

Titre: WIRELESS-DIFFSERV* : Modèle de protection différenciée contre les dégradations de canal dans les réseaux maillés IEEE 802.11
Title: degradations de canal dans les réseaux maillés IEEE 802.11

Auteur: Marnie Vodounou
Author:

Date: 2012

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Vodounou, M. (2012). WIRELESS-DIFFSERV* : Modèle de protection différenciée contre les dégradations de canal dans les réseaux maillés IEEE 802.11 [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/1009/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/1009/>
PolyPublie URL:

Directeurs de recherche: Brunilde Sanso, & André Girard
Advisors:

Programme: Génie Électrique
Program:

UNIVERSITÉ DE MONTRÉAL

WIRELESS-DIFFSERV* : MODÈLE DE PROTECTION DIFFÉRENCIÉE CONTRE
LES DÉGRADATIONS DE CANAL DANS LES RÉSEAUX MAILLÉS IEEE 802.11

MARNIE VODOUNOU
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
DÉCEMBRE 2012

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

WIRELESS-DIFFSERV* : MODÈLE DE PROTECTION DIFFÉRENCIÉE CONTRE
LES DÉGRADATIONS DE CANAL DANS LES RÉSEAUX MAILLÉS IEEE 802.11

présenté par : VODOOUNOU Marnie

en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

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

M. FRIGON Jean-François, Ph.D., président

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

M. GIRARD André, Ph.D., membre et codirecteur de recherche

M. CARDINAL Christian, Ph.D., membre

À ma famille

REMERCIEMENTS

J'aimerais exprimer ma profonde gratitude et mes sincères remerciements à l'endroit de mes directeurs de recherche Mme Brunilde Sansò et M. André Girard pour avoir accepté d'en-cadrer mon travail de recherche, pour la confiance accordée, pour les conseils prodigues et pour leur disponibilité. Je les remercie particulièrement pour le soutien matériel et financier et pour les commentaires pertinents dans ce projet de recherche.

Je remercie également M. Jean-François Frigon et M. Christian Cardinal d'avoir accepté de prendre part au jury de ce mémoire de maîtrise.

Je tiens également à exprimer ma sincère gratitude et ma profonde reconnaissance à ma famille, qui m'a été d'un réel soutien durant cette aventure. Je remercie aussi Frédérique et Astrid, qui m'ont tout le temps encouragée et incitée à ne pas baisser les bras pendant les moments difficiles, sans oublier Roméo, Japhet et Eric pour le soutien moral.

Finalement, je tiens à remercier mes amis et collègues de travail du GERAD pour la bonne ambiance, Hadhami, Sami, Silvia, Luca et tout particulièrement Federico et Arash, pour leurs précieux conseils et leur soutien inconditionnel. Mes remerciements vont aussi à l'endroit des autres membres du GERAD en particulier Marie, Valérie, Carole et Marilyne du service administratif, Pierre, Daniel, Luc et Omar du service technique. Merci à tous d'avoir rendu cet environnement de travail plus agréable.

RÉSUMÉ

L'utilisation massive des applications multimédia et temps réel sur le réseau Internet est une préoccupation très importante car elles sont très gourmandes en bande passante et imposent des exigences de qualité de service très sévères. De plus, le besoin de mobilité des utilisateurs associé aux faibles coûts de déploiement des technologies sans fil sont quelques unes des raisons pour lesquelles les réseaux maillés sans fil vont pouvoir émerger comme une technologie permettant de fournir l'Internet sans fil à haut débit et capable de supporter les applications en temps réel telles que la voix sur IP (VoIP) et la vidéo-conférence. Cependant, la nature variable du canal sans fil et les interférences radio créent une entrave au déploiement d'un backbone entièrement sans fil. En effet, ces phénomènes inhérents aux réseaux sans fil dégradent les performances et sont un véritable frein à l'exploitation du plein potentiel de ces technologies. L'objectif de ce mémoire est d'étudier un mécanisme d'auto-fonctionnement permettant de garantir une qualité de service différenciée dans les réseaux maillés sans fil. Nous avons mis l'accent sur la nécessité d'utiliser une politique de service dynamique qui s'adapte aux variations du canal sans fil. Ainsi, il est important de modifier la politique d'ordonnancement des paquets dans un environnement sans fil variable afin de satisfaire en tout temps aux exigences en qualité de service des trafics prioritaires. La méthode proposée consiste à mettre en œuvre, dans les réseaux sans fil, le modèle *DiffServ**, préalablement conçu pour les réseaux IP optiques. Cette variante est appelée *Wireless-DiffServ**. Elle repose sur l'utilisation conjointe de l'architecture des services différenciés (DiffServ) dans la couche IP d'un réseau et de l'utilisation de canaux disjoints dans sa couche physique. Dans cette étude, le média physique est un canal de Rayleigh, plat en fréquence et à évanouissement lent, modélisé par des chaînes de Markov discrètes à états finis. Un procédé d'adaptation de la modulation et du codage y a ensuite été mis en place. La pondération cyclique ou *Weighted Round Robin* (WRR) est utilisée comme politique d'ordonnancement dans l'architecture des services différenciés (DiffServ). Par la suite, le modèle *Wireless-DiffServ** a été implémenté dans le simulateur de réseau "Network Simulator NS-2" et ses performances ont été comparées avec celles du standard DiffServ puis celles de la politique Premier Arrivé, Premier servi (FIFO). Les résultats montrent que *Wireless-DiffServ** permet *en moyenne* de garantir une meilleure protection que les deux autres politiques lorsque le réseau n'est pas saturé. De plus, *Wireless-DiffServ** diminue la gigue moyenne des paquets, quel que soit le volume de trafic écoulé. Lorsque le réseau est saturé, *Wireless-DiffServ** enregistre environ 1% moins de perte que DiffServ et une augmentation du délai d'à peu près 10 ms. La politique FIFO n'offre pas de différenciation de services mais réduit considérablement les pertes du trafic VoIP. Finale-

ment, le modèle *Wireless-DiffServ** est utilisé pour la mise en place d'un contrôle de qualité. La qualité du canal sans fil est caractérisée par son état d'évanouissement et son rapport signal sur bruit (SNR) moyen. Trois niveaux de qualité correspondant à des ordonnancements différents sont utilisés. Les résultats montrent que l'adaptation des poids du WRR en fonction du SNR moyen donne une autonomie de fonctionnement au réseau et permet de garantir au moins 50% de la bande passante au trafic VoIP. Ce procédé dégrade tout de même les performances et surtout la gigue des autres trafics.

ABSTRACT

Wireless mesh networks (WMNs) are emerging as a new network architecture able to provide broadband Internet and to support higher amount of real-time traffic such as Voice over IP (VoIP) and videoconferencing. Real-time applications require a lot of bandwidth and impose stringent quality of service (QoS) criteria. However, the time-varying fading channel and interference phenomena impair wireless network's performance and do not exploit the full capacity of WMNs or to satisfy the QoS requirements of these applications.

The aim of this research project is to study a self-functioning mechanism to guarantee a given level of differentiated QoS in WMNs. We study the performance of a dynamic packet scheduling algorithm in wireless environment. It is based on Weighted Round Robin (WRR) and adjusts the weights of different classes of traffic (VoIP, video, data) according to the channel quality in order to always protect VoIP traffic from channel failures. To achieve this goal, we implement in "Network Simulator NS-2" a cross-layer design called *Wireless-DiffServ** based on the interaction between the IP, MAC and Physical layers. This model uses IP differentiated services (DiffServ) combined with disjoint multipaths in the Physical layer to provide reliability in WMNs. In this project, the Rayleigh wireless channel, supposed flat and slow fading, is modeled by a finite state Markov chain joined with adaptive modulation and coding (AMC).

*Wireless-DiffServ** performance is compared with a pure DiffServ and First In First Out (FIFO) schemes. Simulation results shows that *Wireless-DiffServ** reduces network jitter, and when the network is not saturated, it is *on average* better than DiffServ and FIFO.

Finally, we add a failure detection mechanism in *Wireless-DiffServ**. Failures in wireless channel are characterized by channel fading and average signal to noise ratio (SNR). We defined three levels of failure. Simulation results show that by varying WRR weights based on the failure level, we allow the self-operation of the network and can guarantee the assignment of at least 50% of the bandwidth to VoIP traffic.

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xiii
LISTE DES ANNEXES	xviii
LISTE DES SIGLES ET ABRÉVIATIONS	xix
INTRODUCTION	1
CHAPITRE 1 LES RÉSEAUX MAILLÉS ET LE STANDARD IEEE 802.11	8
1.1 Les réseaux maillés sans fil	8
1.1.1 Architectures des réseaux maillés sans fil	9
1.1.2 Caractéristiques et problématique des RMSF	10
1.2 Le standard IEEE 802.11	11
1.2.1 Topologie	11
1.2.2 Couche MAC	12
1.2.2.1 Protocoles d'accès au médium	12
1.2.2.2 Trames MAC	13
1.2.3 Couche physique	14
1.3 Performances des réseaux 802.11	16
1.3.1 Temps de transmission d'un paquet	16
1.3.1.1 Temps de transmission d'un paquet sans erreurs	16
1.3.1.2 Temps de transmission avec erreurs	19
1.3.2 Mécanismes de QoS dans les réseaux 802.11	20
1.3.2.1 Différenciation de services	20

