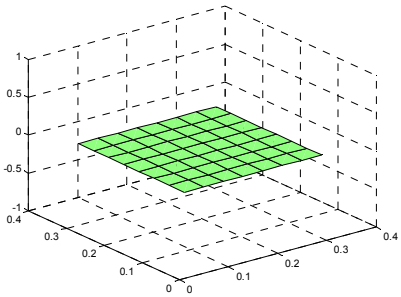
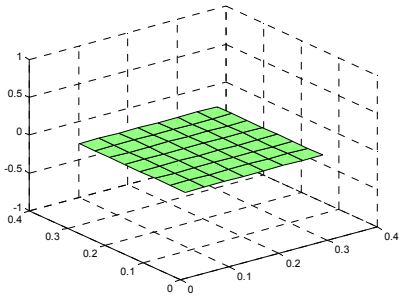
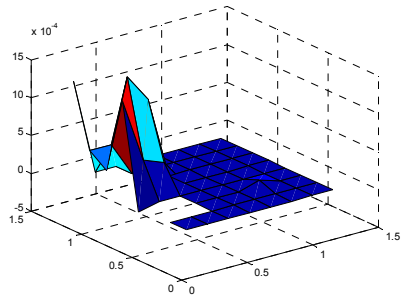
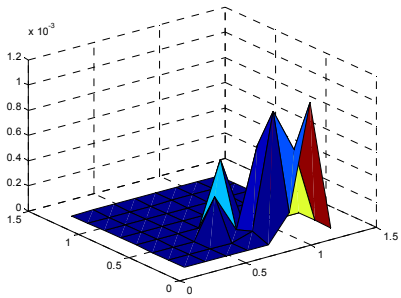
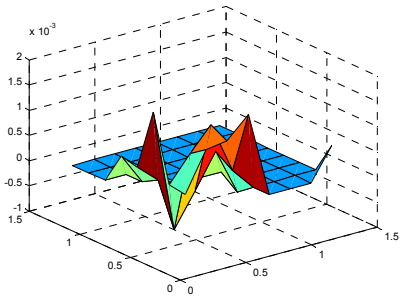
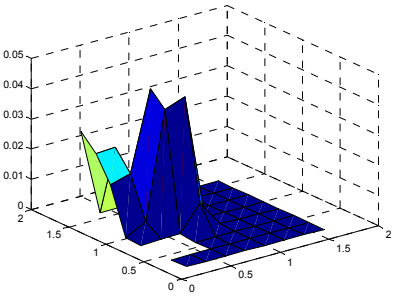
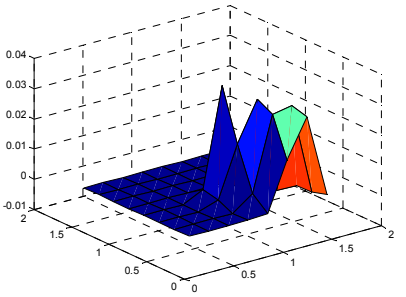
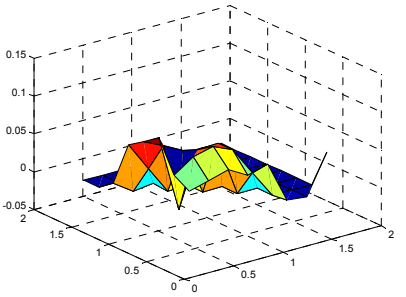
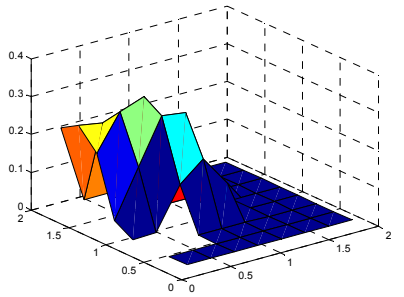
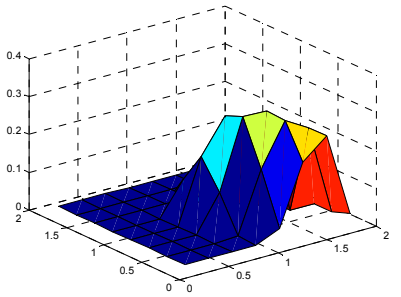
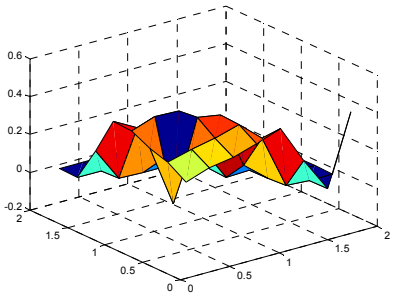
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 74: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Poisson.

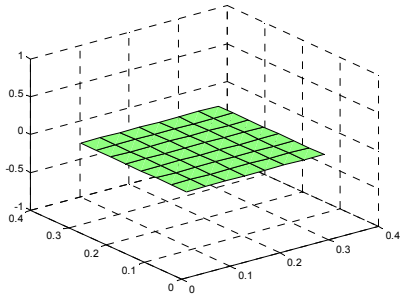
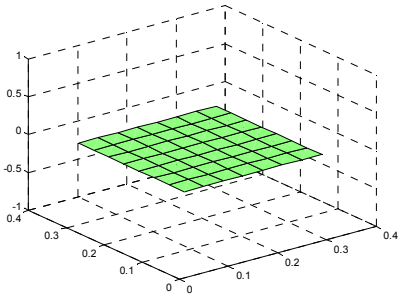
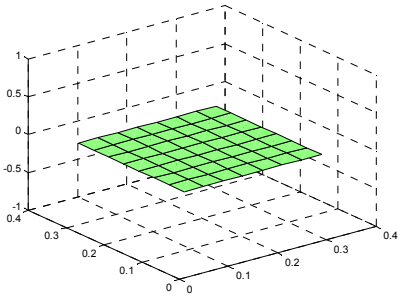
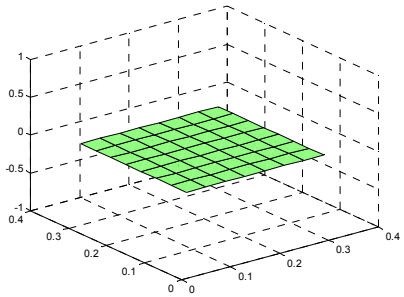
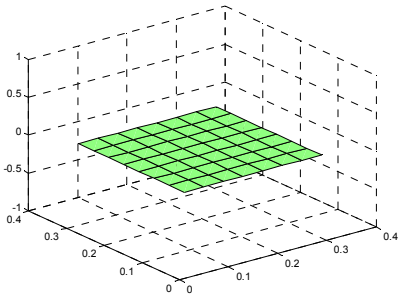
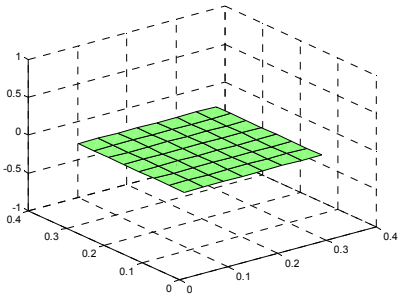
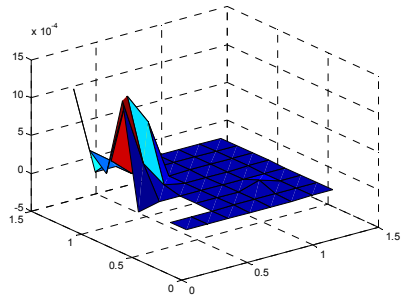
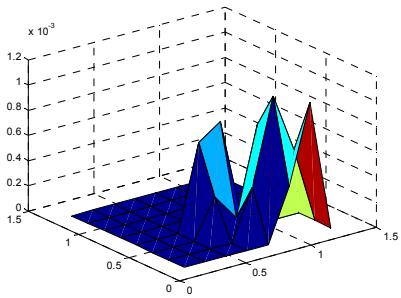
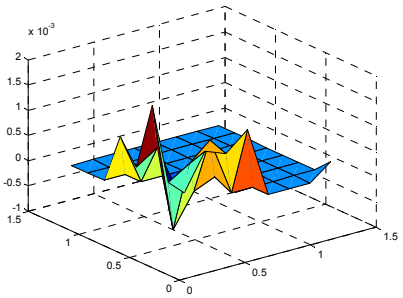
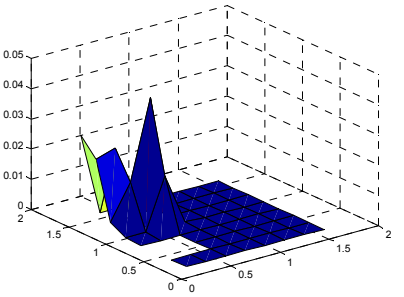
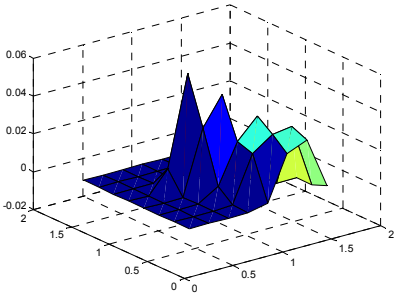
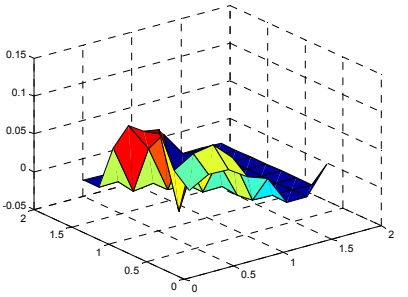
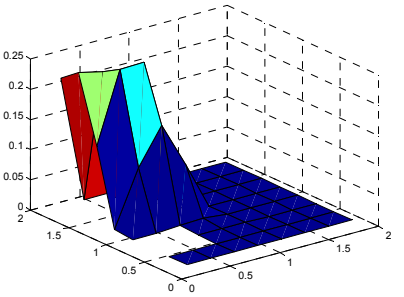
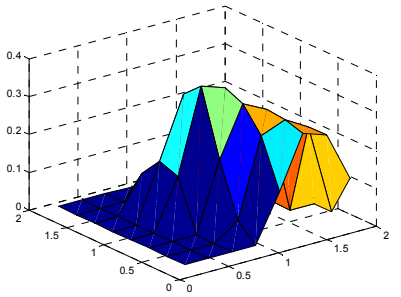
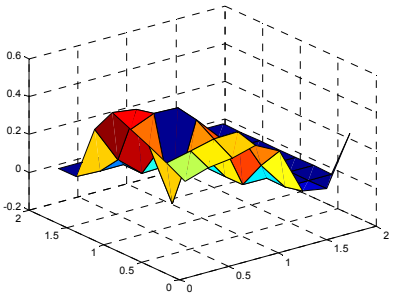
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 75: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Poisson.

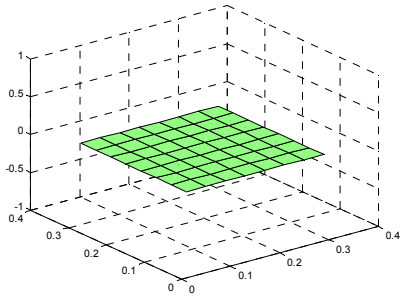
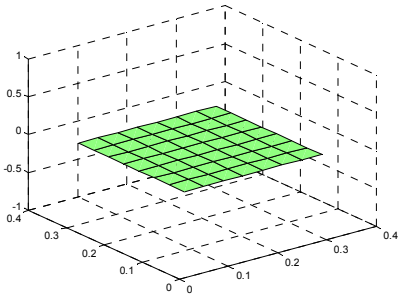
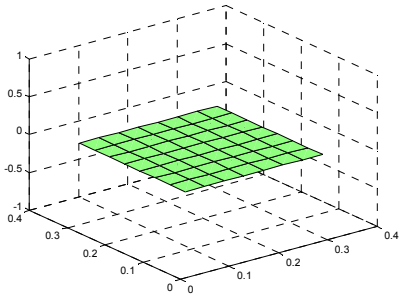
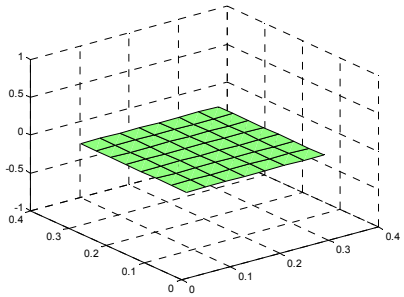
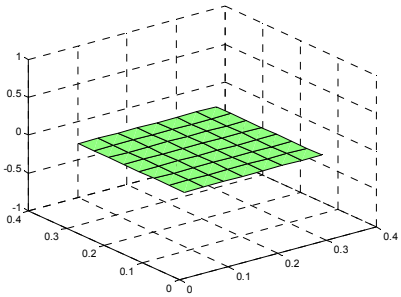
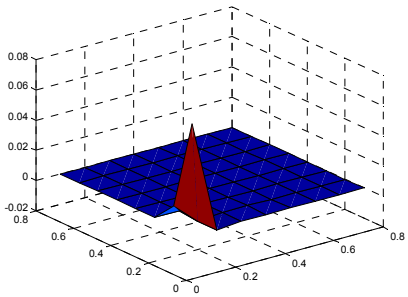
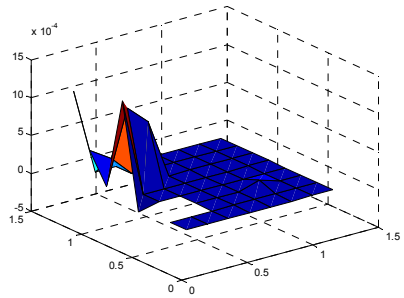
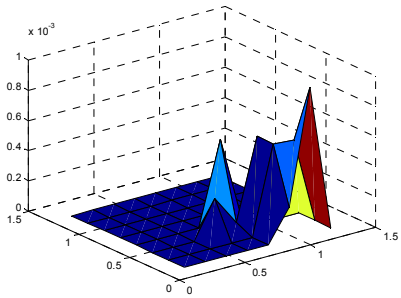
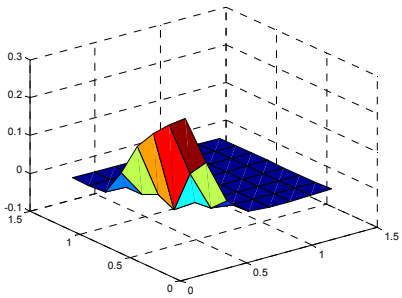
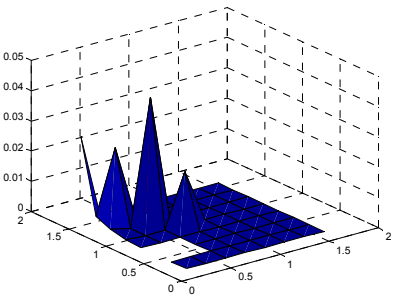
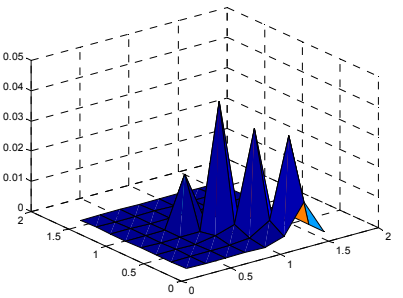
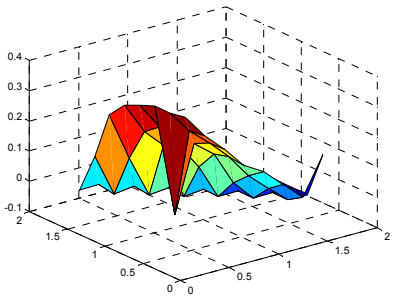
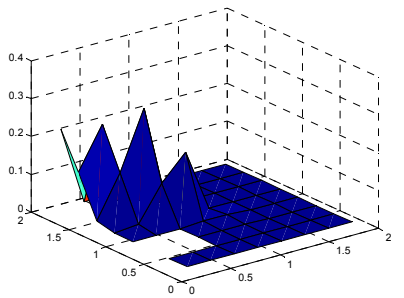
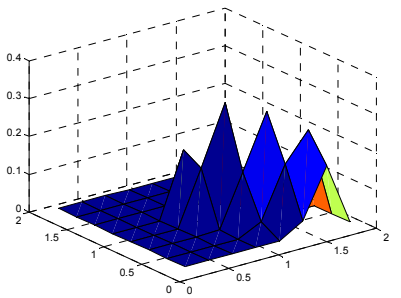
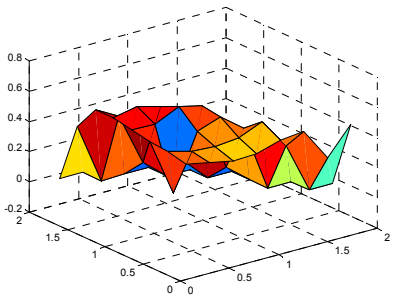
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 76: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Poisson.