1.3.2.2	Contrôle d'admission et réservation de bande passante	20
1.3.2.3	Adaptation de lien	21
CHAPITRE 2 DIFFÉRENCIATION DE SERVICES IP ET MODÈLE WIRELESS-DIFFSERV*		22
2.1	Architecture DiffServ	22
2.1.1	Fonctionnement de DiffServ	23
2.1.2	Politiques d'ordonnancement	24
2.2	Modèle DiffServ*	25
2.2.1	DiffServ* dans les réseaux IP/WDM	25
2.2.2	DiffServ* dans les réseaux 802.11	26
2.3	Modélisation du canal sans fil	27
2.3.1	Description du canal sans fil	27
2.3.2	Paramètres physiques	28
2.3.3	Caractéristiques du Canal de Rayleigh	29
2.3.4	Adaptation de lien	30
2.3.4.1	Principe de l'AMC	30
2.3.4.2	Partitionnement du SNR	31
2.4	Sélection de l'interface de sortie	32
CHAPITRE 3 IMPLÉMENTATIONS ET PERFORMANCES DE DIFFSERV ET DE L'AMC		34
3.1	Modèles de trafic et métriques de performance	34
3.2	Topologie étudiée	35
3.3	Vérification de l'implémentation de DiffServ	36
3.3.1	Règle d'attribution des poids	36
3.3.2	Étude des systèmes 20-2-1 et 20-3-1	38
3.3.3	Étude du système 8-4-1	42
3.4	Implémentation du module d'AMC dans un canal de Rayleigh	44
3.4.1	Validation d'un canal à 2 états	44
3.4.2	Effet d'un canal à plusieurs états	45
3.4.3	Impact des paramètres physiques	49
3.4.3.1	Incidence du SNR moyen	49
3.4.3.2	Incidence du taux d'erreur par trame	50
3.4.3.3	Impact de la fréquence Doppler	54

CHAPITRE 4 PERFORMANCES DE WIRELESS-DIFFSERV*	56
4.1 Temps d'échantillonnage	56
4.2 Performance sur un lien	57
4.2.1 Classe EF – Trafic VoIP	59
4.2.1.1 Mesure des pertes	60
4.2.1.2 Mesure du délai moyen	60
4.2.1.3 Mesure de la gigue	62
4.2.2 Classe AF – Trafic Vidéo	62
4.2.2.1 Mesure des pertes	62
4.2.2.2 Mesure du délai moyen	64
4.2.2.3 Mesure de la gigue	64
4.2.3 Classe BE – Trafic Données	66
4.2.3.1 Mesure des pertes	66
4.2.3.2 Mesure du délai moyen	66
4.2.3.3 Mesure de la gigue	68
4.2.4 Étude du compromis entre DiffServ* et FIFO	68
4.3 Performance dans un réseau	72
4.3.1 Classe EF – Trafic VoIP	74
4.3.1.1 Mesure des pertes	74
4.3.1.2 Mesure du délai moyen	75
4.3.1.3 Mesure de la gigue	76
4.3.2 Classe AF – Trafic Vidéo	77
4.3.2.1 Mesure des pertes	77
4.3.2.2 Mesure du délai moyen	78
4.3.2.3 Mesure de la gigue	79
4.3.3 Classe BE – Trafic Données	79
4.3.3.1 Mesure des pertes	79
4.3.3.2 Mesure du délai moyen	81
4.3.3.3 Mesure de la gigue	81
4.3.4 Étude du compromis entre DiffServ* et FIFO	83
4.4 Sommaire	85
CHAPITRE 5 WIRELESS-DIFFSERV* ET ADAPTATION DE POIDS	86
5.1 Contrôle de qualité du canal	87
5.2 Répartition de la bande passante	89
5.3 Performance sur un lien	94

5.3.1	Comparaison DiffServ vs DiffServ*	94
5.3.1.1	Classe EF – Trafic VoIP	95
5.3.1.2	Classe AF – Trafic Vidéo	95
5.3.1.3	Classe BE – Trafic Données	99
5.3.2	Comparaison poids statiques et poids variables	101
5.3.2.1	Classe EF – Trafic VoIP	101
5.3.2.2	Classe AF – Trafic Vidéo	101
5.3.2.3	Classe BE – Trafic Données	105
5.3.3	Utilisation dynamique de la bande passante	108
5.4	Performance dans un réseau	110
5.4.1	Comparaison DiffServ vs DiffServ*	110
5.4.1.1	Classe EF – Trafic VoIP	113
5.4.1.2	Classe AF – Trafic Vidéo	115
5.4.1.3	Classe BE – Trafic Données	115
5.4.2	Comparaison poids statiques et poids variables	120
5.4.2.1	Classe EF – Trafic VoIP	120
5.4.2.2	Classe AF – Trafic Vidéo	120
5.4.2.3	Classe BE – Trafic Données	123
5.4.3	Utilisation dynamique de la bande passante	126
5.5	Sommaire	127
	CONCLUSION	129
	RÉFÉRENCES	133
	ANNEXES	138
A.1	Modèle de noeud dans NS	138
A.2	Modifications apportées au simulateur	139
A.3	Cheminement des paquets	139
A.3.1	Simulation DiffServ*	139
A.3.2	Simulation DiffServ et FIFO	140

LISTE DES TABLEAUX

Tableau 1.1	Principales normes physiques de la famille 802.11	14
Tableau 1.2	Taux de transmission du Standard IEEE 802.11g	15
Tableau 1.3	Paramètres des couches MAC et PHY	15
Tableau 1.4	Temps d’overhead par paquet en fonction du débit de transmission . .	17
Tableau 1.5	Nombre maximal de sources par classe	18
Tableau 2.1	Correspondance entre états de la chaîne de Markov et taux de transmission	32
Tableau 3.1	Paramètres des 3 classes de trafic.	34
Tableau 3.2	Taille moyenne des paquets (avec et sans en-têtes) par classe	37
Tableau 3.3	Probabilité d’état et débit moyen	45
Tableau 5.1	Niveaux de qualité et répartition de la bande passante	88
Tableau 5.2	Sommaire de la répartition de la bande passante sur un lien “très bon”	91
Tableau 5.3	Sommaire de la répartition de la bande passante sur un “bon” lien . .	92
Tableau 5.4	Mesure de la répartition de la bande passante sur un “mauvais” lien .	93
Tableau 5.5	États des canaux du noeud S par destination et par interface	111
Tableau 5.6	Implémentation des poids variables dans DiffServ et DiffServ*	113

LISTE DES FIGURES

Figure 1.1	Architecture d'un réseau maillé sans fil de type infrastructure	9
Figure 1.2	Architecture d'un réseau maillé sans fil de type client	10
Figure 1.3	Fonctionnement du mécanisme DCF	13
Figure 1.4	Décomposition du temps de transmission d'un paquet	16
Figure 2.1	Fonctionnement standard de DiffServ dans un réseau IP	23
Figure 2.2	Protection DiffServ* dans un noeud mesh	27
Figure 2.3	BER en fonction du SNR pour différentes techniques de modulation .	31
Figure 2.4	Chaîne de Markov à états finis modélisant les conditions du canal . .	32
Figure 3.1	Topologie étudiée	36
Figure 3.2	Déploiement standard de DiffServ dans un noeud sans fil	36
Figure 3.3	Taux de perte dans avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$	39
Figure 3.4	Délai moyen avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$. . .	39
Figure 3.5	Gigue moyenne avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$. .	40
Figure 3.6	Taux de perte avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. .	40
Figure 3.7	Délai moyen avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. . .	41
Figure 3.8	Gigue moyenne avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. .	41
Figure 3.9	Perte moyenne avec la pondération $\alpha_{EF} = 8$, $\alpha_{AF} = 4$ et $\alpha_{BE} = 1$. .	43
Figure 3.10	Délai moyen avec la pondération $\alpha_{EF} = 8$, $\alpha_{AF} = 4$ et $\alpha_{BE} = 1$. . .	43
Figure 3.11	Variation des pertes avec DiffServ dans un canal à 2 états	46
Figure 3.12	Variation du délai avec DiffServ dans un canal à 2 états	46
Figure 3.13	Variation de la gigue avec DiffServ dans un canal à 2 états	47
Figure 3.14	Taux de paquets perdus avec DiffServ dans un canal à 7 états	48
Figure 3.15	Délai moyen avec DiffServ dans un canal à 7 états	48
Figure 3.16	Gigue moyenne avec DiffServ dans un canal à 7 états	49
Figure 3.17	Débit moyen par canal en fonction du SNR moyen.	50
Figure 3.18	Perte moyenne en fonction du SNR moyen	51
Figure 3.19	Délai moyen en fonction du SNR moyen	51
Figure 3.20	Gigue moyenne en fonction du SNR moyen.	52
Figure 3.21	Perte moyenne en fonction du PER	53
Figure 3.22	Délai moyen en fonction du PER	53
Figure 3.23	Gigue moyenne en fonction du PER	54
Figure 4.1	Lien logique et liens physiques.	58
Figure 4.2	Protection DiffServ Standard dans un noeud mesh	58

Figure 4.3	Files d'attente FIFO dans un nœud mesh	59
Figure 4.4	Réseau de 2 nœuds : Pertes du trafic VoIP en fonction de la politique de service.	61
Figure 4.5	Réseau de 2 nœuds : Délai moyen du trafic VoIP en fonction de la politique de service.	61
Figure 4.6	Réseau de 2 nœuds : Gigue du trafic VoIP en fonction de la politique de service.	63
Figure 4.7	Réseau de 2 nœuds : Pertes du trafic vidéo en fonction de la politique de service.	63
Figure 4.8	Réseau de 2 nœuds : Délai moyen du trafic vidéo en fonction de la politique de service.	65
Figure 4.9	Réseau de 2 nœuds : Gigue du trafic vidéo en fonction de la politique de service.	65
Figure 4.10	Réseau de 2 nœuds : Pertes du trafic données en fonction de la politique de service.	67
Figure 4.11	Réseau de 2 nœuds : Délai moyen du trafic données en fonction de la politique de service.	67
Figure 4.12	Réseau de 2 nœuds : Gigue du trafic données en fonction de la politique de service.	69
Figure 4.13	DiffServ* vs FIFO : Perte moyenne par classe dans un réseau de 2 nœuds	69
Figure 4.14	DiffServ* vs FIFO : Délai moyen par classe dans un réseau de 2 nœuds	71
Figure 4.15	DiffServ* vs FIFO : Gigue moyenne par classe dans un réseau de 2 nœuds	71
Figure 4.16	Réseau sans fil à 5 nœuds	73
Figure 4.17	Réseau de 5 nœuds : Pertes du trafic VoIP en fonction de la politique de service.	75
Figure 4.18	Réseau de 5 nœuds : Délai moyen du trafic VoIP en fonction de la politique de service.	76
Figure 4.19	Réseau de 5 nœuds : Gigue du trafic VoIP en fonction de la politique de service.	77
Figure 4.20	Réseau de 5 nœuds : Pertes du trafic vidéo en fonction de la politique de service.	78
Figure 4.21	Réseau de 5 nœuds : Délai moyen du trafic vidéo en fonction de la politique de service.	79
Figure 4.22	Réseau de 5 nœuds : Gigue du trafic vidéo en fonction de la politique de service.	80

Figure 4.23	Réseau de 5 noeuds : Pertes du trafic données en fonction de la politique de service.	80
Figure 4.24	Réseau de 5 noeuds : Délai moyen du trafic données en fonction de la politique de service.	82
Figure 4.25	Réseau de 5 noeuds : Gigue du trafic données en fonction de la politique de service.	82
Figure 4.26	DiffServ* vs FIFO : Perte moyenne dans un réseau de 5 noeuds	83
Figure 4.27	DiffServ* vs FIFO : Délai moyen dans un réseau de 5 noeuds	84
Figure 4.28	DiffServ* vs FIFO : Gigue moyenne dans un réseau de 5 noeuds	84
Figure 5.1	Lien logique entre deux noeuds	90
Figure 5.2	Répartition de la bande passante sur un lien “très bon”	91
Figure 5.3	Répartition de la bande passante sur un lien “bon”	92
Figure 5.4	Répartition de la bande passante sur un “mauvais” lien	93
Figure 5.5	Perte du trafic VoIP sur un lien logique avec poids variables : DiffServ vs DiffServ*	96
Figure 5.6	Délai moyen du trafic VoIP sur un lien logique avec poids variables : DiffServ vs DiffServ*	96
Figure 5.7	Gigue du trafic VoIP sur un lien logique avec poids variables : DiffServ vs DiffServ*	97
Figure 5.8	Perte du trafic vidéo sur un lien logique avec poids variables : DiffServ vs DiffServ*	97
Figure 5.9	Délai moyen du trafic vidéo sur un lien logique avec poids variables : DiffServ vs DiffServ*	98
Figure 5.10	Gigue du trafic vidéo sur un lien logique avec poids variables : DiffServ vs DiffServ*	98
Figure 5.11	Perte du trafic données sur un lien logique avec poids variables : DiffServ vs DiffServ*	99
Figure 5.12	Délai moyen du trafic données sur un lien logique avec poids variables : DiffServ vs DiffServ*	100
Figure 5.13	Gigue du trafic données sur un lien logique avec poids variables : DiffServ vs DiffServ*	100
Figure 5.14	Perte du trafic VoIP sur un lien logique : Poids fixes 20-3-1 vs Poids variables	102
Figure 5.15	Délai moyen du trafic VoIP sur un lien logique : Poids fixes 20-3-1 vs Poids variables	102

Figure 5.16	Gigue du trafic VoIP sur un lien logique : Poids fixes 20-3-1 vs Poids variables	103
Figure 5.17	Perte du trafic vidéo sur un lien logique : Poids fixes 20-3-1 vs Poids variables	103
Figure 5.18	Délai moyen du trafic vidéo sur un lien logique : Poids fixes 20-3-1 vs Poids variables	104
Figure 5.19	Gigue du trafic vidéo sur un lien logique : Poids fixes 20-3-1 vs Poids variables	105
Figure 5.20	Perte du trafic données sur un lien logique : Poids fixes 20-3-1 vs Poids variables	106
Figure 5.21	Délai moyen du trafic données sur un lien logique : Poids fixes 20-3-1 vs Poids variables	107
Figure 5.22	Gigue du trafic données sur un lien logique : Poids fixes 20-3-1 vs Poids variables	107
Figure 5.23	Pondération variable vs statique 20-3-1 : capacité relative sur un lien logique	109
Figure 5.24	Réseau maillé de 5 noeuds	111
Figure 5.25	Perte du trafic VoIP dans un réseau avec poids variables : DiffServ vs DiffServ*	114
Figure 5.26	Délai moyen du trafic VoIP dans un réseau avec poids variables : DiffServ vs DiffServ*	114
Figure 5.27	Gigue du trafic VoIP dans un réseau avec poids variables : DiffServ vs DiffServ*	116
Figure 5.28	Perte du trafic vidéo dans un réseau avec poids variables : DiffServ vs DiffServ*	116
Figure 5.29	Délai moyen du trafic vidéo dans un réseau avec poids variables : DiffServ vs DiffServ*	117
Figure 5.30	Gigue du trafic vidéo dans un réseau avec poids variables : DiffServ vs DiffServ*	117
Figure 5.31	Perte du trafic données dans un réseau avec poids variables : DiffServ vs DiffServ*	118
Figure 5.32	Délai moyen du trafic données dans un réseau avec poids variables : DiffServ vs DiffServ*	119
Figure 5.33	Gigue du trafic données dans un réseau avec poids variables : DiffServ vs DiffServ*	119
Figure 5.34	Perte du trafic VoIP dans un réseau : Poids fixes 20-3-1 vs Poids variables	121

Figure 5.35	Délai moyen du trafic VoIP dans un réseau : Poids fixes 20-3-1 vs Poids variables	121
Figure 5.36	Gigue du trafic VoIP dans un réseau : Poids fixes 20-3-1 vs Poids variables	122
Figure 5.37	Perte du trafic vidéo dans un réseau : Poids fixes 20-3-1 vs Poids variables	122
Figure 5.38	Délai moyen du trafic vidéo dans un réseau : Poids fixes 20-3-1 vs Poids variables	124
Figure 5.39	Gigue du trafic vidéo dans un réseau : Poids fixes 20-3-1 vs Poids variables	124
Figure 5.40	Perte du trafic données dans un réseau : Poids fixes 20-3-1 vs Poids variables	125
Figure 5.41	Délai moyen du trafic données dans un réseau : Poids fixes 20-3-1 vs Poids variables	125
Figure 5.42	Gigue du trafic données dans un réseau : Poids fixes 20-3-1 vs Poids variables	126
Figure 5.43	Pondération variable vs statique 20-3-1 : capacité relative dans un réseau	127
Figure A.1	Cheminement d'un paquet avec DiffServ*	140
Figure A.2	Cheminement d'un paquet avec DiffServ	141
Figure A.3	Cheminement d'un paquet avec FIFO	142

LISTE DES ANNEXES

Annexe A	DESCRIPTION DE L'IMPLÉMENTATION	138
----------	---	-----

LISTE DES SIGLES ET ABRÉVIATIONS

AMC	Adaptation de la Modulation et du Codage
CAFMA-SD	Channel Assignment and Fast MAC Architecture for Service Differentiation
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contention Windows
DCF	Distributed Coordination Function
DIFS	Distributed Inter-Frame Space
DiffServ (DS)	Services Différenciés (Differentiated Services)
DSSS	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
EIFS	Extended Inter-Frame Space
FIFO	Premier Arrivé Premier Servi (First in First Out)
GSM	Global System for Mobile communications
HWMP	Hybrid Wireless Mesh Protocol
IFS	Inter-Frame Space
IP	Internet Protocol
MAC	Medium Access Control
MPOLSR	Multi-Path Optimized Link State Routing
OFDM	Orthogonal Frequency Division Multiplexing
OLSR	Optimized Link State Routing
PCF	Point Coordination Function
PER	Packet Error Rate
PMD	Physical Medium Dependent
PHY	PHYSical Layer
PLCP	Physical Layer Convergence Protocol
QoS	Qualité de Service
RMSF	Réseau Maillé Sans Fil
RR	Round Robin
SIFS	Short Inter-Frame Space
SNR	Signal to Noise Ratio
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voix sur IP (Voice over IP)

WLAN	Wireless Local Area Network
WMN	Wireless Mesh Network
WRR	Weighted Round Robin
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
Wireless-DiffServ* (DS*)	Services différenciés contre les dégradations du canal sans fil

INTRODUCTION

Depuis sa mise en fonction, le réseau Internet a connu une évolution tellement rapide qu'il est aujourd'hui le réseau de communication le plus utilisé dans le monde. Il offre une multitude de service dont les plus courants sont entre autres la téléphonie sur IP, la vidéo sur demande et le transfert de données. De plus, son utilisation s'est fortement accrue d'une part, depuis l'avènement des réseaux sociaux, qui permettent à des groupes de personnes d'échanger des informations d'ordre personnelle ou professionnelle et d'autre part depuis la venue des dispositifs sans fil intelligents comme le téléphone, la tablette, le lecteur de livres numériques, etc, et qui procurent également des interfaces permettant d'accéder à Internet. En outre, l'émergence du *Cloud Computing* ou "Informatique dans le Nuage" est une transformation importante dans l'industrie de l'Internet. En effet, grâce à ce nouveau concept, les utilisateurs d'Internet disposent d'espace de stockage sur des serveurs distants où ils peuvent enregistrer diverses informations : carnets d'adresses, documents, musique, etc. Grâce à cette nouvelle utilisation de l'Internet, tous leurs documents leur seront accessibles en temps réel et de n'importe quel endroit pour peu qu'ils disposent d'une connexion à Internet. Cette connexion est d'autant plus simplifiée par l'utilisation de réseaux sans fil. Ainsi, l'utilisation massive des systèmes de communication sans fil tels que les réseaux de téléphonie cellulaire 3G ou les réseaux locaux sans fil (WLANs) facilite grandement la mobilité des utilisateurs tout en leur offrant l'accès à tous les services du *Cloud*.