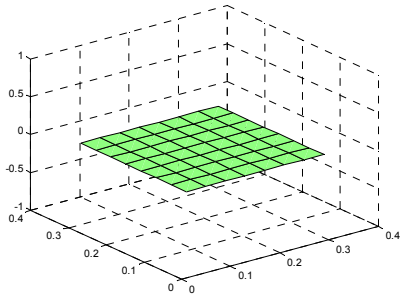
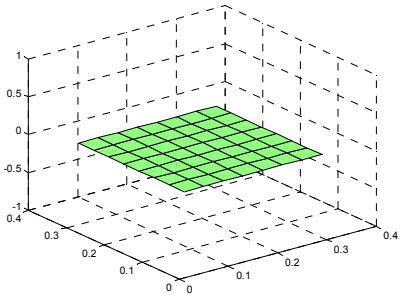
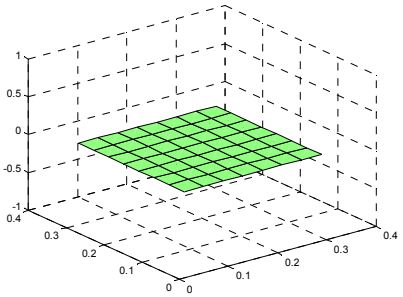
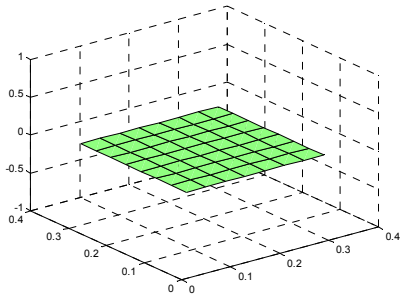
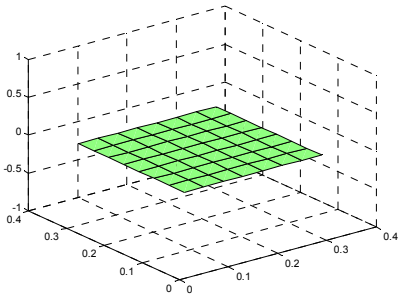
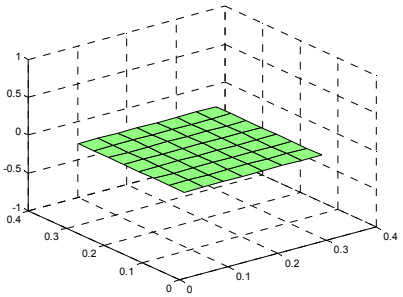
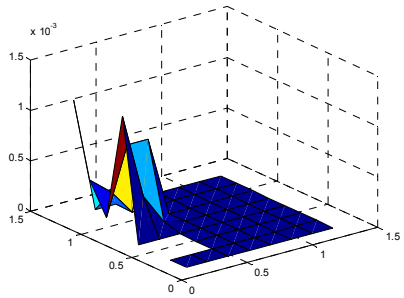
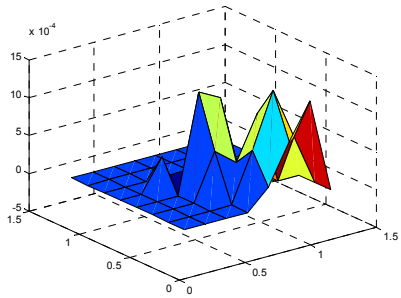
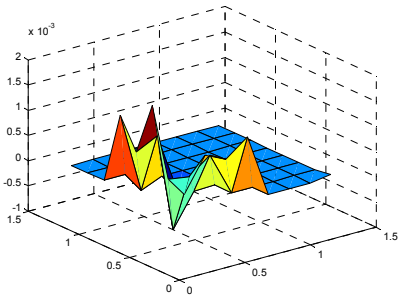
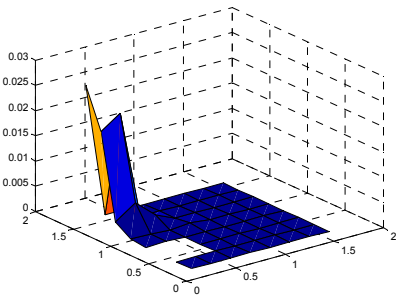
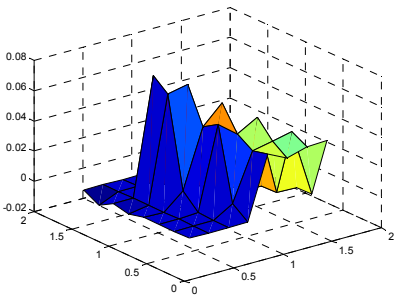
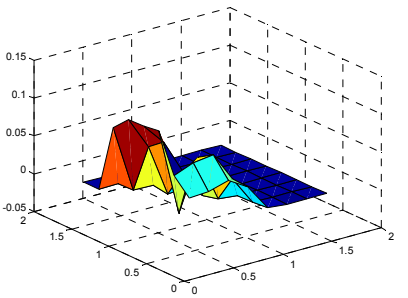
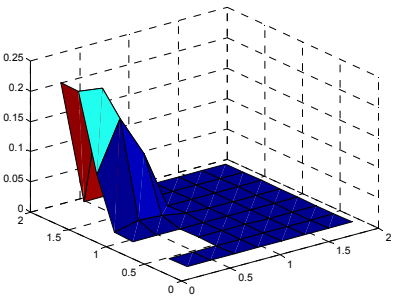
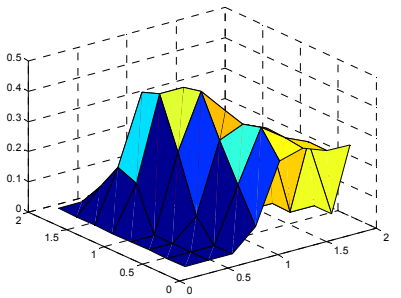
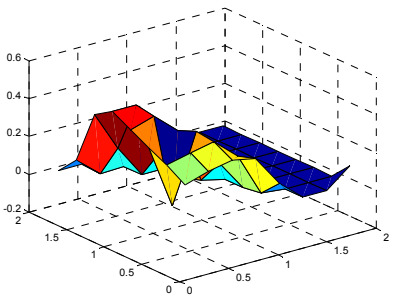
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 77: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1 avec une distribution de la taille des paquets suivant la loi de Poisson.

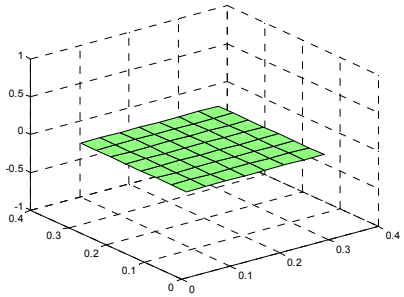
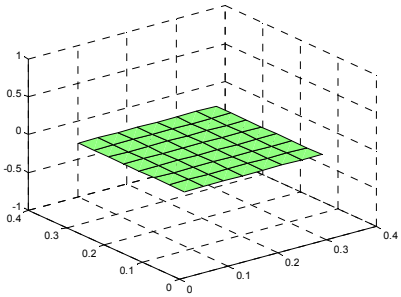
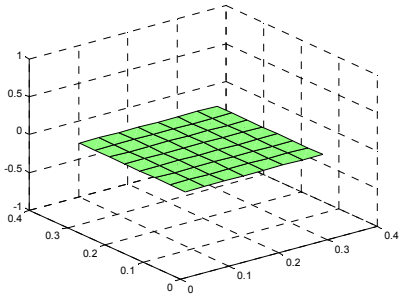
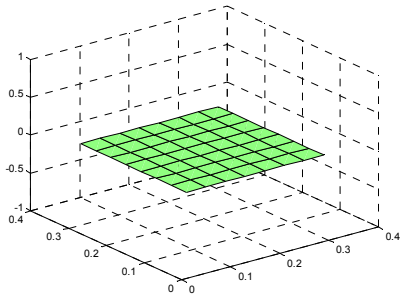
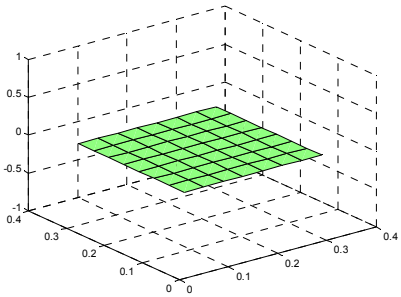
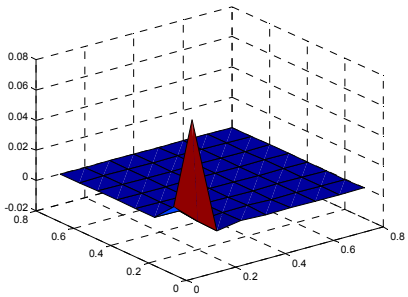
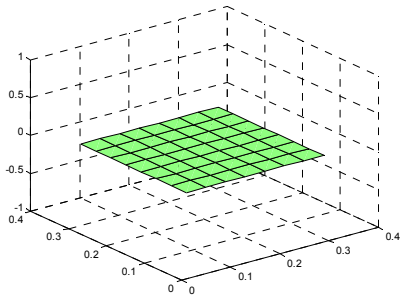
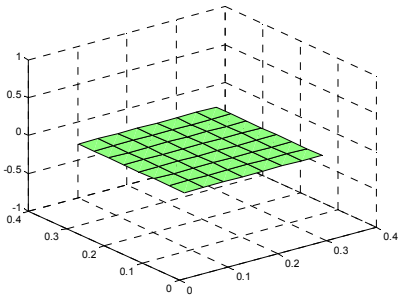
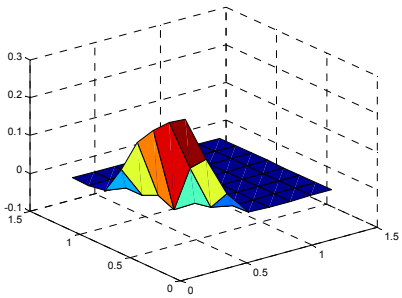
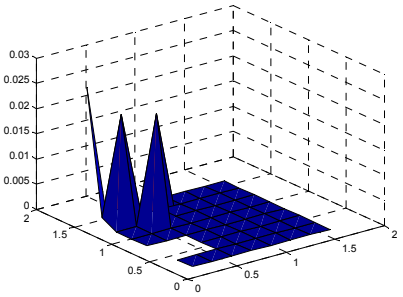
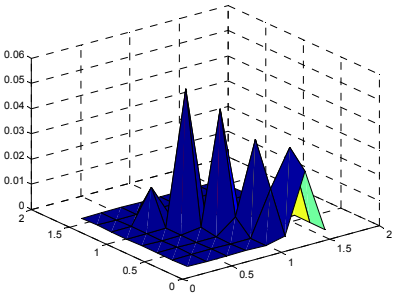
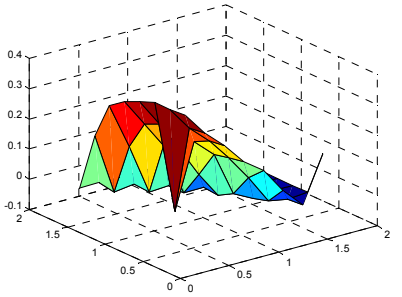
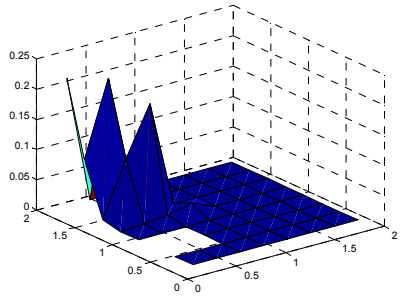
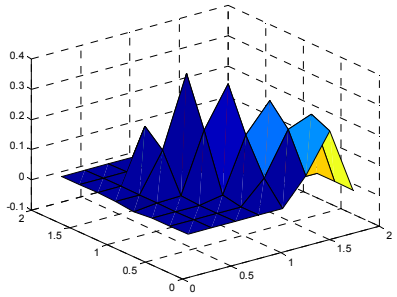
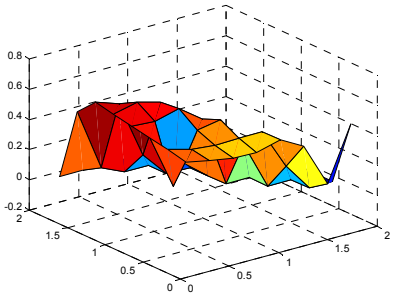
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 78: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Poisson.

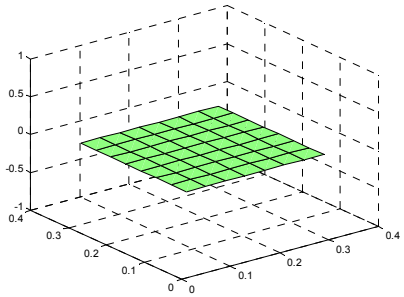
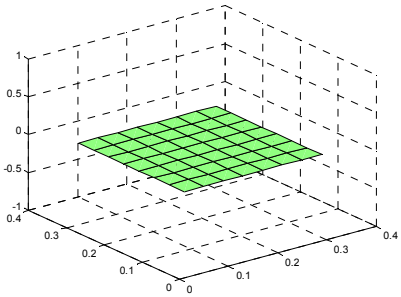
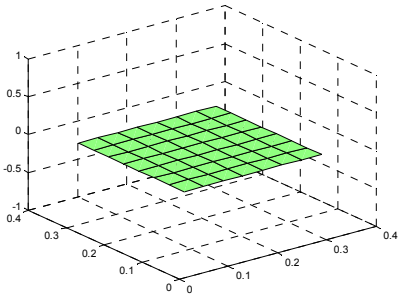
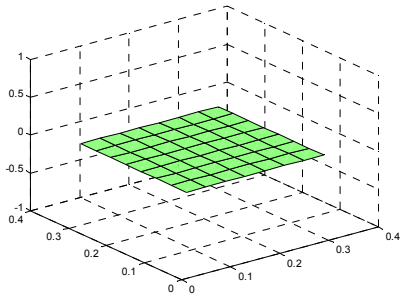
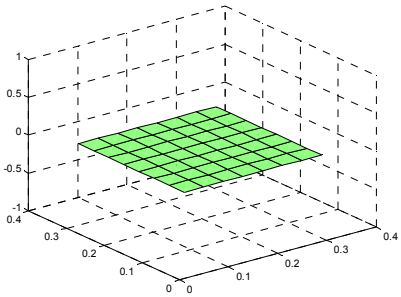
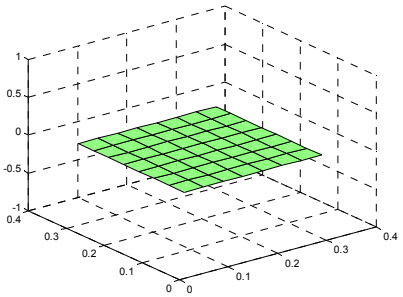
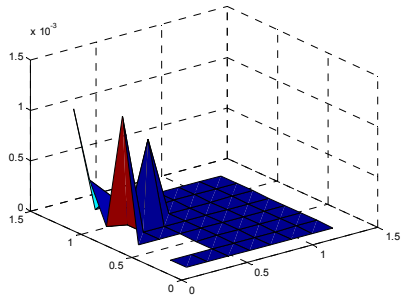
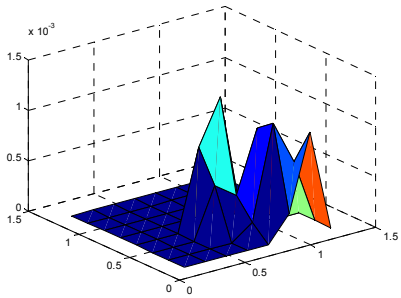
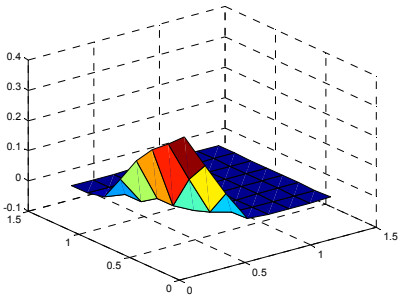
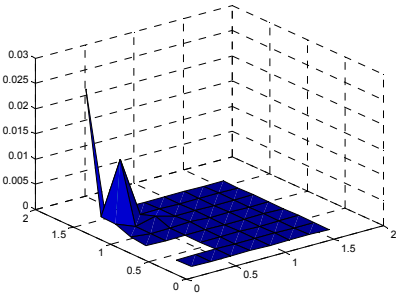
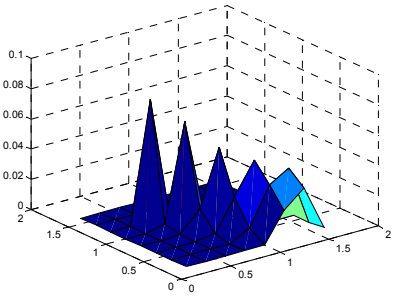
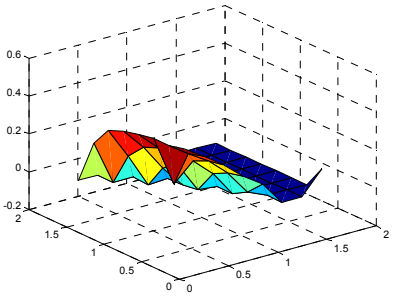
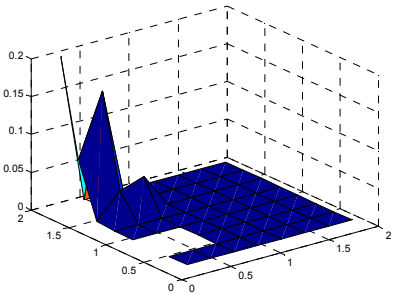
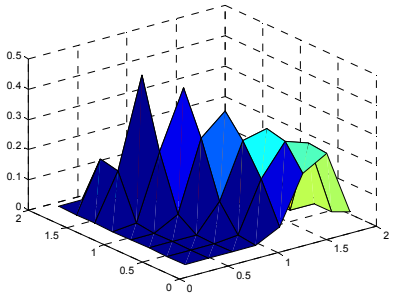
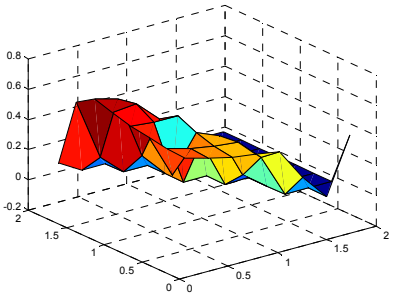
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 79: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1 avec une distribution de la taille des paquets suivant la loi de Poisson.

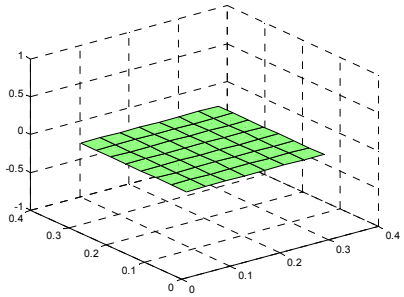
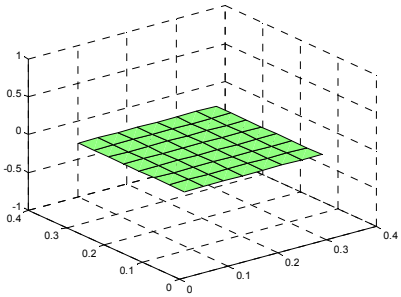
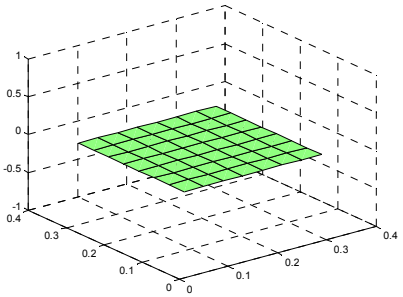
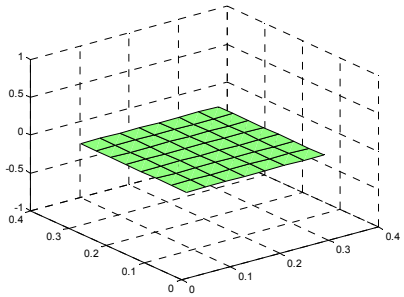
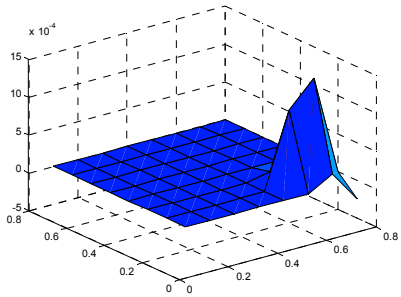
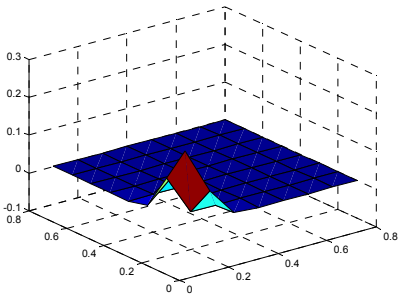
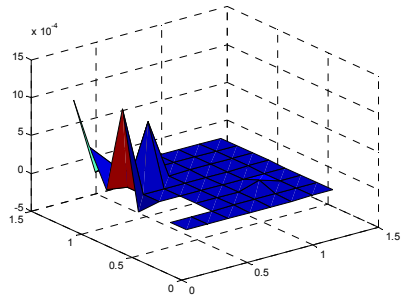
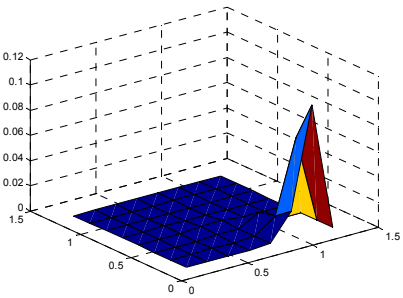
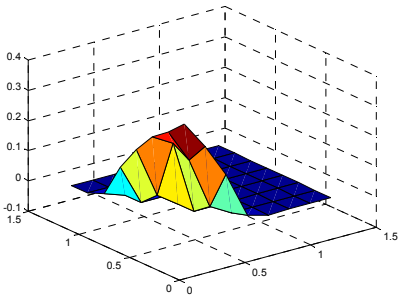
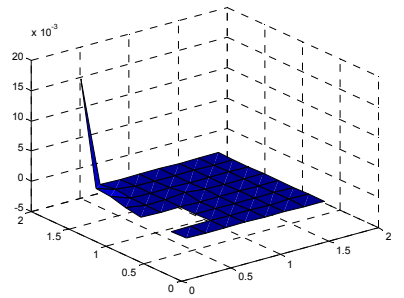
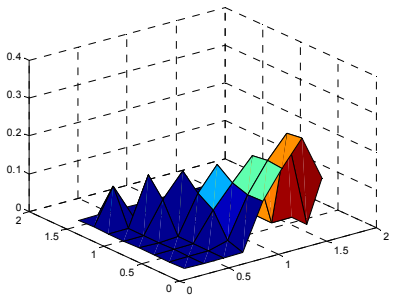
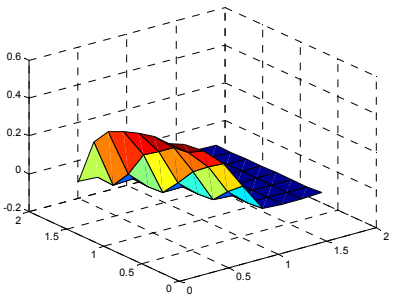
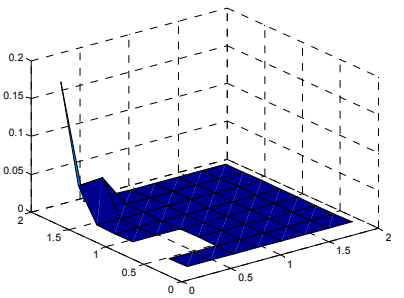
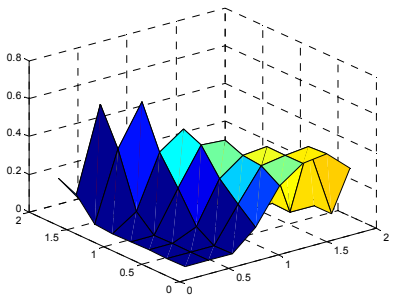
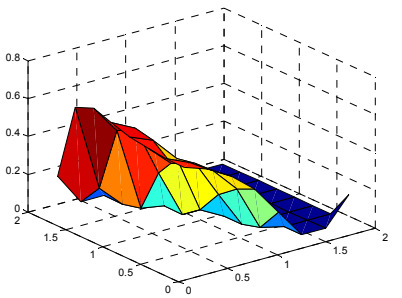
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 80: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/3/3 avec une distribution de la taille des paquets suivant la loi de Pareto.

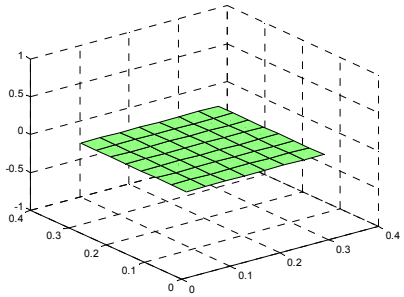
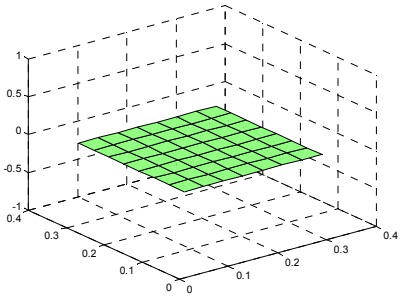
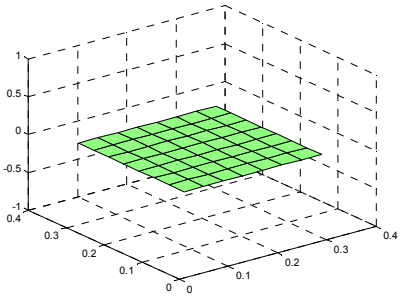
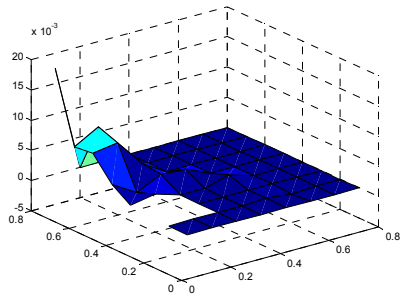
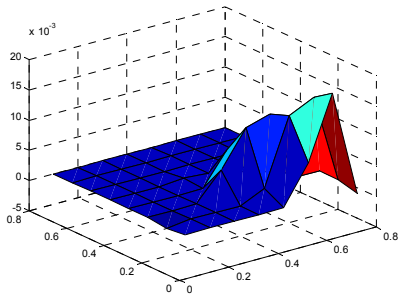
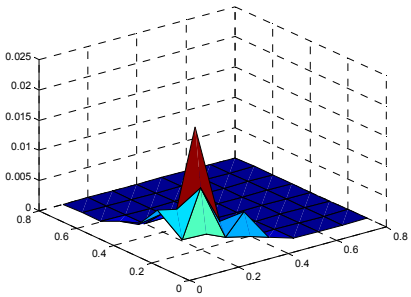
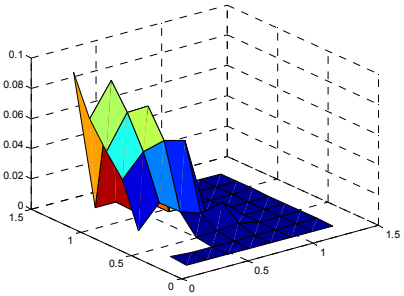
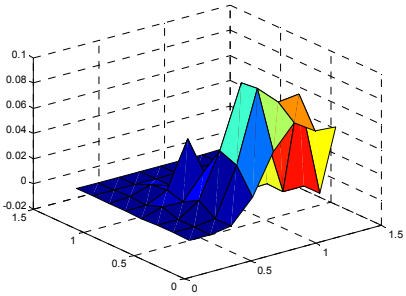
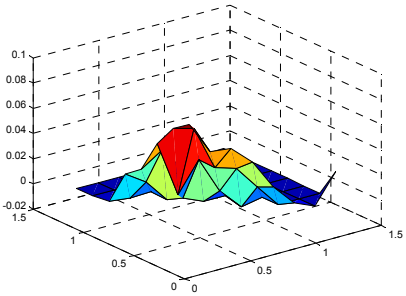
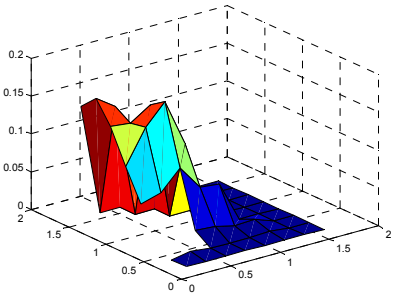
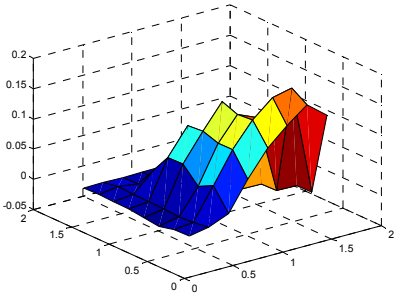
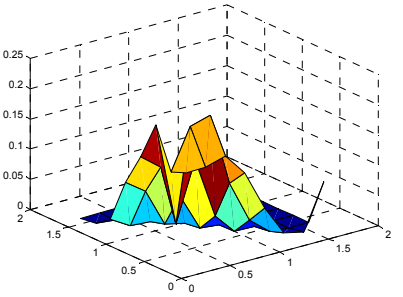
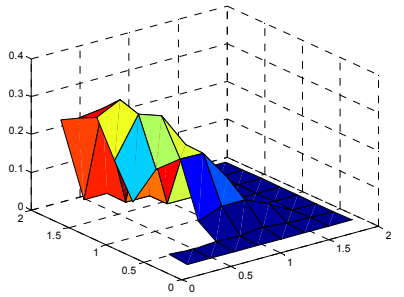
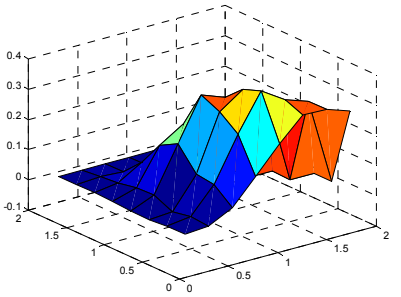
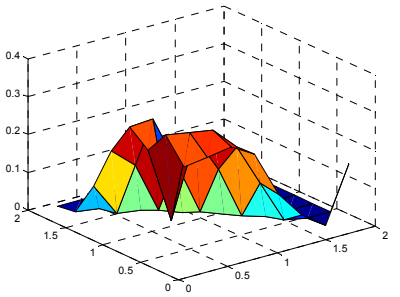
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 81: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 4/4/2 avec une distribution de la taille des paquets suivant la loi de Pareto.

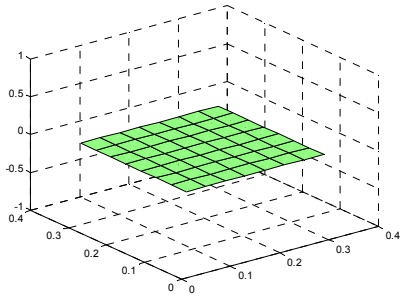
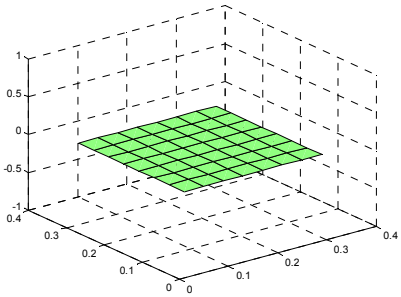
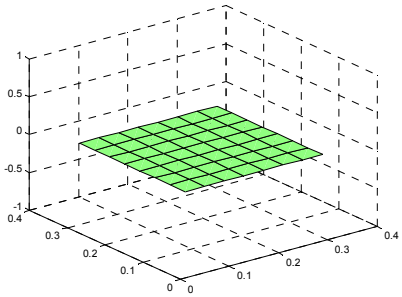
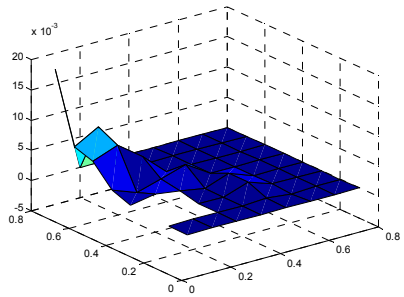
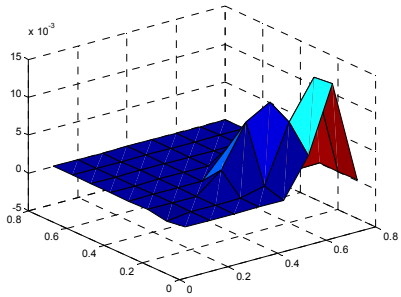
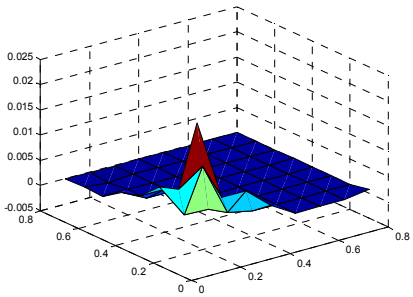
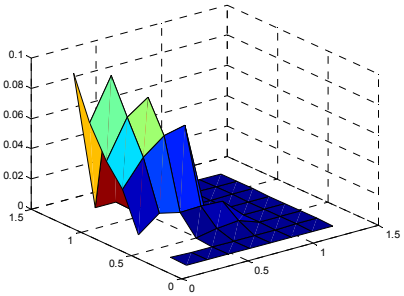
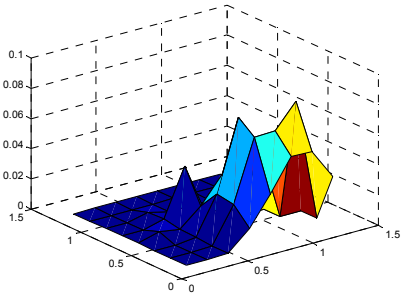
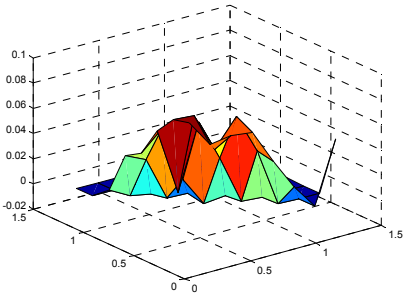
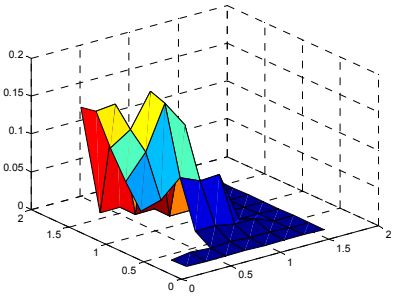
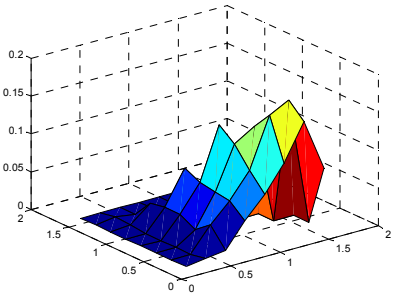
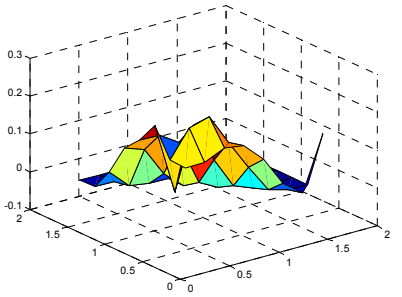
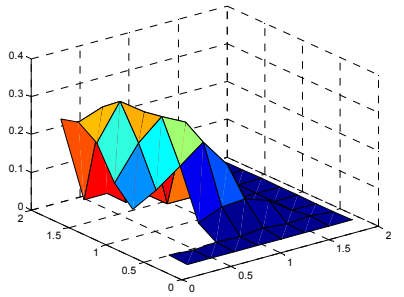
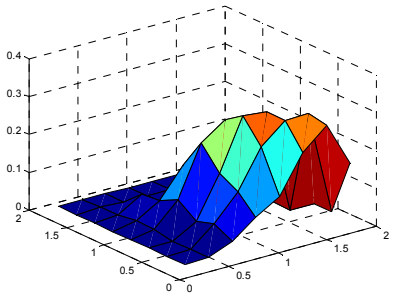
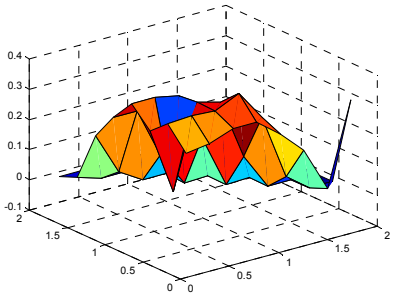
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 82: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/3/2 avec une distribution de la taille des paquets suivant la loi de Pareto.