Cette prédominance des réseaux sans fil est à la base du développement de l'architecture des réseaux maillés sans fil (RMSF). Les réseaux maillés sans fil constituent, à ce jour, l'alternative la moins coûteuse par rapport à l'installation de nouveaux réseaux filaires ou au redimensionnement des réseaux filaires existants. En effet, avec l'apparition du *Cloud Computing* et vu la forte utilisation des applications de voix et de vidéo sur Internet, il est nécessaire d'augmenter la capacité de transmission des réseaux existants et de prévoir des débits de transmission très élevés pour les applications futures. Les réseaux maillés sans fil s'inscrivent alors dans cette optique car ils peuvent couvrir des zones très étendues et supporter des quantités d'informations considérables. Ce sont donc des réseaux à large bande utilisés pour le backhaul des données entre utilisateurs et Internet. À cette fin, ils peuvent être déployés avec des technologies telles que le 802.11n qui peut offrir un débit allant jusqu'à 300 Mbps sur une portée de 100 m ou 250 m, ou le 802.16, qui offre un débit de l'ordre de 75 Mbps sur une portée d'environ 10 km voir plus.

Cependant, le déploiement de cette architecture doit faire face aux problèmes inhérents à tous les réseaux sans fil : les interférences entre les liaisons radio et les évanouissements

dus à l'instabilité des liens sans fil. En effet, puisque la communication dans les réseaux sans fil repose sur la propagation des ondes électromagnétiques, des phénomènes d'interférence se produisent lorsque plusieurs équipements voisins communiquent simultanément en utilisant la même fréquence radio. Les interférences sont principalement dues à la nature multi-sauts des réseaux maillés sans fil. D'autre part, l'instabilité des liaisons sans fil est due aux conditions du milieu ambiant telles que les conditions météorologiques, la mobilité et la distance entre les antennes, la hauteur et le nombre d'obstacles. Ainsi, ces deux problèmes ont pour conséquence de dégrader les performances des réseaux sans fil, c'est-à-dire augmenter le délai et les erreurs de transmission, diminuer la capacité de transmission et augmenter le taux de perte. Il apparaît alors important de trouver des méthodes pour limiter les dégradations observées et garantir aux utilisateurs une bonne qualité de service quelle que soit la nature du problème qui affecte le fonctionnement du réseau.

Étant donné que les services offerts sur Internet ont des exigences de qualité de service différentes, il est nécessaire que les réseaux maillés sans fil puissent faire aussi cette distinction et garantir différents niveaux de qualité de service en tout temps. Les réseaux sans fil de type 802.11 ne permettent pas de faire cette différenciation de qualité de service à cause des caractéristiques de leur couche MAC, qui fournit uniquement des services de *meilleur effort*. En effet, cette dernière étant incapable d'identifier le trafic des couches supérieures, toutes les applications concourent pour l'accès au même canal. Par ailleurs, le fonctionnement de la couche MAC dans ces réseaux est tel qu'il ajoute des temps d'attente supplémentaires pour chaque paquet en vue d'éviter les collisions. Ces temps d'attente peuvent également s'avérer très néfastes pour les applications de voix et de vidéo sur IP. Toutefois, la différenciation reste possible, soit en modifiant le fonctionnement de la couche MAC, soit en utilisant des mécanismes offerts par d'autres couches du modèle TCP/IP. Les deux options sont également possibles. Dans ce cas, il s'agit des modèles inter-couche ou *cross-layer*.

L'utilisation des mécanismes de la couche IP est au centre de nombreuses études réalisées au cours des dix dernières années. La couche IP propose deux mécanismes pour fournir de la qualité de service différenciée dans un réseau : IntServ et DiffServ. IntServ s'appuie sur le principe de réservation de ressource. Ainsi, pour chaque flot, IntServ réserve de bout en bout la largeur de bande requise pour satisfaire les exigences de qualité de service de ce flot. Son inconvénient principal est qu'il n'est pas adapté aux réseaux de grande taille. DiffServ regroupe tous les flots ayant les mêmes besoins en qualité de service dans une classe de service. Chaque classe de service possède une priorité en fonction de laquelle elle reçoit un traitement différent par noeud. DiffServ est donc le mécanisme le plus adéquat dans les réseaux IP de grande taille. Cependant, IntServ et DiffServ ont pendant longtemps été déployés uniquement dans les réseaux filaires.

Jiang *et al.* [23] sont les premiers auteurs à avoir proposé un modèle inter-couche permettant d'intégrer le fonctionnement de DiffServ dans le backbone des réseaux maillés sans fil 802.11. Dans la couche IP, la métrique de routage est basée sur la qualité de service : tous les flots appartenants à une classe de service sont acheminés par le même chemin jusqu'à ce que ce chemin n'ait plus suffisamment de ressources pour accepter un nouveau flot. Un contrôle d'admission est donc mis en place par la passerelle de destination, afin de déterminer la bande passante disponible et savoir si un nouveau flot de données peut être accepté sur ce chemin tout en respectant ses besoins de qualité de service. Enfin, ces auteurs proposent également que le mécanisme d'accès de la couche MAC basé sur le DCF soit remplacé par une variante du mécanisme EDCA proposé par la norme IEEE 802.11e. Proposé pour les réseaux locaux sans fil 802.11, EDCA est une technique de différenciation de service qui modifie les paramètres d'attente de la couche MAC en fonction des différentes classes de service. Cependant les mécanismes proposés par le standard IEEE 802.11e ne garantissent pas la qualité de service des connexions en cours lorsque le réseau devient fortement chargé ni celle des applications en temps réel lorsque plusieurs équipements séparent la source et la destination. Bien que le modèle proposé n'ait pas été implémenté, il met en relief les principaux aspects à considérer pour fournir de la qualité de service dans les réseaux maillés sans fil basés sur le standard IEEE 802.11.

Riggio *et al.* [38] ont, quant à eux, déployés une architecture qui ne nécessite aucune modification de la couche MAC mais s'appuie seulement sur l'utilisation de DiffServ et d'un algorithme d'agrégation de paquets pour éviter les effets de l'*overhead* produit normalement par la couche MAC sur les paquets de petite taille. Les résultats de leurs tests montrent que l'agrégation de paquets permet d'augmenter le nombre de connexions de voix admises sur un chemin sans en dégrader les performances lorsque le nombre de paquets de l'agrégat est adéquatement choisi. La taille du paquet agrégé est une fonction de l'ETX et de la probabilité de collision. ETX est une métrique basée sur le nombre moyen de retransmissions nécessaires pour recevoir correctement un paquet. En effet, dans un environnement sensible au bruit et riche en interférence, le nombre moyen de retransmission par lien augmente. Prenant ainsi en compte les problèmes d'interférence et d'instabilité de liens, leur étude montre que l'agrégation des paquets peut s'avérer être une bonne option pour maximiser la capacité de la couche MAC. Par ailleurs, leurs résultats corroborent ceux de Niculescu *et al.* [30] qui ont montré qu'en utilisant plusieurs interfaces par noeud, en diversifiant les routes entre une origine et une destination et en agrégeant plusieurs paquets de voix sur IP (VoIP), un réseau maillé sans fil 802.11b peut supporter treize fois plus de sessions de VoIP.

Cheng *et al.* [10] présentent une revue de littérature des différentes modifications à apporter à la couche MAC afin de maximiser la bande passante disponible et rendre ainsi son

fonctionnement plus efficient dans les réseaux maillés sans fil. De plus, leur étude fait ressortir l'importance d'utiliser un modèle *cross-layer* pour que les prestations de qualité de service suivent le dynamisme des réseaux étant donné que la qualité des liaisons sans fil varient dans le temps et en fonction de la charge du réseau. Un modèle *cross-layer* autorise la collaboration entre certaines couches du modèle TCP/IP afin de leur permettre d'échanger des informations telles que l'état du média de transmission, le nombre de connexions actives, les nœuds qui peuvent transmettre des paquets simultanément. Cette collaboration permet aussi aux mécanismes de gestion de la qualité de service de prendre des décisions plus efficaces et de subvenir ainsi plus adéquatement aux exigences des différents trafics.

Dans cet ordre d'idées, Bononi *et al.* [6] ont proposé une architecture nommée CAFMA-SD, permettant à la couche IP d'attribuer dynamiquement la fréquence de communication sur chaque lien sans fil, et à la couche MAC de faire une différenciation de services en se basant sur le fonctionnement de l'EDCA. C'est un modèle conçu pour les réseaux maillés sans fil multi-interfaces et multi-fréquences (en anglais *multi-radio multi-channel*). Chaque routeur du backbone maillé est équipé d'au moins trois interfaces dont une est réservée à l'acheminement des paquets de signalisation et les autres au transfert des données utilisateurs. Au cours du processus de découverte d'une route entre un routeur et une passerelle Internet, les routeurs appartenant au même rayon d'interférence se voient attribuer des fréquences différentes. La différenciation de services dans la couche MAC permet de varier les temps d'attente dans cette couche en fonction des exigences de chaque classe de trafics, et d'octroyer aux applications en temps réel un accès plus fréquent au canal. Le modèle a été implémenté et les performances de l'architecture CAFMA-SD ont été évaluées par simulation. Le réseau étudié est composé de vingt nœuds regroupés en quatre chaînes linéaires de cinq nœuds. Chaque noeud est équipé de trois interfaces et chaque chaîne génère une classe de trafic différente. Les extrémités de chaque chaîne sont la source et la destination de chaque trafic. Les résultats ont révélé que le modèle CAFMA-SD permet effectivement de différencier le délai bout à bout et le taux de paquets délivrés. Cependant, l'effectivité de l'algorithme d'attribution de fréquence n'est pas présentée.

Il existe dans la littérature d'autres modèles de type cross-layer qui ne s'appuient ni sur DiffServ, ni sur la différenciation de services dans la couche MAC pour donner de la qualité de service. Wei *et al.* [48] ont proposé un mécanisme conjoint de contrôle d'admission et de gestion de bande passante qui repose sur l'interaction entre les couches IP et MAC. La capacité de chaque routeur est calculée après avoir construit le graphe de contention du réseau et déterminé les routeurs pouvant communiquer simultanément.

Bien que tous ces travaux proposent des solutions pour pallier les phénomènes d'interférence et procurer de la qualité de service dans les réseaux maillés sans fil, ils ne considèrent

pas la variabilité du canal sans fil dans le temps, ni les effets causés par les évanouissements du signal.

Objectif du mémoire

L'objectif principal de ce projet est de concevoir et déployer un modèle *cross-layer* entre les couches IP, MAC et physique capable de garantir en tout temps la qualité de service des classes de haute priorité dans un environnement sans fil variable dans le temps. Ce modèle nommé *Wireless-DiffServ** est une adaptation du modèle *DiffServ** [39, 3] proposé dans les réseaux de fibre optique (IP/WDM) pour protéger les trafics prioritaires aussi bien contre la congestion que les pannes de la couche physique. Ainsi, *DiffServ** traite en priorité les trafics de voix et de vidéo pour garantir leur qualité de service lorsqu'il détecte des pannes au niveau du support de transmission optique ou bien en cas de congestion de la couche logique. *Wireless-DiffServ** reprend en partie ce concept mais l'applique cette fois-ci aux réseaux sans fil. Pour cela, le mécanisme DiffServ chargé de faire la différenciation de services dans la couche IP, communique avec la couche MAC pour connaître l'état du canal et le débit de transmission sur ce dernier afin de réajuster, si besoin y est, la politique de service des classes. La couche MAC, quant à elle, communique avec la couche physique pour connaître les techniques d'encodage et de modulation de l'information afin de renseigner la couche IP du débit de transmission utilisé. Dans le contexte actuel, les pannes de la couche physique désignent les états du canal. L'implémentation de ce modèle est suivie par l'étude de performance de trois différentes classes de trafic : la voix sur IP, la vidéo sur demande et le transfert de données. L'analyse des données porte sur la mesure du taux de perte, le délai moyen de bout en bout et la gigue. L'étude est réalisée pour des réseaux maillés sans fil IEEE 802.11g.

L'efficacité du modèle *DiffServ** repose essentiellement sur l'utilisation de trois chemins optiques disjoints et indépendants. Ainsi, en vue d'atteindre l'objectif général énoncé ci-haut, trois objectifs spécifiques ont été formulés :

- acheminer le trafic dans un réseau maillé sans fil par trois routes disjointes ;
- assurer autant que possible l'indépendance des routes et
- modéliser les variations temporelles du canal sans fil et définir les états du canal.

La méthodologie proposée repose sur une approche par simulation. Nous utilisons pour cela le simulateur de réseaux NS-2 afin de mettre en place le modèle *Wireless-DiffServ** et évaluer ses performances. Dans un premier temps, nous devons répertorier les algorithmes de routage multi-chemins qui existent dans les réseaux maillés sans fil et ceux qui ont été implémentés dans le simulateur NS-2 en vue de choisir celui qui est le plus adéquat pour la suite des

simulations. Ensuite, nous intégrerons à cet algorithme le support de multiples interfaces configurées sur des fréquences statiques afin d'augmenter la capacité du réseau. Les variations temporelles du canal sans fil sont modélisées par un canal de Rayleigh dans lequel on applique un procédé d'adaptation de modulation et de codage pour pallier les effets d'évanouissement du médium. L'utilisation d'un procédé d'adaptation de modulation et de codage permet d'adapter le débit de transmission des informations en fonction de la qualité du médium sans fil à un instant donné. En adaptant le débit de transmission, on définit ainsi différentes valeurs pouvant caractériser différents états du canal sans fil. L'interaction entre la couche MAC et la couche IP permet à cette dernière de connaître le débit de transmission qui est utilisé sur le lien adjacent et de décider s'il faut ou non modifier la dynamique de service des classes de trafic afin de garantir la qualité de service des classes prioritaires.

Contributions

Les principales contributions de ce mémoire sont les suivantes :

- nous avons d'abord fait une revue de littérature des réseaux maillés sans fil basé sur le standard 802.11 et des algorithmes de routage ;
- ensuite, nous avons cherché un modèle de canal sans fil qui met en évidence l'interaction entre la couche physique et la couche MAC et qui a déjà été testé avec le simulateur de réseaux NS-2 ;
- puis, nous avons implémenté l'architecture DiffServ et la politique de service de la pondération cyclique (WRR) dans l'architecture des noeuds sans fil du simulateur NS-2 et finalement,
- nous avons proposé une nouvelle architecture des noeuds sans fil qui met en relief l'interaction entre les couches IP, MAC et physique. Par la suite, nous avons implémenté cette architecture dans le simulateur de réseau NS-2. Il s'agit d'une contribution majeure car elle n'a jamais été réalisée auparavant dans l'outil de simulation NS-2.

Organisation du mémoire

Le présent mémoire est divisé comme suit. Le chapitre 1 fait une brève revue des réseaux maillés sans fil et du standard IEEE 802.11. L'objectif de ce chapitre est de présenter d'une part l'architecture des réseaux maillés sans fil, leurs caractéristiques et les défis liés à leur déploiement et d'autre part le fonctionnement général des réseaux sans fil 802.11, les imperfections de leur couche MAC et les solutions permettant d'y offrir de la qualité de service. On y présente aussi quelques caractéristiques de la couche physique des réseaux 802.11g. Le deuxième chapitre décrit l'architecture DiffServ, ses composantes et leur fonctionnement. La

description détaillée du modèle de protection *Wireless-DiffServ** y est également présentée, de même que le procédé d'adaptation de modulation et de codage. Le chapitre 3 porte sur la vérification de l'implémentation du procédé d'adaptation de modulation et de codage et de l'implémentation de l'architecture DiffServ. Il présente aussi les différentes classes de trafic (voix, vidéo, et données) et leurs caractéristiques. Le chapitre 4, quant à lui, présente les résultats obtenus après l'implémentation de notre modèle *cross-layer*; ils montrent l'efficacité du modèle *Wireless-DiffServ** comparé au modèle DiffServ. Dans ces simulations, le contrôle de la qualité du canal est désactivé et le WRR utilise une pondération constante quel que soit l'état du canal. Le chapitre 5 présente les résultats obtenus après activation du contrôle de qualité et adaptation de la politique de service en fonction de l'état du canal.

CHAPITRE 1

LES RÉSEAUX MAILLÉS ET LE STANDARD IEEE 802.11

Les technologies de communications sans fil telles que le GSM, le WiFi, le WiMAX pour ne citer que celles là, offrent comme principal avantage la mobilité des utilisateurs. En effet, l'utilisateur est joignable n'importe où et à n'importe quel moment par téléphone, et peut également consulter ses courriers électroniques ou faire des recherches sur Internet en temps réel, quelle que soit sa position géographique. Cependant, le problème inhérent à tous les réseaux sans fil est l'instabilité des liaisons de communication ayant pour conséquence la perte de certaines informations d'une part et le délai de réception d'autre part. Le maillage introduit dans les réseaux sans fil permet, un tant soit peu, de remédier à ce problème en procurant plusieurs routes entre une source et une destination. Cela suppose alors que toutes les liaisons ne peuvent être instables au même moment.

Dans ce chapitre, nous expliquons de manière détaillée le concept de réseau maillé sans fil (en anglais *Wireless Mesh Network (WMN)*). Nous y présentons entre autres les différentes topologies et les difficultés liées à leur déploiement.

1.1 Les réseaux maillés sans fil

Un réseau maillé est un réseau ayant une topologie en forme de grille, c'est-à-dire que chaque équipement peut avoir un ou plusieurs voisins et établir une connexion avec chacun d'eux. Dans un Réseau Maillé Sans Fil (RMSF), les équipements sans fil se connectent de proche en proche par le biais de liaisons radio. Il existe deux catégories d'équipements : les équipements utilisateurs comme les ordinateurs de bureau, ordinateurs portables, imprimantes, etc, appelés *clients mesh*, et les équipements réseau, appelés *routeurs mesh*. Les clients mesh peuvent être fixes ou mobiles tandis que les équipements réseau sont toujours fixes. Les routeurs mesh relaient l'information entre une source et sa destination. Les passerelles mesh sont des routeurs mesh qui interconnectent deux réseaux de type différents. Dans la suite du document, nous désignons sous le terme "nœud" tout équipement réseau ou client. Dans la section suivante, nous abordons les différentes topologies qui peuvent être déployées avec ces équipements.

1.1.1 Architectures des réseaux maillés sans fil

Selon Akyildiz et Wang [2], on distingue trois architectures de RMSF : infrastructure, clients et hybrides.

Mode infrastructure

C'est un réseau dont la dorsale ou backbone est constitué uniquement d'équipements réseau installés à des emplacements fixes et bien déterminés. C'est l'architecture illustrée sur la figure 1.1. Les lignes en trait plein et les lignes en pointillé représentent respectivement les connexions filaires et sans fil. Les routeurs communiquent via une ou plusieurs interfaces radio. De plus, l'accès à l'Internet ou l'intégration des réseaux sans fil existants sont possibles grâce aux passerelles mesh. Dans le cadre de ce projet, nous allons nous concentrer sur le backbone des RMSFs.