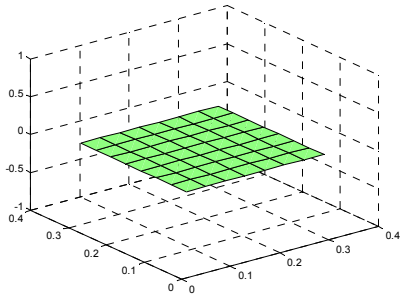
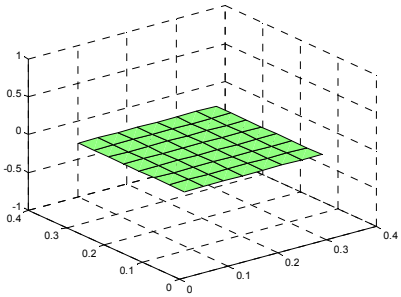
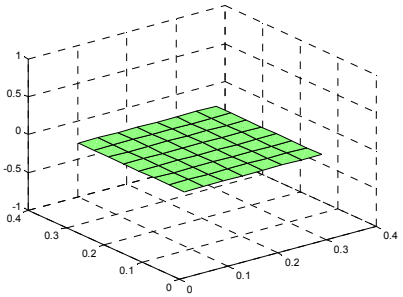
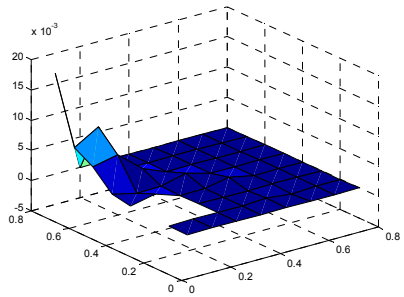
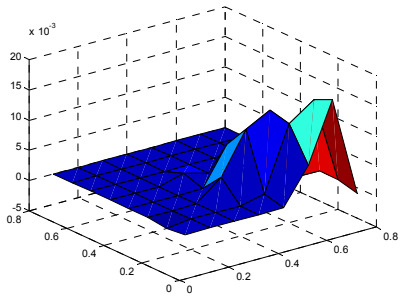
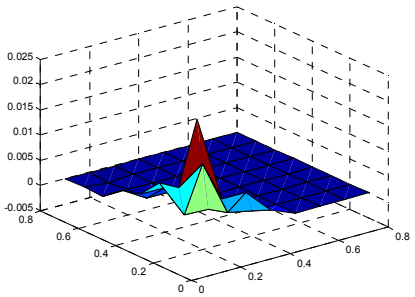
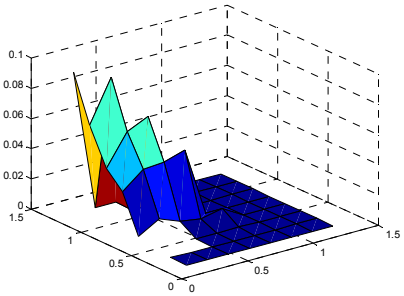
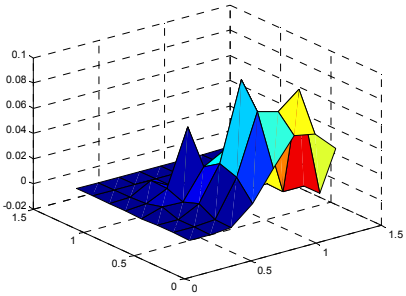
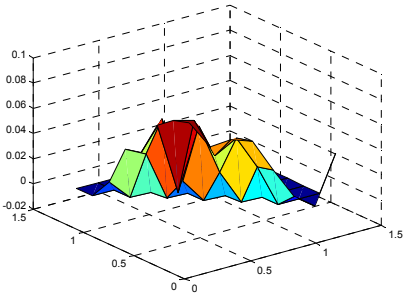
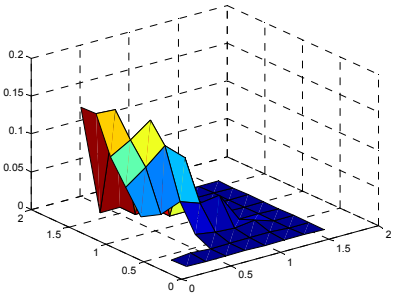
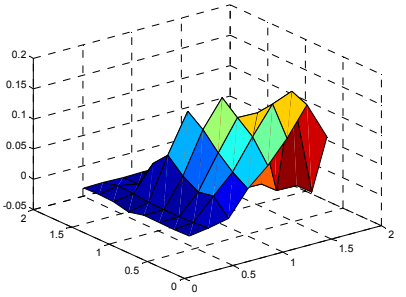
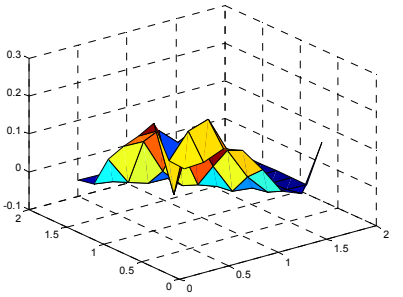
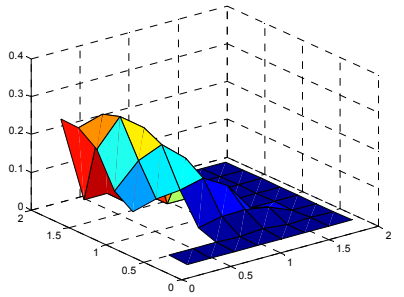
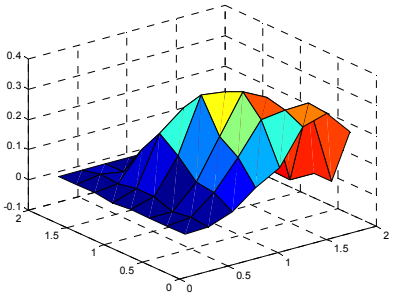
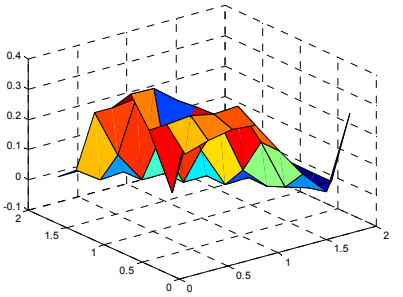
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 83: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 5/4/1 avec une distribution de la taille des paquets suivant la loi de Pareto.

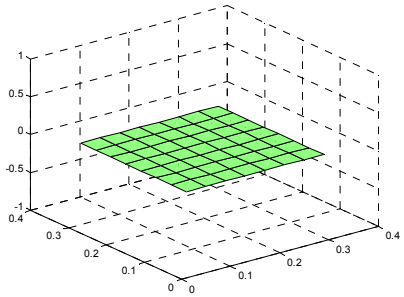
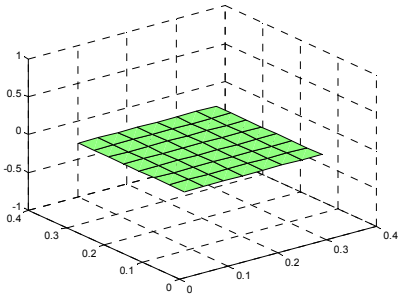
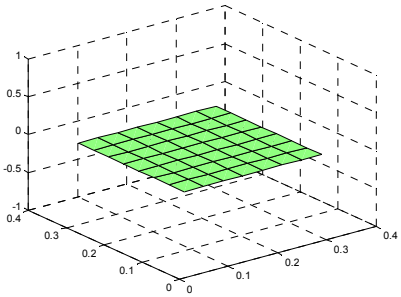
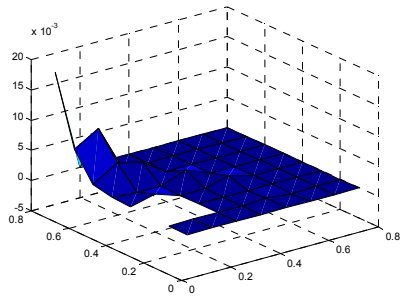
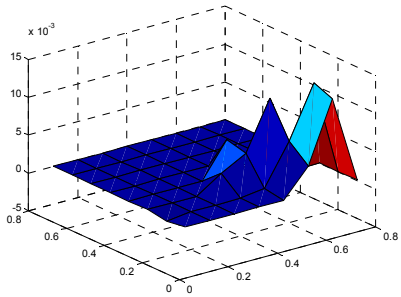
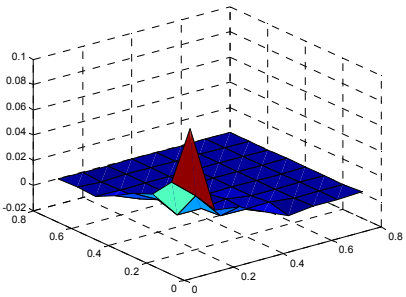
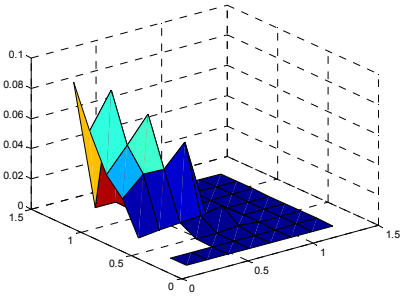
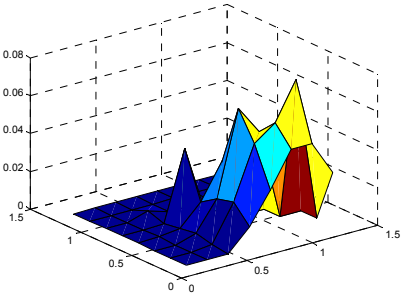
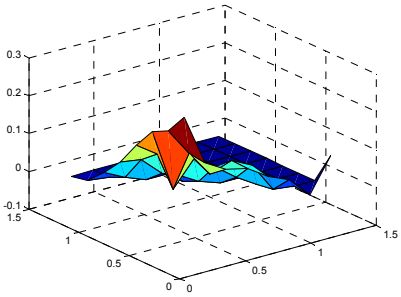
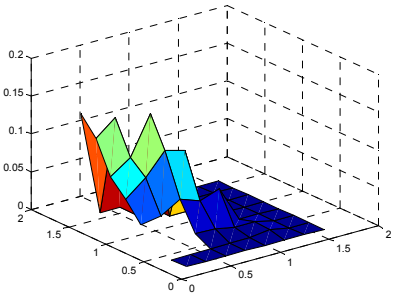
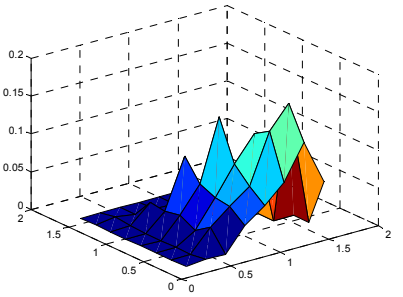
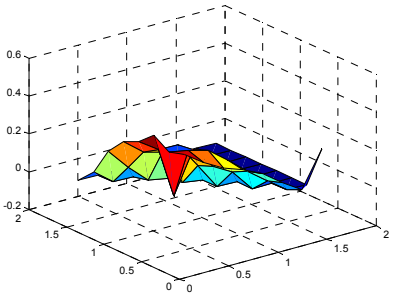
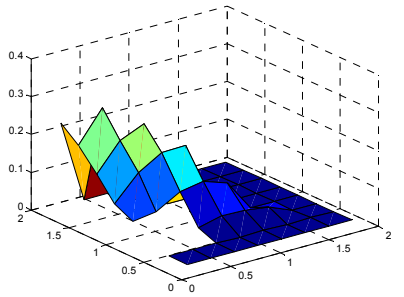
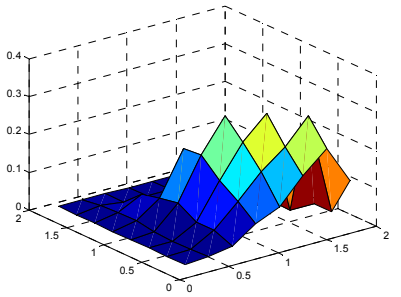
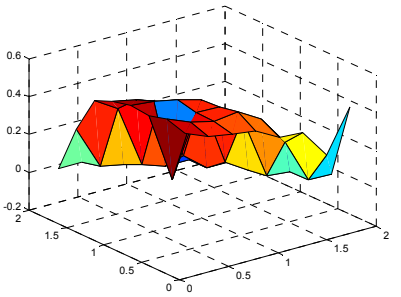
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 84: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/2/2 avec une distribution de la taille des paquets suivant la loi de Pareto.

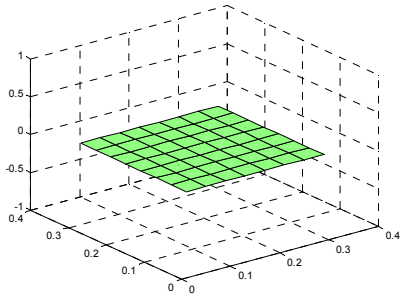
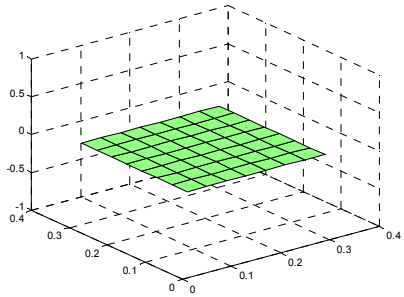
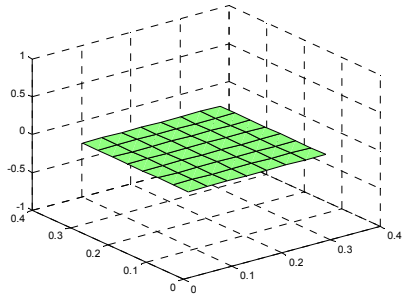
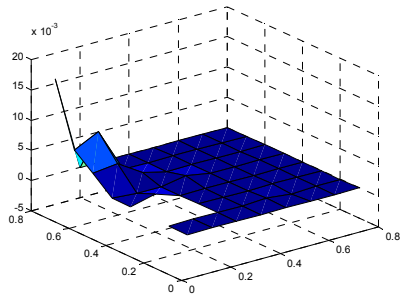
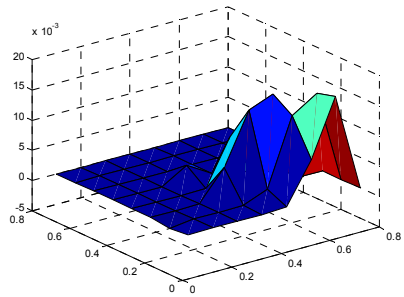
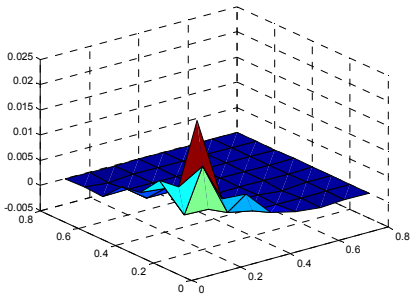
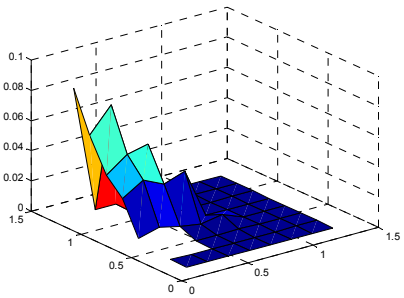
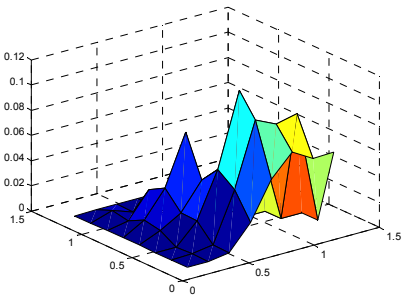
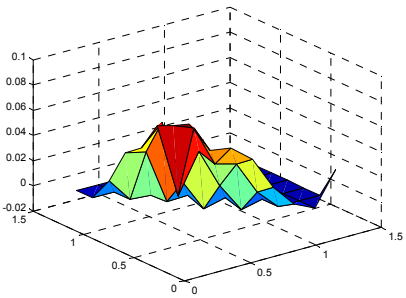
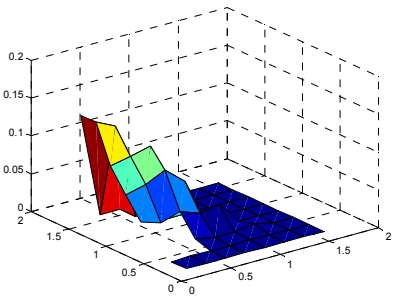
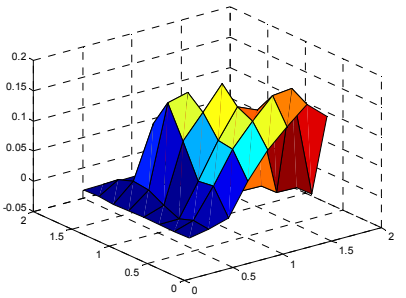
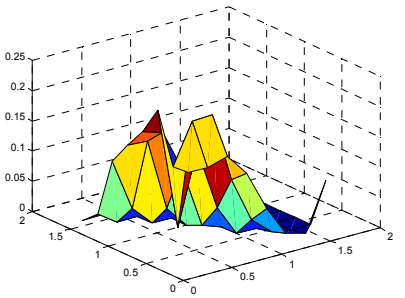
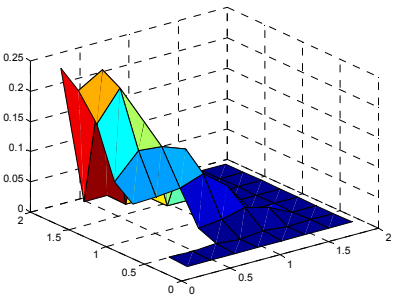
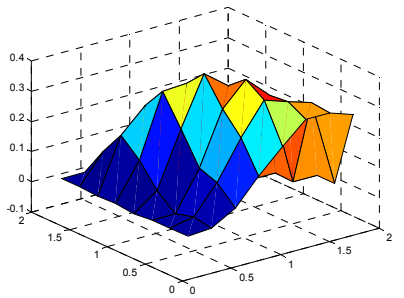
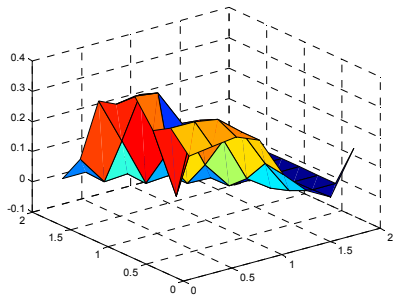
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 85: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 6/3/1 avec une distribution de la taille des paquets suivant la loi de Pareto.

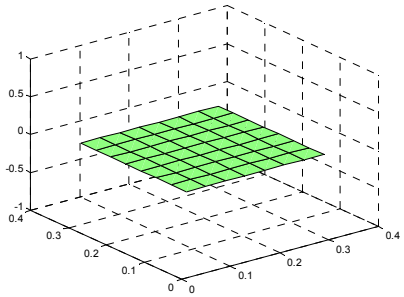
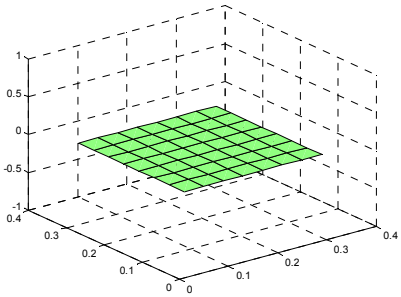
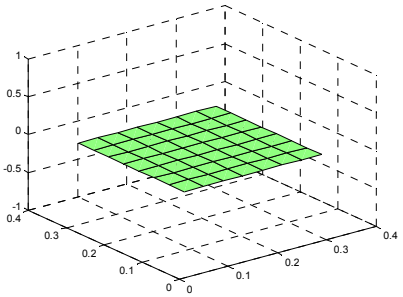
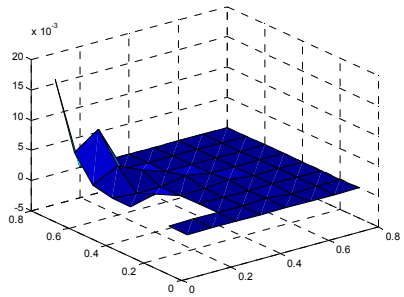
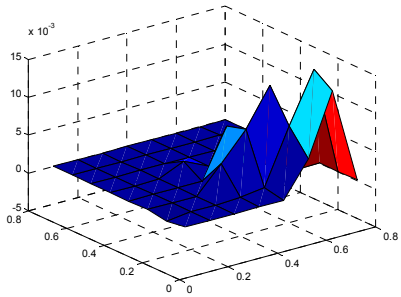
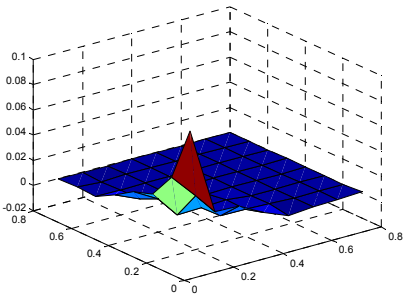
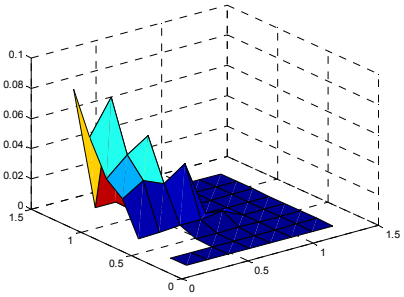
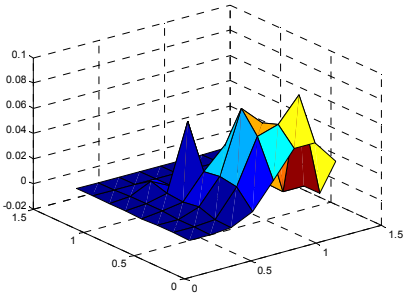
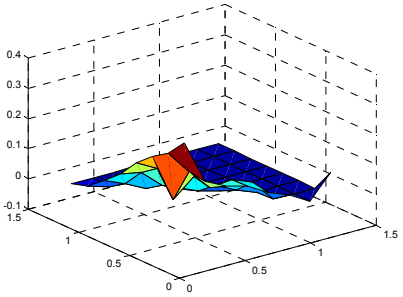
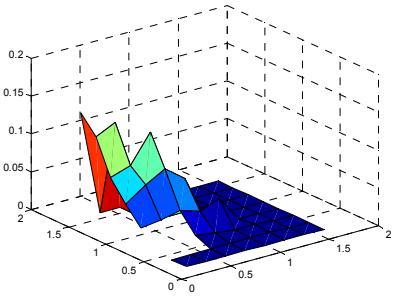
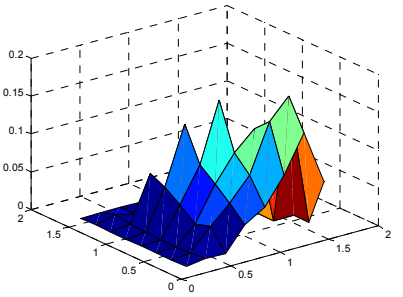
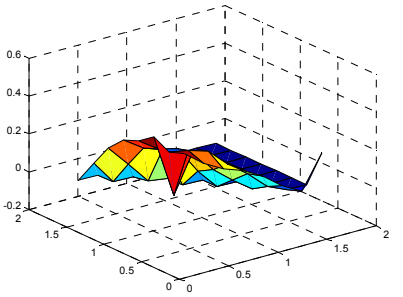
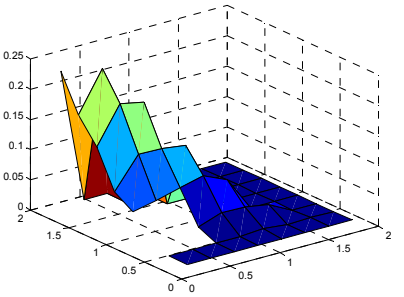
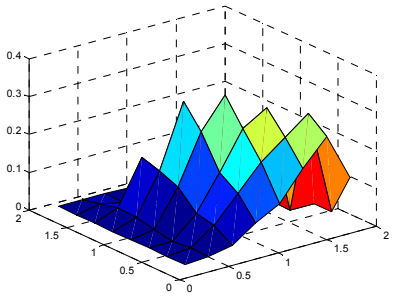
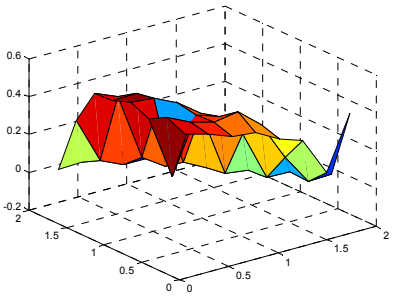
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 86: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 7/2/1 avec une distribution de la taille des paquets suivant la loi de Pareto.

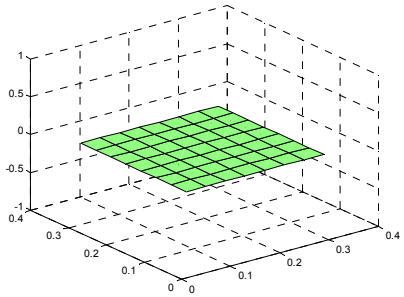
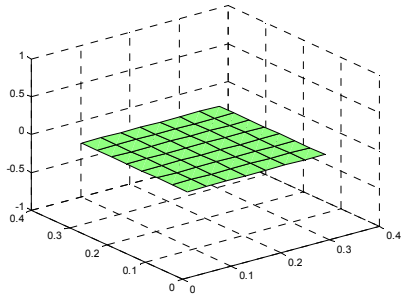
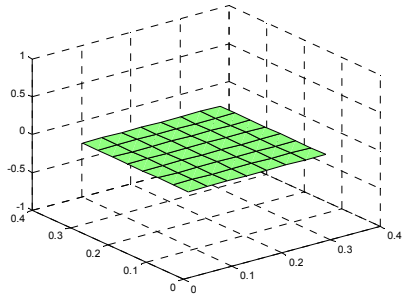
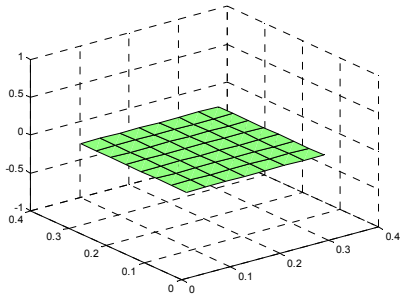
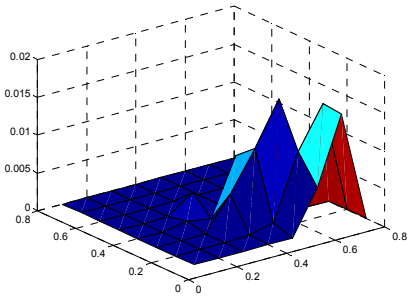
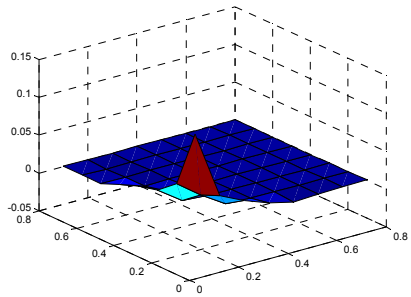
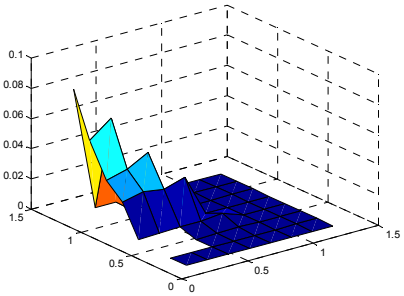
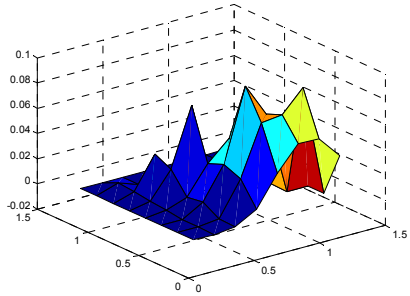
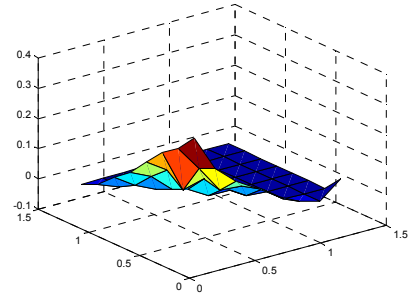
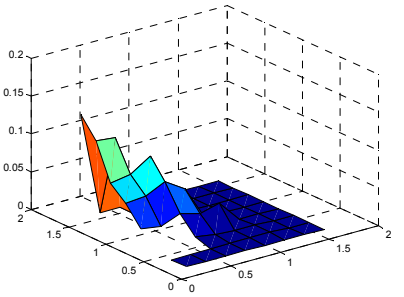
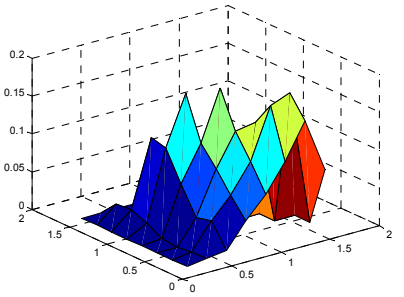
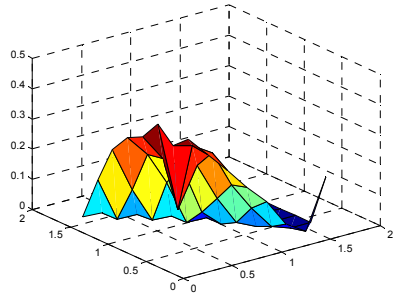
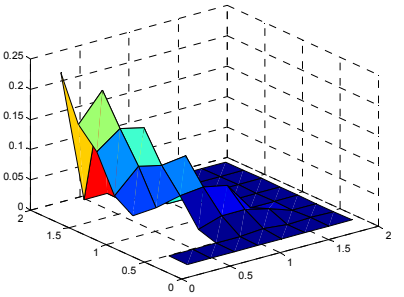
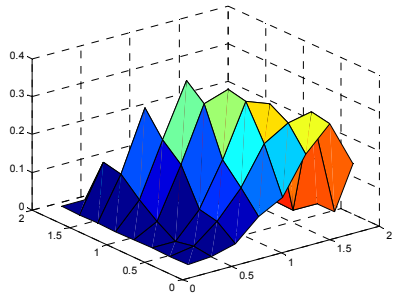
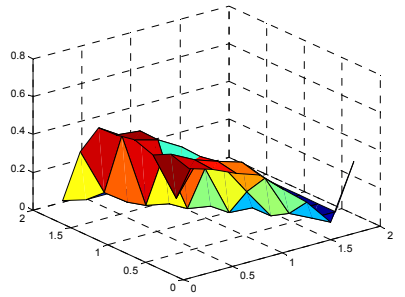
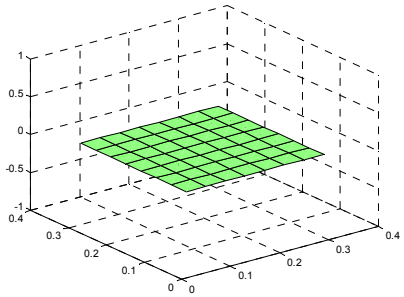
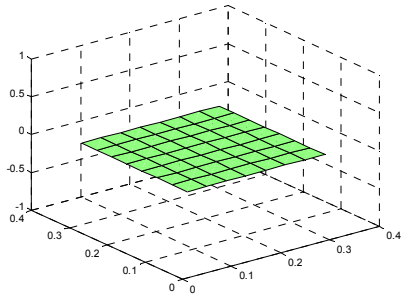
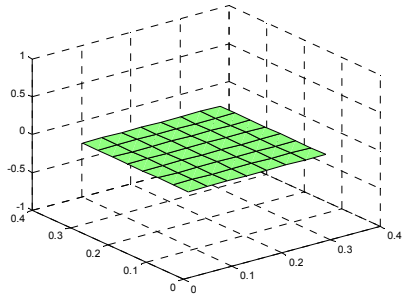
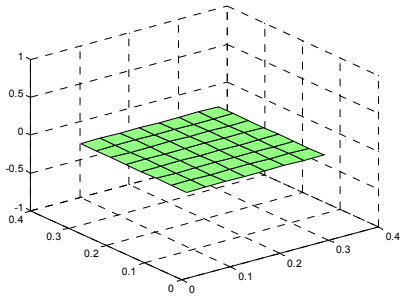
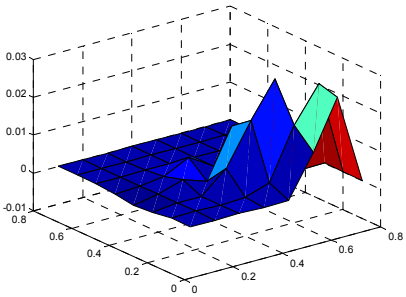
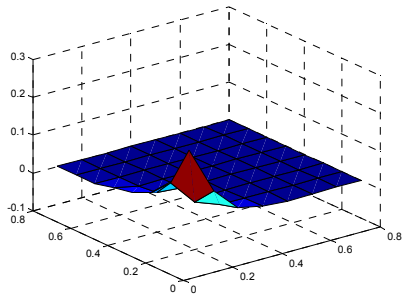
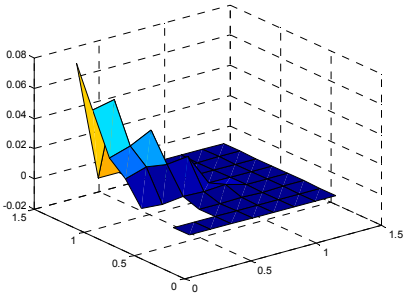
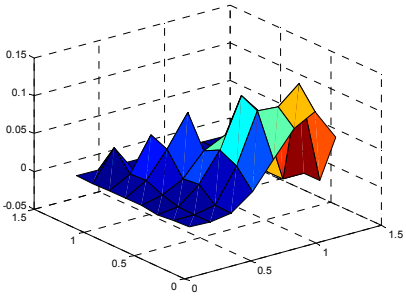
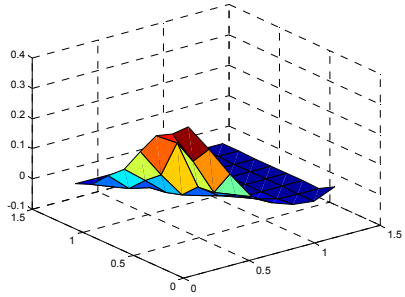
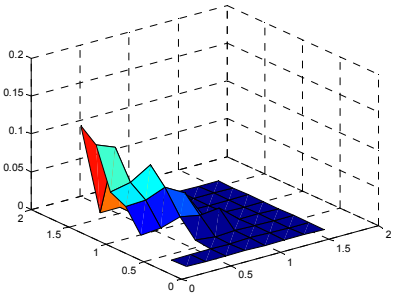
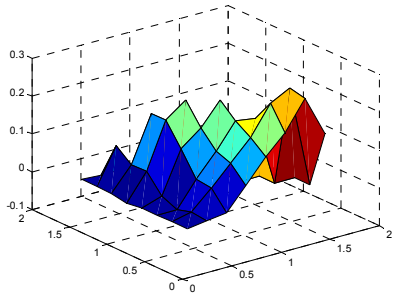
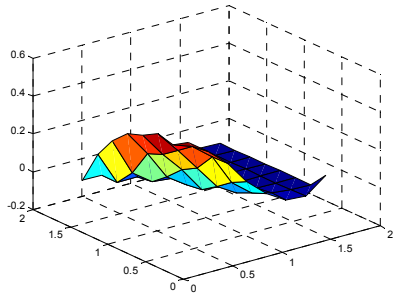
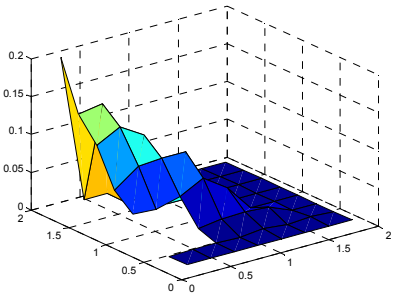
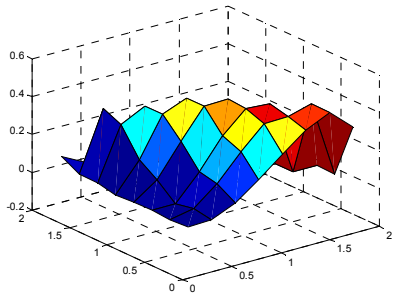
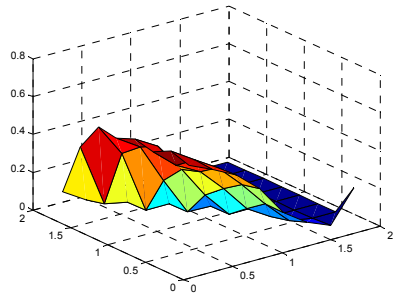
Classe 1	Classe 2	Classe 3
		
		
		
		
		

Tableau 87: La variation du taux de perte en fonction de la charge des classes 1 et 2 pour une pondération 8/1/1 avec une distribution de la taille des paquets suivant la loi de Pareto.

Classe 1	Classe 2	Classe 3
		
		
		
		
		

ANNEXE 5 :
RESULTATS COMPLEMENTAIRES DE L'ORDONNANCEMENT

Cette annexe fait suite au paragraphe 3.4.4.4.

1. Performance de CB-WRR dans le cas 1.

Dans ce paragraphe, les mesures des gigues et des taux de perte de l'ordonnanceur CB-WRR avec les paramètres et les flux notés "***cas 1***" sont illustrées.

Les figures ci-dessous (Figure 98, Figure 99 et Figure 100) illustrent les valeurs des gigues pour les différentes valeurs du poids du trafic AF.