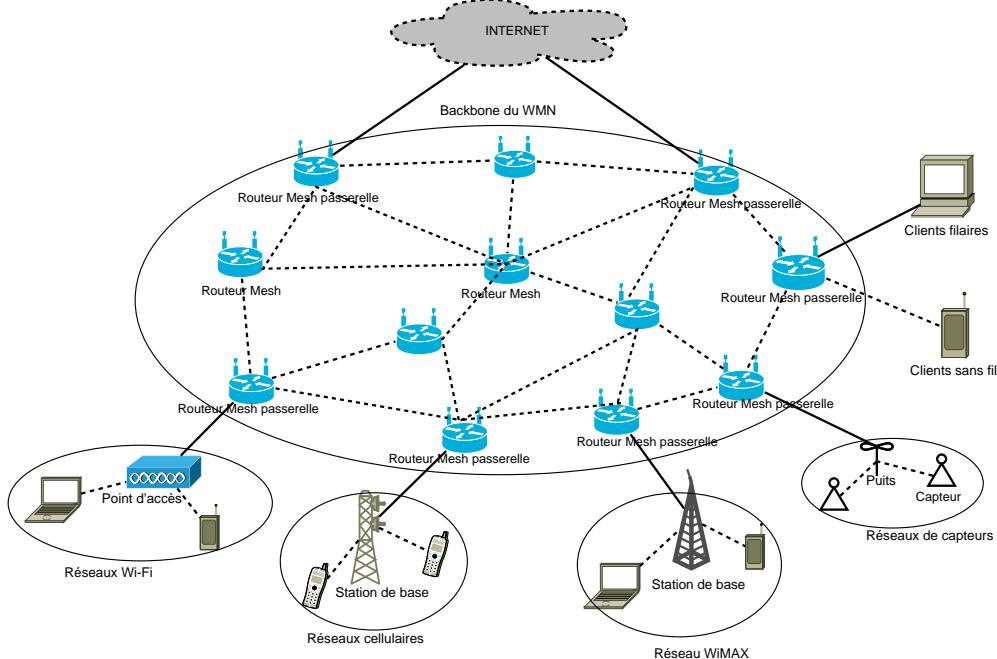


Figure 1.1 Architecture d'un réseau maillé sans fil de type infrastructure

Mode client

Cette architecture ne nécessite pas de routeur mesh. Elle est constituée uniquement de clients mesh connectés en mode Ad hoc. Ces derniers, mobiles ou fixes, assurent les fonctions de configuration et de routage et garantissent aux utilisateurs l'accès aux applications désirées. Un exemple typique de ce réseau est présenté sur la figure 1.2.

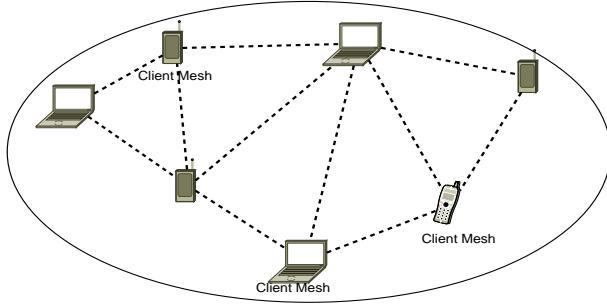


Figure 1.2 Architecture d'un réseau maillé sans fil de type client

Réseau Maillé Sans Fil hybride

Cette architecture résulte de la combinaison des deux configurations précédentes, c'est-à-dire que le backbone est maillé et une passerelle mesh est reliée à un équipement du réseau de clients maillés afin de leur permettre d'accéder à Internet.

1.1.2 Caractéristiques et problématique des RMSF

Contrairement aux réseaux filaires, dans lesquels le maillage est dispendieux et difficile à déployer, le maillage dans les réseaux sans fil est une solution moins coûteuse, plus facile à déployer et confère aux RMSF les principales caractéristiques suivantes :

- réseau multi-saut : les données des clients sont acheminées à travers un ou plusieurs nœuds intermédiaires [26] avant d'atteindre le nœud destination qui, dans ce cas, est toujours la passerelle vers Internet [17, 5].
- réseau adaptatif : un RMSF est un réseau capable de s'auto-déployer, car les connexions entre les équipements sont dynamiques, de s'auto-configurer, car il ne nécessite pas l'intervention humaine pour les opérations de maintenance, et de s'auto-cicatriser, car il est capable de se ré-organiser pour rester fonctionnel lorsqu'une panne survient [32].
- réseau étendu : l'ajout d'un équipement réseau permet d'étendre la portée du RMSF [29]. En effet, le réseau étant adaptatif, une nouvelle connexion s'établit entre deux équipements pour peu qu'ils se situent dans le rayon de communication l'un de l'autre.

Les RMSF peuvent être déployés avec n'importe quelle technologie sans fil : WiFi (802.11s), WiMAX (802.16d), Bluetooth et Zigbee (802.15) [14, 29]. Les groupes de travail *IEEE 802.11s*, *IEEE 802.16 mesh* et *IEEE 802.15.5* s'occupent respectivement des questions relatives à la standardisation de ces trois (3) types de réseaux [27, 21].

Bien que les avantages liés au maillage des réseaux sans fil soient nombreux, leur mise en place reste un véritable défi [32, 2]. On relève dans la littérature plusieurs problèmes relatifs à leur déploiement notamment le problème de positionnement des routeurs/passerelles mesh

[45, 12], le routage et d'ordonnancement de liens [16, 17, 5, 43, 44, 11], l'attribution de fréquence [35, 36], sans oublier les problèmes de fiabilité et de qualité de service [38, 23, 30]. De plus, ces problèmes sont tous étroitement liés [32] et les approches de solution expérimentées jusqu'à ce jour affectent les performances globales du réseau (délai, débit, taux de livraison, etc). Jun et Sichitiu [24] ont montré que lorsque la taille du backbone du réseau sans fil augmente, la capacité effective du réseau diminue de $O(1/n)$ où n est le nombre de noeuds dans le réseau. Il apparaît alors un dilemme en ce sens que la portée du réseau augmente au détriment de la qualité de service.

Dans ce projet, nous nous intéressons à la qualité de service dans les réseaux maillés sans fil. Puisque ces réseaux peuvent être déployés avec n'importe quelle technologie sans fil, nous avons choisi d'étudier uniquement les réseaux de type 802.11 car c'est la technologie la plus utilisée. Ainsi dans la section suivante, nous abordons les aspects fondamentaux de la norme IEEE 802.11.

1.2 Le standard IEEE 802.11

Les différences principales entre les réseaux sans fil et les réseaux filaires se situent au niveau des couches MAC et PHY. En effet, les supports de transmission étant différents (câbles de cuivre vs ondes électromagnétiques), cela se traduit par des protocole différents.

Le standard IEEE 802.11 définit les spécifications des couches MAC (en anglais *Medium Access Control*) et physique (PHY) des réseaux locaux sans fil ou Wireless Local Area Networks (WLANs) en anglais.

1.2.1 Topologie

Le standard 802.11 définit deux configurations de base pour faciliter la communication des équipements. Dans ce contexte, un équipement, utilisateur ou réseau, est appelé *station*. Un ensemble de stations forme un BSS (Basic Service Set en anglais). Les stations peuvent être connectées en mode ad hoc ou en mode infrastructure.

BSS indépendants

Encore appelé réseaux ad hoc, les BSS indépendants sont composés de stations, mobiles ou fixes, communiquant directement entre elles sans l'intermédiaire d'un point d'accès. Cette architecture est identique à celle des réseaux maillés de type client (voir section 1.1.1). La seule différence réside dans le protocole utilisé. Dans les réseaux ad hoc, tous les équipements utilisent le protocole 802.11 pour accéder au canal.

BSS avec infrastructure

Configurées en mode infrastructure, les stations communiquent entre elles uniquement via un point d'accès sans fil. Le point d'accès sert également d'interface entre le réseau sans fil et n'importe quel autre réseau sans fil, filaire ou Internet. C'est cette architecture que l'on retrouve généralement chez les particuliers ou dans les entreprises. Lorsque deux ou plusieurs BSS sont connectés, on parle de ESS (de l'anglais *Extended Service Set*).

1.2.2 Couche MAC

Comme son nom l'indique, la couche MAC ou Medium Access Control contrôle l'accès au media. Elle détermine quand une transmission peut être initiée, quand un paquet est corrompu et doit être retransmis. Elle dispose, pour cela, de deux protocoles d'accès et de trames de signalisation.

1.2.2.1 Protocoles d'accès au médium

Le standard IEEE 802.11 spécifie deux (2) protocoles d'accès au médium sans fil : *Point Coordination Function* (PCF) et *Distributed Coordination Function* (DCF).

Le PCF est un algorithme d'ordonnancement centralisé qui permet au point d'accès sans fil de déterminer quand chaque station qui désire transmettre un paquet peut accéder au canal. Cette méthode permet une meilleure gestion de la bande passante puisqu'elle évite les collisions. Cependant, la plupart des équipements présents sur le marché n'offrent pas cette option mais uniquement le DCF [47].

Le DCF est le mécanisme basé sur le protocole CSMA/CA (en anglais *Carrier Sense Multiple Access with Collision Avoidance*). Selon ce mécanisme, lorsqu'une station veut transmettre un paquet, elle écoute le canal et s'assure que ce dernier est disponible pendant un temps DIFS (en anglais *Distributed Inter-Frame Space*). Elle choisit ensuite aléatoirement un temps de backoff dont la durée en nombre de slots varie entre 0 et $CW - 1$. Le temps de backoff permet d'éviter que deux stations transmettent simultanément des paquets. La valeur de CW (en anglais *Contention Window*) est égale à la taille de la fenêtre de contention. Ce temps de backoff diminue progressivement aussi longtemps que le canal est disponible après un temps DIFS, s'interrompt quand ce dernier est occupé à transmettre le paquet d'une autre station, et est réactivé lorsque le canal est de nouveau disponible pendant un temps DIFS. Une station n'est autorisée à transmettre que lorsque son temps de backoff atteint 0. Une fois le paquet transmis, la destination doit envoyer, dans un intervalle de temps égal à SIFS (en anglais *Short Inter-Frame Space*), un ACK à la source pour lui signifier que le paquet a bien été reçu. Si après un intervalle EIFS (en anglais *Extended Inter-Frame Space*) la source ne

reçoit pas l'ACK, elle retransmet le paquet en supposant qu'il y a eu une collision et recalcule le temps de backoff. Au départ le temps de backoff est uniformément distribué dans l'intervalle $[0, CW_{min}]$. En cas de collision ou de corruption de paquet, cet intervalle est doublé à chaque retransmission du paquet jusqu'à atteindre la valeur maximale $CW_{max} = 2^m CW_{min}$ [38, 47]. L'augmentation exponentielle du temps de backoff augmente non seulement le temps de séjour des paquets dans les files d'attente mais, comme nous le verrons dans la sous-section 1.3.1, diminue aussi la capacité effective du réseau. La figure 1.3 schématise le fonctionnement du mécanisme DCF tel que décrit ci-haut.

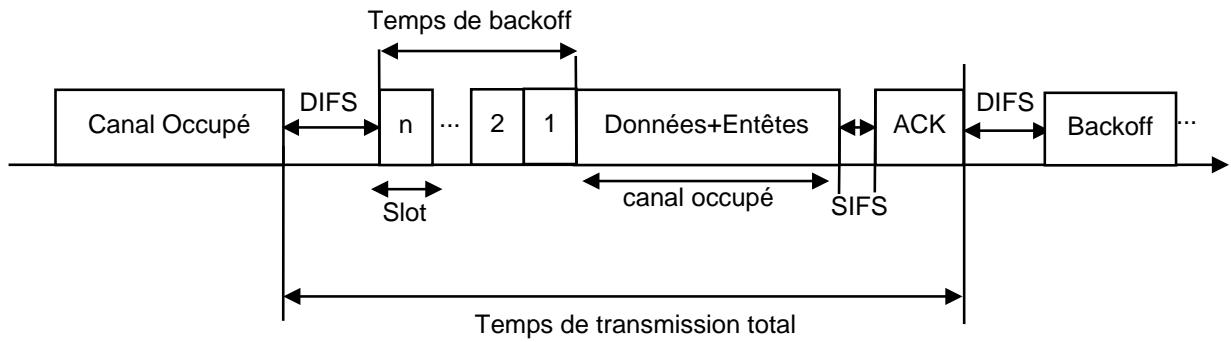


Figure 1.3 Fonctionnement du mécanisme DCF

1.2.2.2 Trames MAC

Il existe deux types de trames dans la couche MAC des réseaux 802.11 : données et contrôle.

Une trame de données comprend un en-tête, les données à transmettre et un champ de vérification d'erreurs. L'en-tête indique entre autres la nature et la durée de la trame ainsi que les adresses source et destination. Le champ de vérification d'erreurs, quant à lui, permet de déterminer si la trame reçue est erronée ou non.

On distingue plusieurs trames de contrôle, mais toutes sont composées de deux parties : un en-tête et un champ de vérification d'erreurs.

1. la trame RTS : cette trame (RTS ou Ready To Send) est envoyée par le nœud source lorsque ce dernier est prêt à envoyer un paquet vers une destination connue ;
2. la trame CTS : lorsque le nœud destination est prêt à recevoir une trame, en réponse à la trame RTS reçue, il envoie à la source une trame CTS (Clear To Send). Le processus RTS/CTS permet au couple émetteur-récepteur de réserver de la bande passante afin de réduire les risques de collisions avec les autres stations. Cependant, nous ne l'utilisons pas dans notre étude car il génère du trafic supplémentaire.

3. Trame ACK : C'est le paquet envoyé par la destination lorsqu'elle reçoit correctement et sans erreur une trame de données.

1.2.3 Couche physique

La couche physique, telle que spécifiée par le standard IEEE 802.11 [22], est subdivisée en deux sous couches : la sous-couche PLCP (en anglais *Physical Layer Convergence Protocol*), et de la sous-couche PMD (en anglais *Physical Medium Dependent*). La sous-couche PLCP sert d'interface entre la couche MAC et la couche PHY ; c'est elle qui indique à la couche MAC que le médium sans fil est libre et prêt à transmettre un paquet. La sous-couche PMD quant à elle, s'occupe des techniques de codage/décodage et de modulation/démodulation des données. Soulignons que c'est l'entête PLCP qui contient la valeur du taux de transmission des données, ce qui permet à la sous-couche PMD d'appliquer la technique d'encodage/décodage et de modulation/démodulation qui convient.

Le standard IEEE 802.11 définit différentes normes physiques qui se distinguent par la technique de modulation et la fréquence de fonctionnement. Le tableau 1.1 présente les caractéristiques des normes de la famille 802.11.

Tableau 1.1 Principales normes physiques de la famille 802.11

Norme	Caractéristiques
IEEE 802.11a	Bande de fréquence 5 GHz Technique OFDM Taux de transmission jusqu'à 54 Mbps
IEEE 802.11b	Bande de fréquence 2,4 GHz - 2,4835 GHz Technique DSSS Taux de transmission jusqu'à 11 Mbps
IEEE 802.11g	Bande de fréquence 2,4 GHz - 2,4835 GHz Technique OFDM Taux de transmission jusqu'à 54 Mbps
IEEE 802.11n	Bande de fréquence 2,4 GHz - 2,4835 GHz et 5 GHz Technique OFDM et MIMO Taux de transmission jusqu'à 300 Mbps

Bien que la norme 802.11n offre des débits très élevés, faisant d'elle le meilleur candidat pour le backbone des réseaux maillés, nous portons notre étude sur la norme 802.11g parce que d'une part, elle est la plus déployée, et d'autre part, parce qu'elle est déjà implémentée dans le simulateur que nous utiliserons par la suite.

La norme 802.11g utilise le multiplexage OFDM (en anglais *Orthogonal Frequency Division Multiplexing*) qui consiste à décomposer un système large bande en plusieurs sous-

systèmes orthogonaux à bande étroite. Généralement appelé sous-porteuse, chaque sous-système est utilisé pour transmettre des données. L'orthogonalité des sous-porteuses annule les interférences mutuelles et optimise l'utilisation de la bande passante. La norme 802.11g utilise le codage convolutionnel afin de minimiser la probabilité d'erreur par paquet face aux bruits et aux atténuations du signal.

La combinaison de divers schémas de modulation avec différents taux de codage permet d'atteindre un débit de transmission compris entre 6 Mb/s et 54 Mb/s. Le tableau 1.2 présente tous les schémas de modulation et codage spécifiés par la norme 802.11g [20]. Il est à noter que n'importe laquelle de ces combinaisons peut être utilisée pour transmettre les paquets de données alors que les paquets de contrôle sont envoyés au taux de 6 Mb/s.

Le tableau 1.3 présente les principaux paramètres des couches MAC (espacement inter-trames, taille de la fenêtre de contention) et PHY (en-tête PLCP) dans la norme 802.11g.

Tableau 1.2 Taux de transmission du Standard IEEE 802.11g

Modulation	Taux de codage	Taux de transmission (Mb/s)
BPSK	1/2	6
BPSK	3/4	9
QPSK	1/2	12
QPSK	3/4	18
16QAM	1/2	24
16QAM	3/4	36
64QAM	2/3	48
64QAM	3/4	54

Tableau 1.3 Paramètres des couches MAC et PHY

Paramètres	Norme 802.11g
slot	$9 \mu s$
SIFS	$10 \mu s$
DIFS = SIFS + $2 t_{slot}$	$28 \mu s$
En-tête PLCP	15 octets
CW_{min}	15
CW_{max}	1023

1.3 Performances des réseaux 802.11

Dans cette section, nous allons étudier l'impact du mécanisme CSMA/CA et des paramètres de la couche PHY sur la performance des réseaux 802.11g afin de mettre en évidence les problèmes liés à la Qualité de Service (QoS) et l'impact des temps inter-trames sur la capacité réelle du réseau.

1.3.1 Temps de transmission d'un paquet

Nous avons vu dans la section 1.2.2.1 que le protocole CSMA/CA ajoute des temps d'attente assez variable entre l'envoi de 2 paquets consécutifs. La figure 1.4 montre la décomposition du temps total requis pour transmettre un paquet avec succès. Comme on peut le voir, il est composé de 2 parties : le temps de service de la trame de données (T_L) et le temps d'overhead (T_o) qui comprend les temps DIFS et SIFS, le temps de backoff, le temps de transmission des en-têtes et le temps d'acquittement.

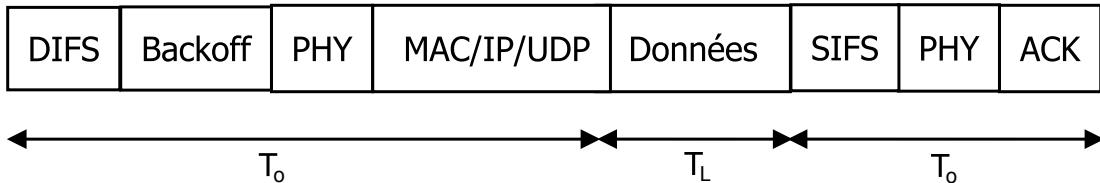


Figure 1.4 Décomposition du temps de transmission d'un paquet

1.3.1.1 Temps de transmission d'un paquet sans erreurs

Soit T_L le délai de transmission des données brutes de longueur L octets. Si R est le taux de transmission du canal, il s'ensuit que :

$$T_L = \frac{8L}{R} \quad (1.1)$$

Le débit R dépend de la modulation et du codage utilisés et peut varier entre 6 Mb/s et 54 Mb/s.

Le délai d'attente T_o engendré par l'overhead est défini par :

$$T_o = DIFS + T_{PHY} + T_{UDP/IP/MAC} + SIFS \quad (1.2)$$

$T_{UDP/IP/MAC}$ est le temps nécessaire pour transmettre les en-têtes des protocoles UDP, IP et MAC. Le protocole UDP est le plus souvent utilisé pour les applications en temps réel. Ces

en-têtes sont envoyées au même taux que les données brutes soit à un débit R . T_{PHY} est le temps de transmission de l'en-tête PLCP de la couche physique. Il est envoyé au taux de 6 Mb/s.