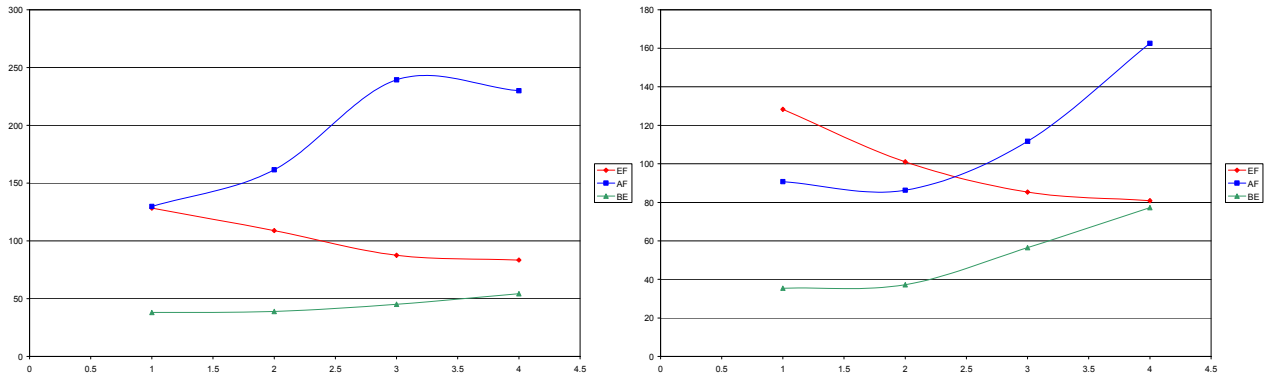


Figure 98: Gigue des flux (1/1) et (2/1)

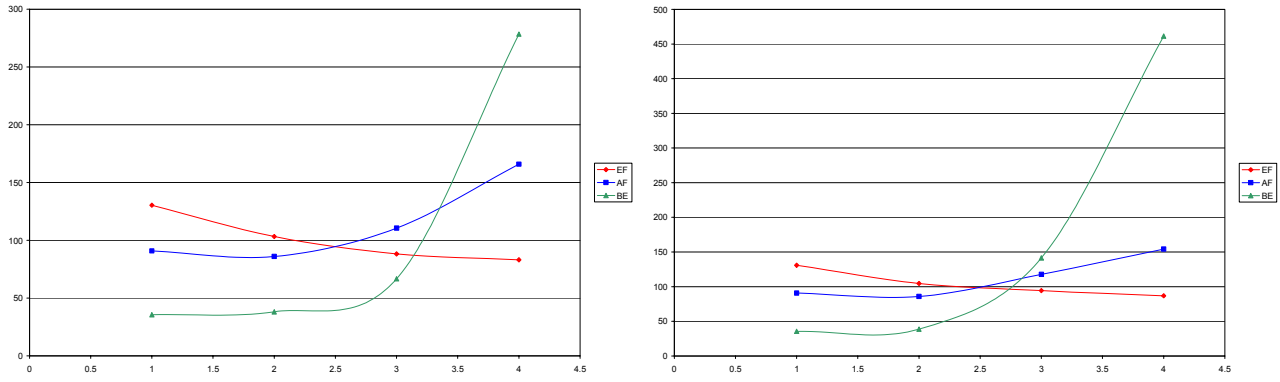


Figure 99: Gigue des flux (3/1) et (4/1)

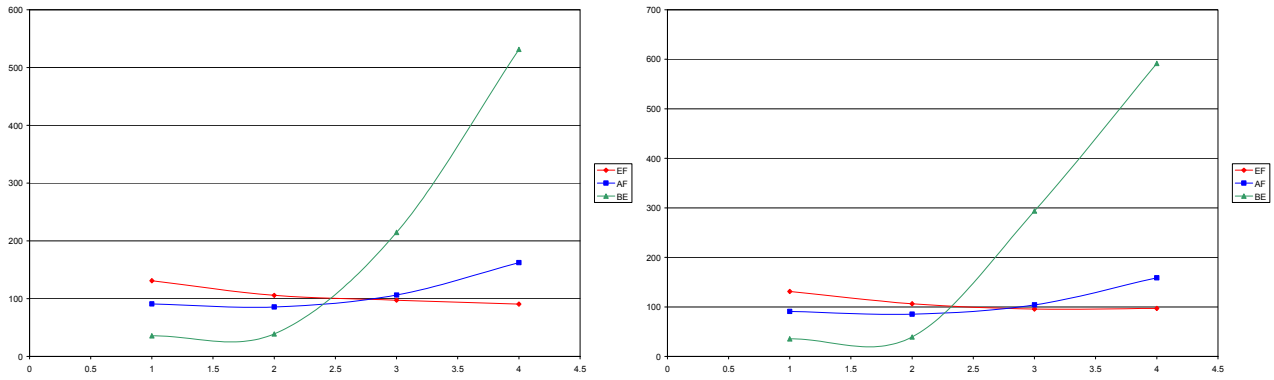


Figure 100: Gigue des flux (5/1) et (6/1)

Les figures ci-dessous (Figure 101, Figure 102 et Figure 103) illustrent les valeurs des taux de perte pour les différentes valeurs du poids des flux AF.

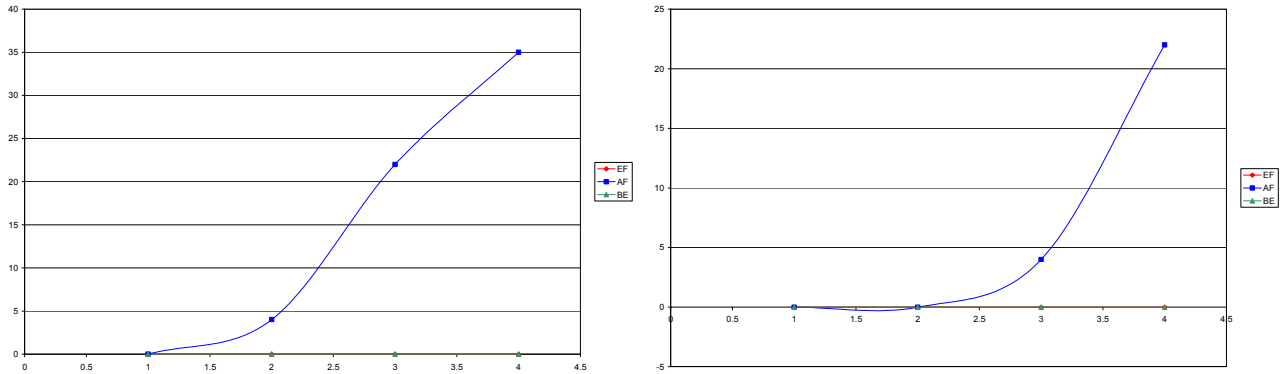


Figure 101: Perte des flux (1/1) et (2/1)

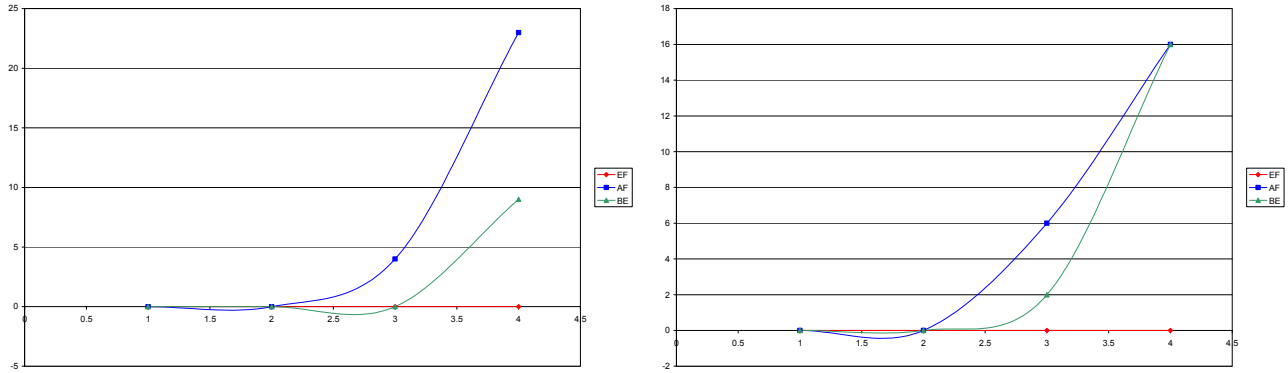


Figure 102: Perte des flux (3/1) et (4/1)

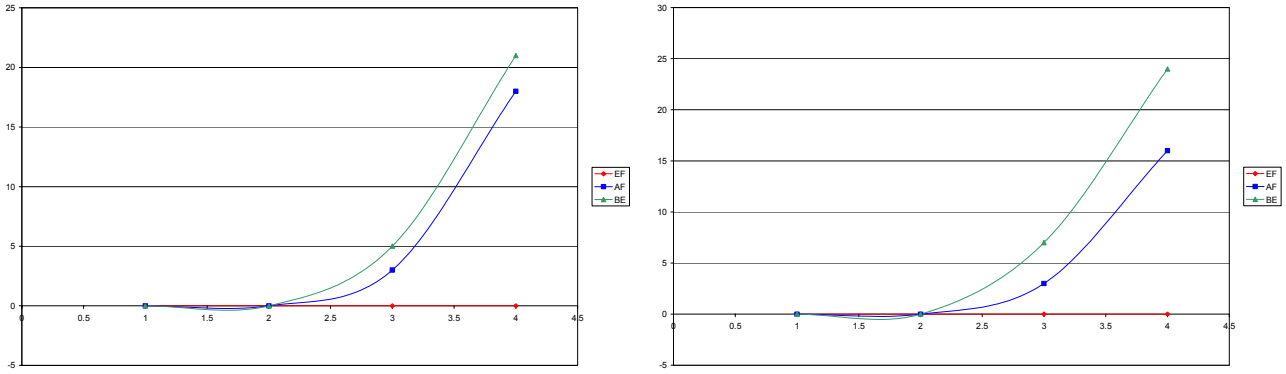


Figure 103: Perte des flux (5/1) et (6/1)

Les courbes des délais, gigues et taux de perte convergent vers la même interprétation: du fait du rapport du trafic AF et BE qui circule dans le réseau et pour avoir une QoS des flux AF meilleure que celle des flux BE, il faudra implémenter au niveau des ordonnanceurs des poids respectifs de 6 et de à 1.

2. Performance de CB-WRR dans le cas 2.

Les figures ci-dessous (Figure 104, Figure 105 et Figure 106) illustrent les valeurs des taux de perte pour les différentes valeurs du poids du trafic AF/BE.

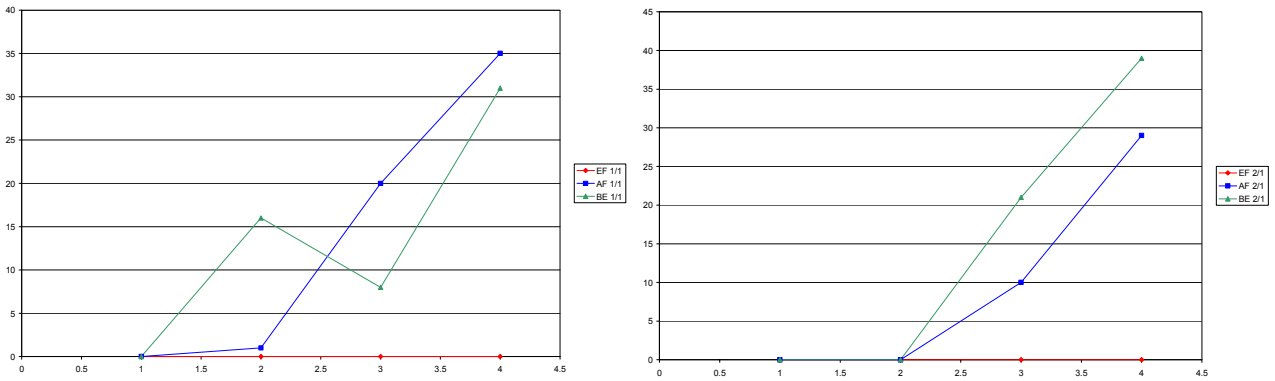


Figure 104: Taux de perte des flux (1/1) et (2/1)

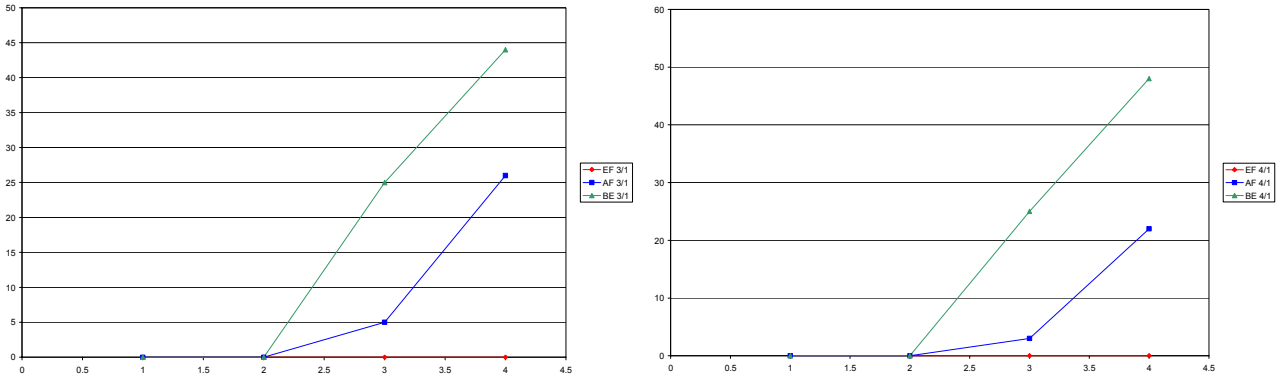


Figure 105: Taux de perte des flux (3/1) et (4/1)

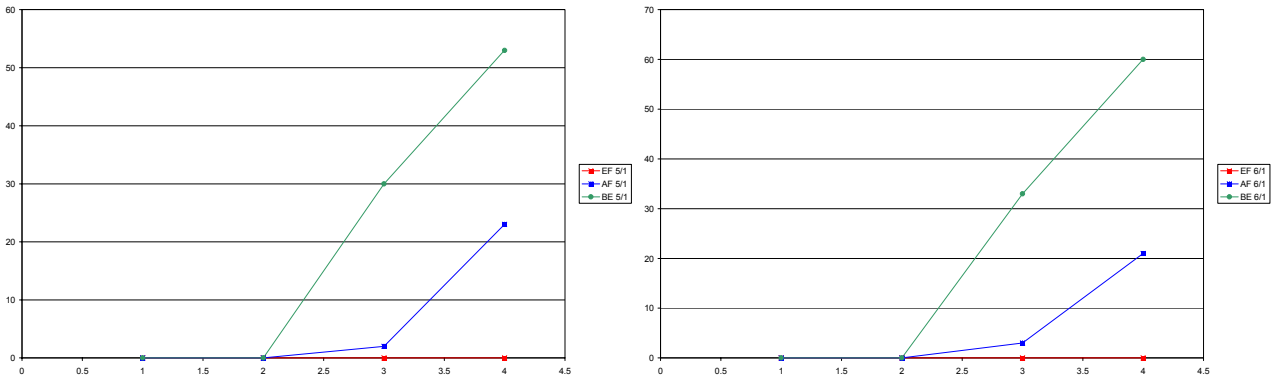


Figure 106: Taux de perte des flux (5/1) et (6/1)

Les figures ci-dessous (Figure 107 à Figure 109) illustrent les valeurs de la gigue pour les différentes valeurs du poids du trafic AF/BE.

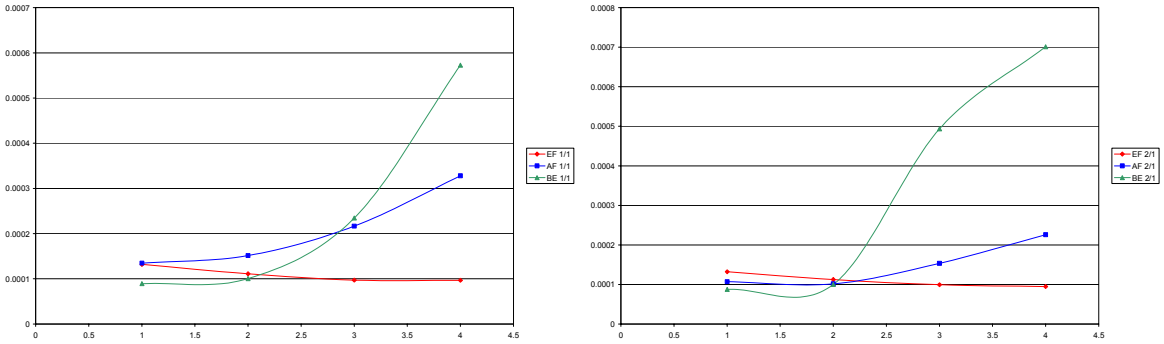


Figure 107: La gigue des flux (1/1) et (2/1)

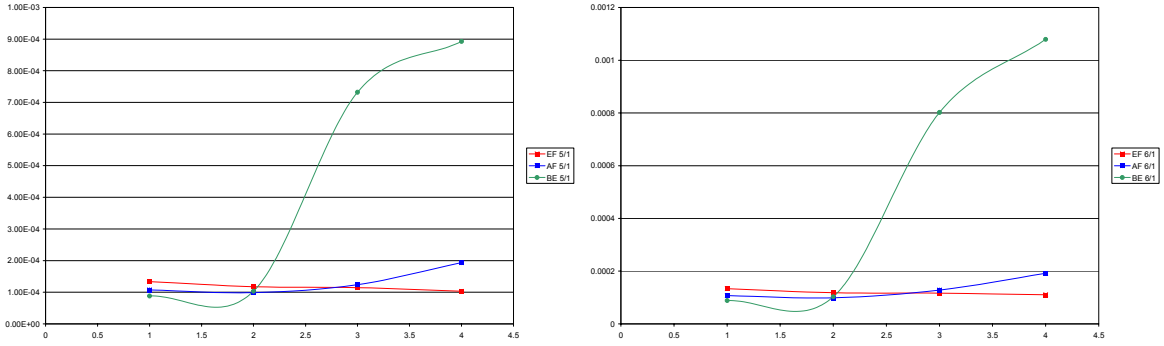


Figure 108: La gigue des flux (3/1) et (4/1)

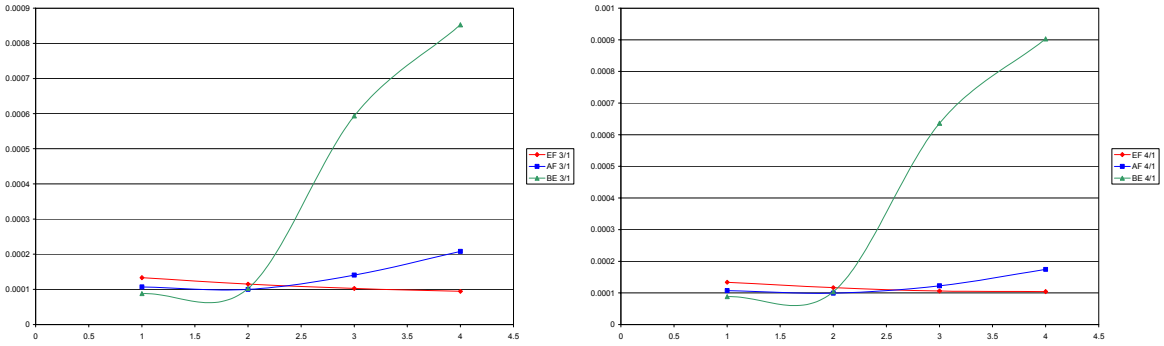


Figure 109: La gigue des flux (5/1) et (6/1)

3. Performance de CB-WRR dans le cas 3.

Les figures ci-dessous (Figure 110 à Figure 112) illustrent les valeurs du taux de perte pour les différentes valeurs du poids du trafic AF/BE.

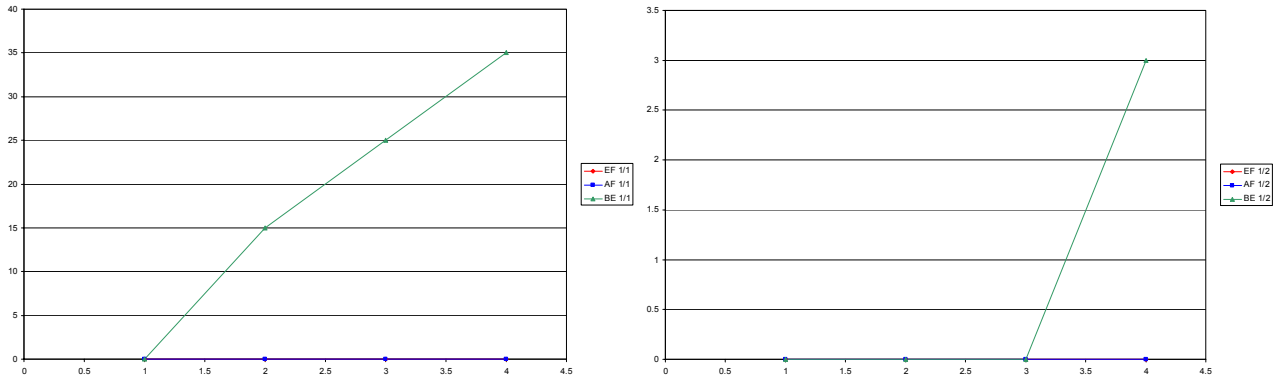


Figure 110: Taux de perte des flux (1/1) et (1/2)

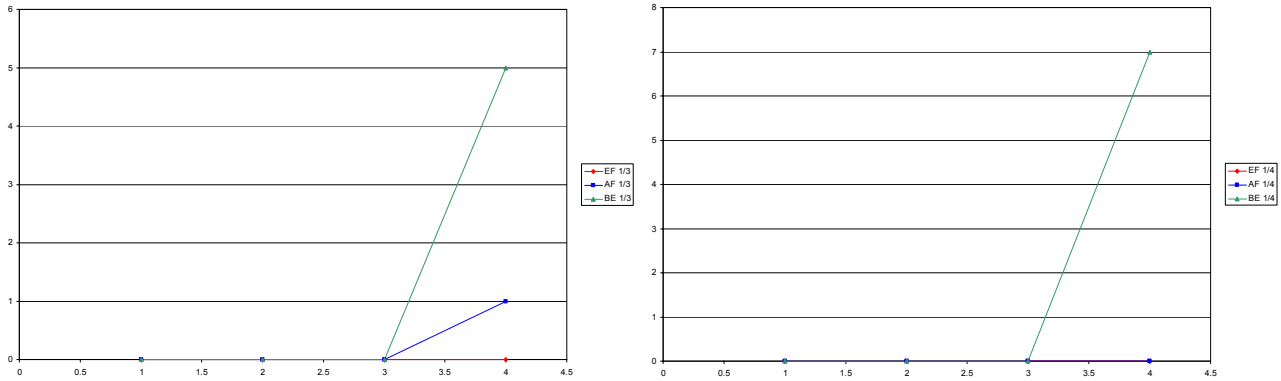


Figure 111: Taux de perte des flux (1/3) et (1/4)

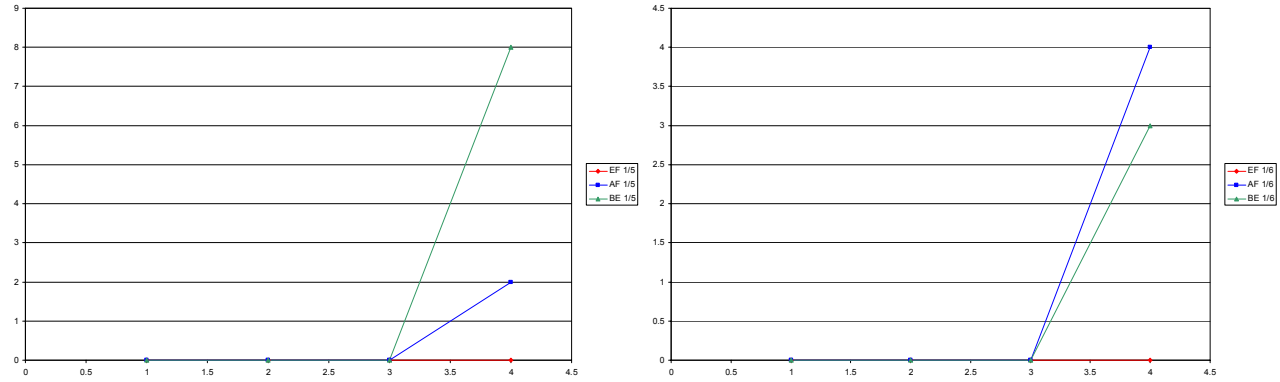


Figure 112: Taux de perte des flux (1/5) et (1/6)

Les figures ci-dessous (Figure 113 à Figure 115) illustrent les valeurs des gigues pour les différentes valeurs du poids du trafic AF/BE.

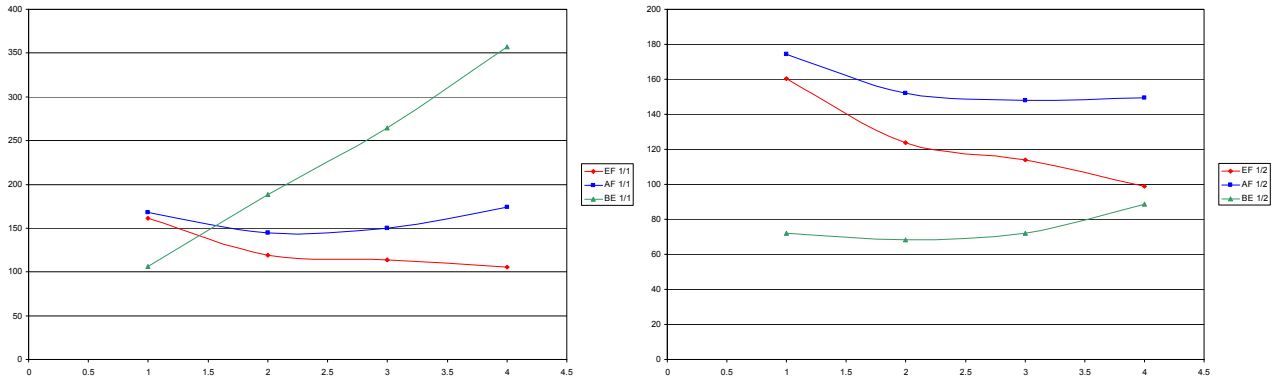


Figure 113: La gigue des flux (1/1) et (1/2)

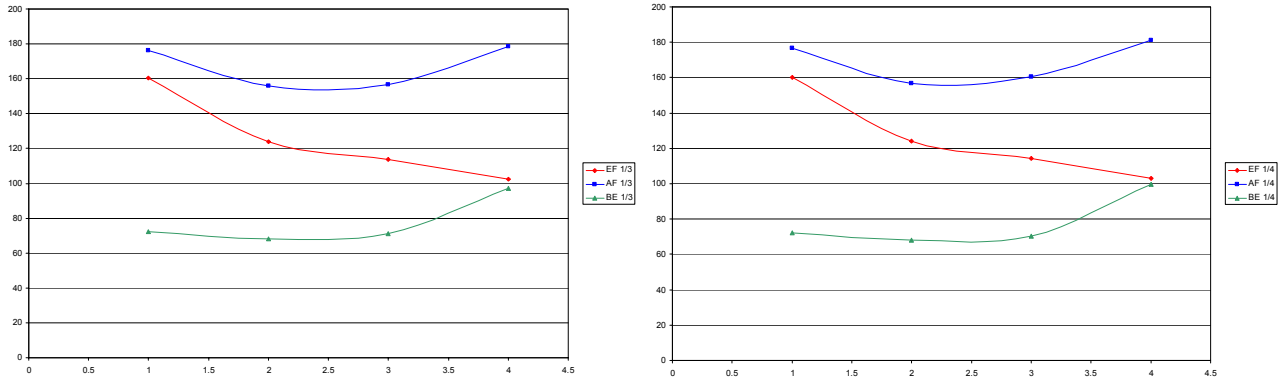


Figure 114: La gigue des flux (1/3) et (1/4)

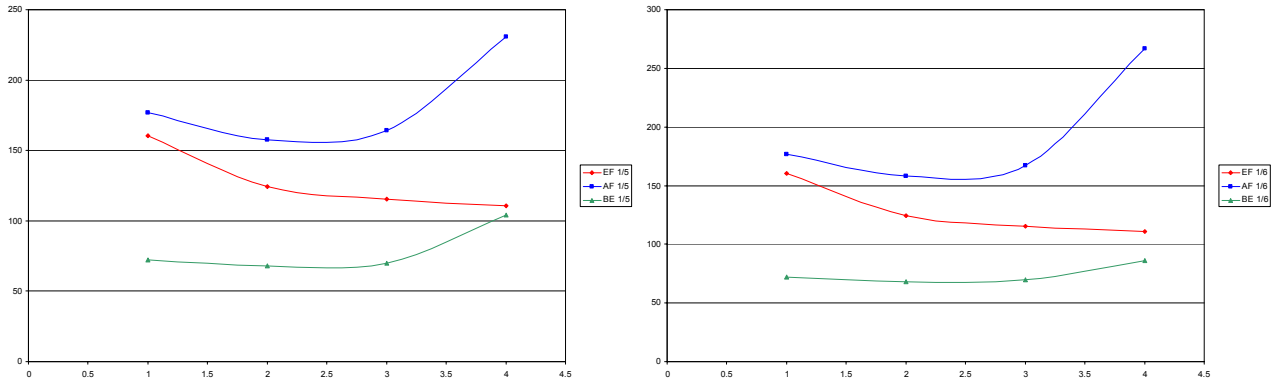


Figure 115: La gigue des flux (1/5) et (1/6)

4. Performance des flux en fonction des ordonnanceurs

a. Le taux de perte

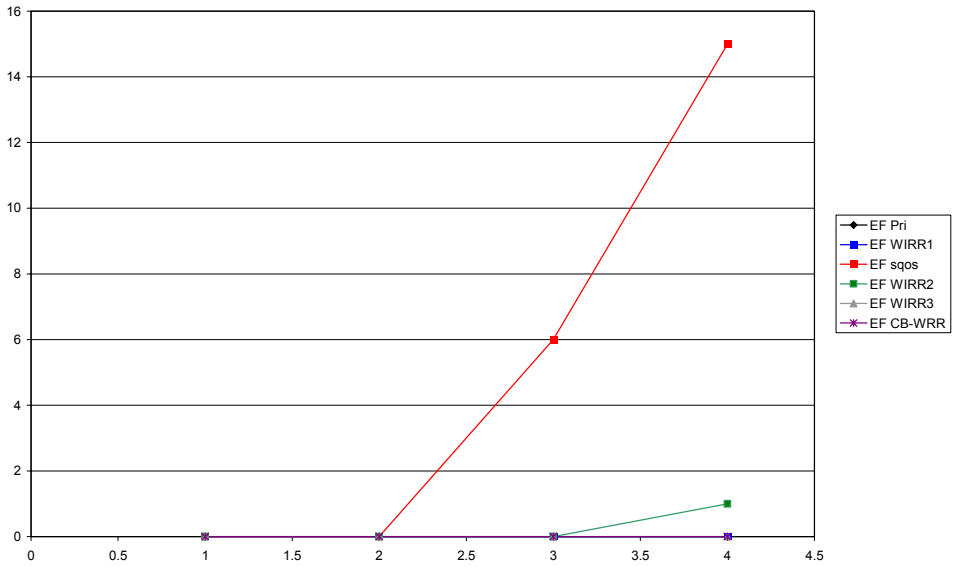


Figure 116: Perte EF

Dans la Figure 116 nous observons que le taux de perte des flux EF est toujours nul sauf pour les cas sans QoS et WIRR2. Dans ce dernier cas, les poids implémentés sont 4/4/2. Cette pondération est mauvaise.

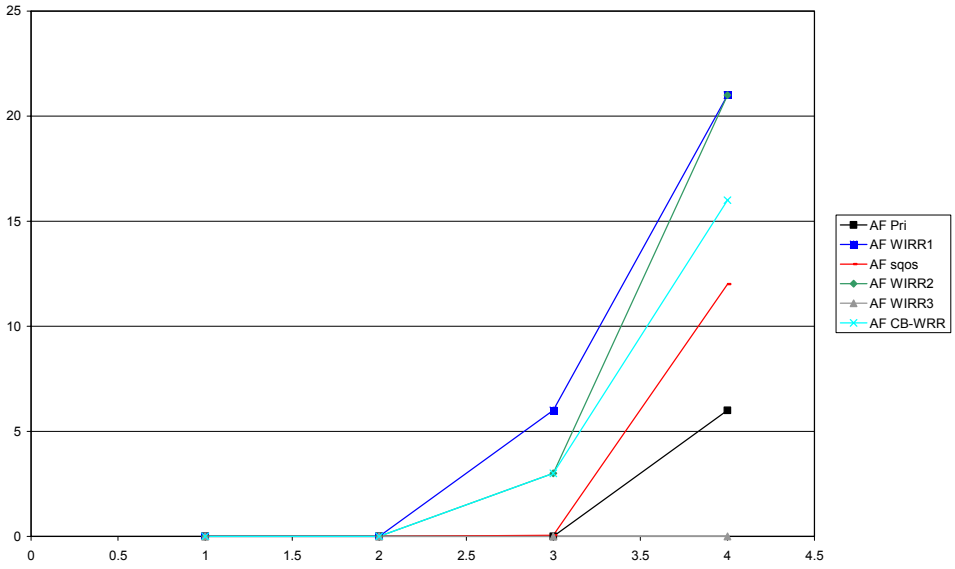


Figure 117: Taux de perte AF

Les flux AF peuvent transiter sans perdre de paquets dans le cas d'un partage de charge uniquement.

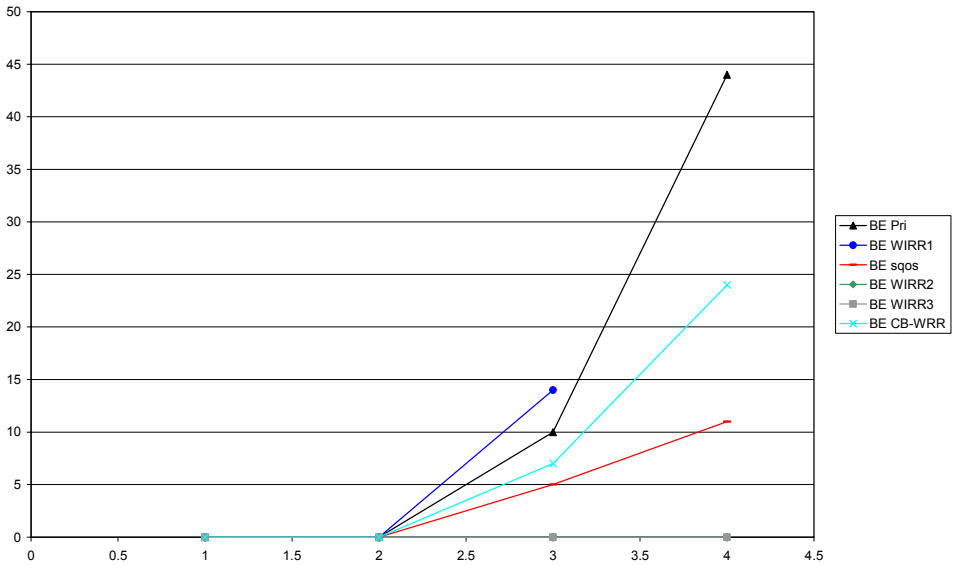


Figure 118: Perte BE

Dans le cas de partage de charge, les paquets BE ne subissent pas de perte. Mais aussi, quand le poids associé aux flux BE est supérieur à sa part réelle (WIRR2 avec 20% de la bande), le taux de perte est nul. Dans les autres cas restant, les flux BE sont servis moins bien que dans le cas où la QoS n'est pas implémentée dans le réseau.

b. Le délai moyen

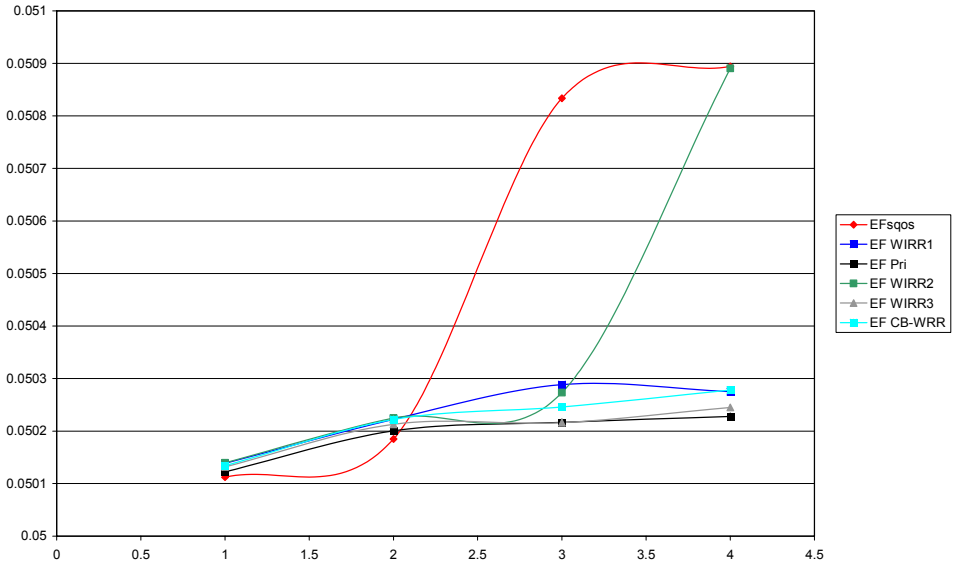


Figure 119: Le délai EF

Le délai moyen le plus faible est celui de la priorité absolue des flux EF par rapport aux autres flux. Avec le partage de charge, le délai est faible : les flux sont divisés sur deux routes différentes. Le délai CB-WRR est un peu plus élevé. C'est la perte de temps pour exécuter l'algorithme.

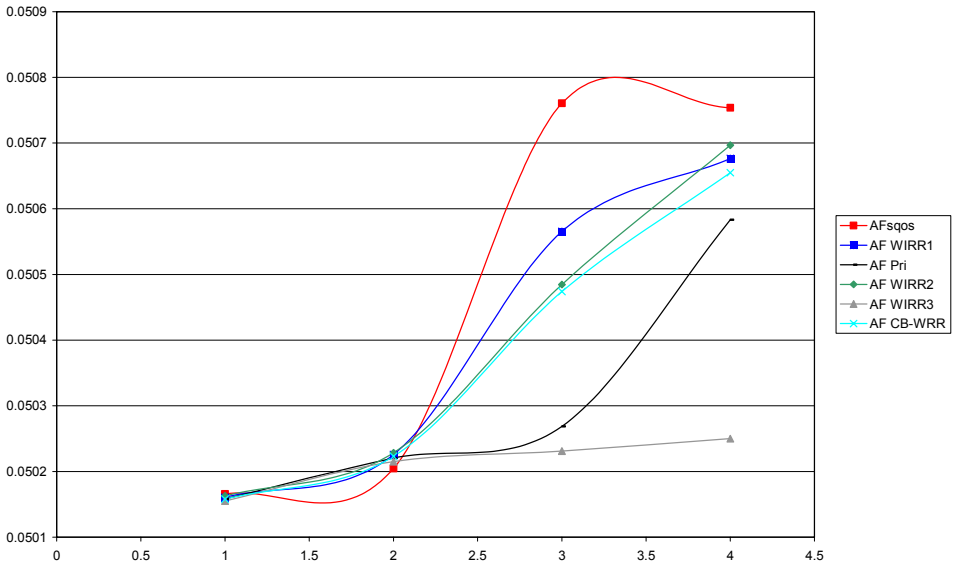


Figure 120: Délai AF

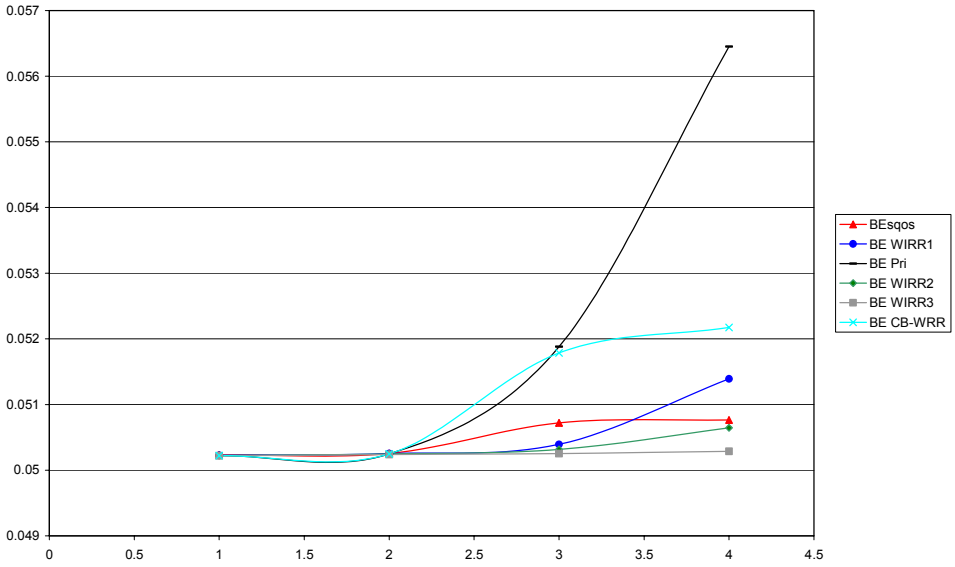


Figure 121: Délai BE

Glossaire

AF	Assured Forwarding
AP	Access Point
AS	Autonomous System
ATM	Asynchronous Transfer Mode
BE	Best Effort
BER	Bit Error Ratio
BGP	Border Gateway Protocol
CDMA	Code Division Multiple Access
CDMA/DS	CDMA Direct Sequence
CFP	Contention Free Period
CN	Core Network
CP	Contention Period
CR-LDP	Constraint-based Routing Label Distribution Protocol
CSMA/CA	Carrier Sense Multiple Access with Collision avoidance
CTS	Clear to Send
CW	Congestion Window
DCF	Distributed Coordination Function
DiffServ	Differentiated Services
DS	Distributed System
DSCP	Differentiated Services Code Point
EDD	Earliest Due Date
EF	Expedited Forwarding
EM	Expectation Maximisation
EM	Expectation Maximisation
ER-LSP	Explicit Routed LSP
FEC	Forward Equivalence Class
FTP	File Transfer Protocol
GPS	Generalised Processor Sharing
HMM	Hidden Markov Model
HMM	Hidden Markov Model
HOL	Head Of the Line
IEEE	Institute of Electrical and Electronics Engineers
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IntServ	Integrated Services
IP	Internet Protocol

IS-IS	Intermediate System-Intermediate System
ISM	Industrial Scientific and Medical
ITC	Intermediary Transfer Capability
LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switch Pass
LSR	Label Switch Router
MAC	Medium Access Control
MAI	Multiple Access Interference
MLE	Maximum Likelihood Estimation
MLE	Maximum Likelihood Estimation
MMPP	Markov Modulated Poisson Process
MPLS	Multi Protocol Label Switching
MTC	MAC Transfer Capabilities
OSPF	Open Shortest Path First
PC	Point Coordinator
PCF	Point Coordination Function
PGPS	Packet per packet Generalised Processor Sharing
PHB	Per Hop Behaviour
PIFS	PCF interframe space
PN	Pseudo random Noise code
QoS	Quality of Service
RED	Random Early Detection
RIP	Routing Information Protocol
RSVP	Resource ReserVation Protocol
RSVP-TE	RSVP-Traffic Engineering
RTC	Rapid Transfer Capability
RTS	Ready to Send
SF	Spreading Factor
SGA	Standard Gaussian Approximation
SIFS	Short Interframe space
SNR	Signal to Noise Ratio
SPF	Shortest Path First
STC	Slow Transfer Capability
S-VAD	Slow Voice Activity Detector
TB	Transport Block
TCA	Traffic Conditioning Agreement
TDMA	Time Division Multiple Access
UMTS	Universal Mobile Telecommunication Service
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice over IP
VPN	Virtual Private Network
WCDMA	Wideband Code Division Multiple Access
WFQ	Weighted Fair Queuing
WiFi	Wireless Fidelity
WM	Wireless Medium
WRR	Weighted Round Robin

Publications

Articles de conferences

H. Aouad, S. Tohme, "Using network tomography for dynamic path adaptation ", *IEEE Wireless Communications and Networking Conference, 2005. WCNC 2005.*

H. Aouad, A. Ibrahim, S. Tohme, "UTRAN traffic parameters estimation ", *IEEE Consumer Communications and Networking Conference, 2004. CCNC 2004*.

Rapports du projet RNRT ESQUIMAUX :

H. Aouad, et al. "Estimation des paramètres MMPP d'un trafic à la sortie d'un réseau UTRAN", Janvier 2003.

H. Aouad, et al. "Simulation et Evaluation de Performance d'un Réseau IP Intégrant Diffserv et MPLS", Novembre 2002.

H. Aouad, et al. "La Qualité de Service dans les Réseaux Basés Politiques ", Juin 2002.

H. Aouad, et al. "Qualité de Service et Intégration de MPLS dans les Réseaux Mobiles", Novembre 2001.

H. Aouad, et al. "Dossier état de l'art de la technologie MPLS sur IP ", Mars 2001.

Bibliographie

[1]. A. Baiocchi, N.B. Melazzi, M. Listanti, A. Roveri, R. Winkler, "Loss performance analysis of an ATM multiplexer loaded with high-speed on-off sources", *IEEE Journal on Selected Areas in Communications, Volume 9, Issue 3,* Avril 1991.

[2]. A. Ibrahim, S. Tohmé, "Service Multiplexing and Resource Allocation in S-CDMA LEO Systems", *Proceedings PIMRC, Lisboa, Portugal,* 2002.

[3]. A. Parekh and R. Gallager. "A generalized processor sharing approach to flow control in integrated services networks: The single node case", *IEEE/ACM Transactions on Networking*, Juin 1993.

[4]. A. Sayenko, T. Hamalainen, J. Joutsensalo et J. Siltanen, "An adaptive approach to WFQ with the revenue criterion", *Proceedings ISCC*, 2003.

[5]. Abbas Ibrahim, "Protocoles d'accès multiple orientés qualité de service en constellation de satellite à orbite basse", *Thèse, Télécom Paris [ENST]*, 2002.

[6]. Abhay K. Parekh et Robret G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case", *IEEE/ACM Transaction on Networking, Vol 1, Issue 3,* Juin 1993.

[7]. Ahsan Habib, Bharat Bhargava, "Network Tomography-based Unresponsive Flow Detection and Control". *Proceedings of the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems* (FTDCS'03)

[8]. B. Davie, A. Charny, J.C.R. Bennett, K. Benson, J.Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu et D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", *IETF RFC 3246*, mai 2002.

[9]. Brendan J. Frey, "Graphical Models for Machine Learning and Digital Communication", *MIT Press, Cambridge, Massachusetts*, 1998.

[10]. CAIDA: Cooperative Association for Internet Data Analysis. Disponible: http://www.caida.org/tools/.

[11]. Chin-Chang Li; Shiao-Li Tsao; Meng Cheng Chen; Yeali Sun; Yueh-Min Huang; "Proportional delay differentiation service based on weighted fair queuing" *Proceedings IEEE Computer Communications and Networks Conference,* Oct. 2000

[12]. D. Ferrari and D. Verma, "A scheme for real-time channel establishment in wide area networks," *IEEE J. Select. Areas Commun., vol. 8,* Avril 1990.

[13]. D. Grossman, "New Terminology and Clarifications for Diffserv", *IETF RFC 3260*, avril 2002.

[14]. D.C. Verma, H. Zhang, et D. Ferrari, "Delay jitter control for real-time communication in a packet switching network", *Proceedings de TRICOMM '91*.

[15]. D.P. Heyman, A. Tabatabai, T.V. Lakshman, "Statistical analysis and simulation study of video teleconference traffic in ATM networks", *IEEE Transactions on Circuits and Systems for Video Technology, Volume 2, Issue 1,* Mars 1992.

[16]. Dempster, Laird et Rubin , "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society, Series B, 39,* 1977.

[17]. E. Rosen, A. Viswanathan et R. Callon, "Multiprotocol Label Switching Architecture", *IETF RFC 3031*, janvier 2001.

[18]. E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. et Conta, "MPLS Label Stack Encoding", *IETF RFC 3032*, janvier 2001.

[19]. ETSI TR 101 112 V3.2.0 (1998-04) "Selection procedures for the choice of radio transmission technologies of the UMTS", *Technical Report Universal Mobile Telecommunications System (UMTS); ETSI (UMTS 30.03 version 3.2.0).*

[20]. ETSI, "Radio Equipment and Systems (RES); Trans-European Trucked Radio (TETRA); Voice + Data", *ETS 300 392, Part I. Edition 1.* 1996.

[21]. ETSI, Technical Specification, "Network architecture", *3G TS 23.002 V3.3.0 (2000-03)* (Release 1999).

[22]. ETSI, UMTS technical report, "Selection procedure for the choice of radio transmission technologies of the UMTS", *TR 101 112 V3.2.0* (1998).

[23]. F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval et J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services", *IETF RFC 3270*, mai 2002.

[24]. F. Natterer, "The Mathematics of Computerized Tomography", *New-york:Wiley, 1986*.

[25]. Gilad Koren, Dennis Shasha, "An optimal Scheduling Algorithm with Cometitive Factor for Real Time Systems", *rapport interne du département informatique de l'université de new-york*, Juillet 1991.

[26]. H. Aouad, A. Ibrahim, S. Tohmé, "UTRAN traffic parameters estimation", *IEEE CCNC 2004.*

[27]. H. Heffes, D. Lucantoni, "A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance", *IEEE Journal on Selected Areas in Communications, Volume 4, Issue 6,* Septembre 1986.

[28]. H. W. Lee, J.W. Mark, "ATM network traffic characterization using two types of on-off sources", *INFOCOM,* Avril 1993.

[29]. IEEE 8802.11-1999, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.

[30]. J. Cao, D. Davis, S. Vander Wiel et B. Yu, "Time-variying network tomography: Router link data", *J. Amer, Statist. Assoc, vol 95,* 2000.

[31]. J. Heinanen, F. Baker, W. Weiss et J. Wroclawski, "Assured Forwarding PHB Group", *IETF RFC 2597*, Juin 1999.

[32]. J.C.R. Bennett et Hui Zhang, "Hierarchical packet fair queueing algorithms", *IEEE/ACM Transactions on Networking , Vo 5, Issue 5,* Octobre 1997.

[33]. J.C.R. Bennett et H. Zhang, "WF²Q: Worst-case Fair Weighted Fair Queuing"**,** *Proceedings INFOCOM,* 1996.

[34]. K. Nichols, S. Blake, F. Baker et D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", *IETF RFC 2474*, December 1998.

[35]. L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks ", *ACM SIGCOMM Computer Communication Review, Volume 20 Issue 4*, Août 1990.

[36]. M. F. Neuts, "Structured Stochastic matrices of M/G/1 type and their applications", *Marcel Dekker, New York,* 1989.

[37]. M. Shreedhar et G. Varghese, "Efficient fair queuing using deficit round-robin", *IEEE/ACM Transactions on Networking, Vol 4, Issue 3,* Juin 1996.

[38]. M.F. Neuts, "Matrix geometric solutions in stochastic models: an algorithmic approach", *Baltimore, MD: john Hopkins Univ. press,* 1981.

[39]. Mark Coates, Alfred O. Hero III, Robert Nowak, and Bin Yu, "Internet tomography", *IEEE SIGNAL PROCESSING MAGAZINE*, Mai 2002.

[40]. Mark Coates, Robert Nowak, "Network Loss Inference Using Unicast end-to-end Measurement", *ITC Seminar on IP Traffic, Measurement and Modelling, Monterey, CA,* Sep 2000.

[41]. Mark Coates, Robert Nowak, "Network Tomography for internal delay estimation", *proc. IEEE Int. conf. Acoust., Speech, Signal Process.,* salt lake city, UT, Mai 2001.

[42]. Multicast-based inference of network-internal characteristics (MINC), Disponible: http://gaia.cs.umass.edu/minc.

[43]. N. D. Wilson , R. Ganesh, K. Joseph, D. Raychaudhuri, "Packet CDMA versus Dynamic TDMA for Multiple Access on an Integrated Voice/Data PCN", *IEEE Journal on Selected Areas in Communications, Volume 11,* August 1993.

[44]. N.G. Duffield, J. Horowitz, F. Lo Presti et D. Towsley, "Multicast topology inference from end-to-end measurements", *ITC Seminar on IP Traffic, Measurement and Modelling, Monterey, CA,* Sep 2000.

[45]. N.G. Duffield, J. Horowitz, F. Lo Presti et D. Towsley, "Multicast topology inference from measured end-to-end loss", *IEEE Trans. Inform. Theory, vol 48*, Janvier 2002.

[46]. P. O'Sullivan, "A Statistical Perspective on Ill-posed Inverse Problems", *Statistical Science, vol.1, no. 4, 1986*.

[47]. R. Caceres, N. Duffield, J. Horowitz et D. Towsley, "Multicast-based inference of network-internal loss characteristics", *IEEE Trans. Inform. Theory, vol. 45,* Nov 1999.

[48]. R.J. Mammone, "Inverse Problems and Signal Processing", *The digital signal processing handbook. CRC press, 1998.*

[49]. RTP: A Transport Protocol for Real-Time Applications, janvier 1996. *IETF RFC 1889*.

[50]. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang et W. Weiss, "An Architecture for Differentiated Services", *IETF RFC 2475*, December 1998.

[51]. S. H. Kang, D. K. Sung, "Two-state MMPP modelling of ATM superposed traffic streams based on the characterization of correlated interarrival times", *Global Telecommunications Conference,* 1995.

[52]. S. Keshav, "An Engineering Approach to Computer Networking", Reading, MA: Addison-Wesley, 1997.

[53]. S. Ratnassamy et S. McCanne, "Inference of multicast routing trees ans bottleneck bandwidths using end-to-end measurements", *Proc. IEEE INFOCOM 1999*.

[54]. S.J. Golestani, "A self-clocked fair queueing scheme for broadband applications", *Proceedings de Infocom 94*.

[55]. Tobias Rydén, "An EM algorithm for estimation in Markov-modulated Poisson processes", *Computational Statistics & Data Analysis Volume 21, Issue 4*, 1996

[56]. Tobias RYDÉN, "Parameter Estimation For Markov Modulated Poisson Processes", *Stocastic Models*, 1994.

[57]. Wilson, Ganesh, Joseph, Raychaudhuri, "Performance of cellular Packet CDMA in an Integrated Voice/Data Network", *International Journal Wireless Information Networks, Volume 1,* 1994.

[58]. Wolfgan Fischer, Kathleen Meir-Hellstern, "The Markov-modulated Poisson Process (MMPP) Coolbook", *Performance Evaluation 18*, 1992.