Le délai T_A nécessaire pour acquitter un paquet bien reçu est défini par :

$$T_A = T_{PHY} + T_{ACK} \quad (1.3)$$

où T_{ACK} est le délai de transmission d'un ACK. Les ACK ont une taille de 14 octets et sont envoyés au taux de 6 Mb/s.

Ces valeurs sont présentées dans le tableau 1.4.

Tableau 1.4 Temps d'overhead par paquet en fonction du débit de transmission

Overhead	Taille (octets)	Temps (μs)
En-tête UDP/IP	20	$20 \times 8/R$
En-tête MAC	34	$34 \times 8/R$
En-tête PHY	15	20
ACK	14	18,67

Le temps de backoff T_B dépend des erreurs de transmission et des collisions dans le réseau. Le temps moyen de backoff après i retransmissions peut être estimé par l'équation (1.4) proposée par [34] :

$$T_B(i) = \frac{2^i (CW_{min} + 1) - 1}{2} t_{slot} \quad (1.4)$$

où CW_{min} est la largeur minimale de la fenêtre de contention et t_{slot} , la durée d'un slot. Ainsi, le temps minimum nécessaire pour transmettre correctement un paquet lorsqu'aucune retransmission n'est requise est T_{succ} défini par :

$$T_{succ} = T_o + T_L + T_B(0) + T_A \quad (1.5)$$

Étude de cas

Considérons un lien entre un nœud source et un nœud destination. Les deux nœuds utilisent la norme 802.11g et on suppose qu'il n'y a aucun risque de collision entre paquets. De plus, si la probabilité qu'un paquet soit corrompu est très faible ($\leq 10^{-4}$), nous pouvons aussi supposer qu'aucun paquet n'est retransmis. Par conséquent, en remplaçant les termes de l'équation (1.5) par les valeurs numériques des tableaux 1.3 et 1.4, le temps moyen de

transmission (en μs) d'un paquet devient :

$$T_{\text{succ}} = (28 + 2 \times 20 + 10 + 18, 67) + \frac{8 \times (34 + 20 + L)}{R} + \frac{9 \times CW_{\min}}{2} \quad (1.6)$$

Calculons le temps de transmission des paquets de trois différents trafics : voix, vidéo et données. Les trafics sont de type ON-OFF. Les durées moyennes des périodes ON et OFF suivent des lois exponentielles de moyenne 500 ms. Les paquets des trafics voix et données ont une longueur constante de 80 octets et 1440 octets respectivement. Les paquets du trafic vidéo ont une longueur uniformément distribuée dans l'intervalle [100, 1500] octets. La taille moyenne des paquets vidéo est 800 octets.

Connaissant le temps de transmission et le taux d'arrivée des paquets, nous pouvons déterminer, pour chaque trafic, le nombre maximal de sources qui peuvent être actives simultanément par la formule :

$$\beta = \frac{1}{\lambda T_{\text{succ}}} \quad (1.7)$$

où λ est le taux d'arrivée en paquet/s par source. La charge maximale (sans en-tête) que le réseau peut supporter est donnée par l'expression :

$$\mathcal{D} = \frac{8L}{T_{\text{succ}}} \quad (1.8)$$

On suppose que le taux de transmission R du canal est constant : $R = 54 \text{ Mb/s}$ et $CW_{\min} = 15$ dans les réseaux 802.11g. Le tableau 1.5 présente les valeurs numériques de β , \mathcal{D} et T_{succ} pour chaque type de trafic.

Tableau 1.5 Nombre maximal de sources par classe

Traffic	Voix	Vidéo	Données
L (octets)	80	800	1440
λ (pqt/s)	50	5	3
$T_L (\mu s)$	11,852	118,519	213,333
$T_o (\mu s)$		66	
$T_B(0) (\mu s)$		67,5	
$T_A (\mu s)$		38,67	
$T_{\text{succ}} (\mu s)$	184,02	290,69	385,50
β (sources)	109	688	934
\mathcal{D} (Mb/s)	3,48	22,02	29,88

L'analyse de ce tableau révèle que les temps d'overhead de la couche MAC ont un effet néfaste sur la capacité maximale de sources de voix admissibles. En effet, les résultats de cet

exemple montrent que le réseau étudié ne peut pas accepter plus de 109 connexions de voix sans dégrader la qualité de service offerte à ce type de trafic. Ceci est lié au fait que les temps de backoff ($T_B(0)$) et d'overhead (T_o) sont plus importants que le délai de transmission de l'information brute (T_L). Le réseau passe donc seulement 6% du temps total de transmission d'un paquet de voix à envoyer les données utiles contre 73% de ce temps à envoyer des en-têtes ou à rien faire. C'est ce qui explique la limite observée pour le débit effectif du trafic voix. On constate, toutefois, que l'impact de la couche MAC s'amenuise en augmentant la taille des paquets. En effet, la valeur de β pour les trafics de vidéo et données indiquent que la capacité maximale du réseau peut être augmentée en augmentant la taille des paquets à transmettre. Par ailleurs, soulignons que les valeurs de β et \mathcal{D} vont décroître lorsque le taux de transmission diminue ou en cas de retransmission, tandis que celle de T_{succ} augmentera.

1.3.1.2 Temps de transmission avec erreurs

Si l'acquittement n'est reçu qu'après n retransmissions alors la durée totale $T(n)$ d'une transmission réussie est définie par :

$$T(n) = (n + 1)(T_o + T_L) + \sum_{i=0}^n T_B(i) + T_A, \quad n \geq 1. \quad (1.9)$$

En remplaçant les variables de l'équation (1.9) par les valeurs numériques du tableau (1.4), on obtient :

$$T(n) = (n + 1) \left(58 + \frac{8(54 + L)}{R} \right) + \sum_{i=0}^n T_B(i) + 38,67 \quad (1.10)$$

où $T(n)$ est exprimé en μs .

Le temps de transmission d'un paquet dépend donc de la longueur L du paquet, du taux de transmission R et du temps de backoff qui augmente de façon exponentielle lorsqu'une retransmission du paquet s'avère nécessaire. La connaissance du temps moyen de transmission donne finalement une idée de la capacité maximale du réseau par la formule :

$$\mathcal{D} = \frac{8L}{T(n)} \quad (1.11)$$

L'équation (1.11) indique que la capacité effective des réseaux 802.11 est fortement liée à la taille des paquets à transmettre. De plus, comme nous l'avons vu dans la section 1.3.1.1, cette capacité augmente lorsque la taille des paquets est grande.

Cette étude nous a permis de mettre en évidence l'impact du protocole CSMA/CA sur la

capacité effective des réseaux 802.11 où les paquets de petite taille, comme la voix, surchagent le réseau d'en-têtes. La transmission de petits paquets n'est donc pas efficace.

Le second inconvénient de la couche MAC 802.11 est lié au fait que tous les paquets envoyés par les utilisateurs sont traités de la même façon, sans tenir compte de leurs exigences en qualité de service. Étant donné que les applications de voix et de vidéo ont des critères de qualité de service très sévères, et qu'elles doivent rivaliser avec les applications de transfert de données pour accéder au canal, on a proposé, dans le standard 802.11e, des mécanismes permettant de satisfaire dans la mesure du possible la qualité de service des applications de voix et de vidéo dans les réseaux sans fil 802.11.

1.3.2 Mécanismes de QoS dans les réseaux 802.11

Dans les réseaux 802.11, les paquets sont délivrés avec le *mieux effort*, puisque la norme ne définit aucune méthode pour donner de la QoS. Dans [49], les auteurs ont présenté plusieurs solutions, adaptées au mécanisme DCF, pour offrir une qualité de service relativement meilleure que le *mieux effort*. Ces méthodes peuvent être classifiées en trois catégories.

1.3.2.1 Différenciation de services

La différenciation de service définie dans la norme IEEE 802.11e, fonctionne de deux manières dans les WLANs. L'une consiste à accorder une priorité de service à certains flots de trafic en réduisant la taille maximale de la fenêtre de contention (CW), le temps de backoff et les espacements inter-trames IFS qui leur sont associés dans la couche MAC.

La deuxième approche consiste à partager la largeur de bande de façon équitable entre les applications en leur accordant des poids. Les flots provenant d'applications sensibles au délai reçoivent les poids les plus élevés. La valeur des poids est ensuite prise en compte dans le calcul des paramètres de contention associés à chaque application. Ainsi, le temps de backoff qui précède l'envoi d'un paquet appartenant à un flot prioritaire, donc de poids élevé, est plus petit que le temps calculé pour les flots non prioritaires, de poids plus faible. Il en est de même pour les temps IFS et la taille de la fenêtre de contention.

Bien que les spécifications du standard 802.11e permettent d'améliorer la qualité de service des applications les plus sensibles, les effets de l'overhead ne sont pas pour autant résolus et réduisent toujours la performance du réseau.

1.3.2.2 Contrôle d'admission et réservation de bande passante

Le contrôle d'admission et la réservation de bande passante sont deux méthodes qui permettent de garantir la qualité de service du trafic qui sera admis dans le réseau. En effet,

le contrôle d'admission permet de déterminer s'il est possible de transmettre un nouveau flot d'information sans dégrader la qualité de service exigée par les flots en cours de transmission. Il est donc nécessaire de connaître la largeur de bande totale disponible et celle qui a déjà été réservée par les applications. Cependant, la variabilité du canal sans fil et sa nature partagée ne permettent pas de connaître avec précision la bande passante disponible. Cette situation rend particulièrement difficile le contrôle d'admission et la réservation de bande passante dans les réseaux sans fil. Deux approches ont tout de même été proposées pour faire du contrôle d'admission dans le 802.11. La première consiste à estimer les conditions du canal en mesurant le délai et le débit moyen tandis que la deuxième calcule un coefficient de saturation à partir d'informations telles que le nombre de stations actives, leur débit de transmission, la longueur des paquets à transmettre [4, 25]. La valeur de ce coefficient ou les conditions du canal indique si un nouveau flot peut être accepté dans le réseau.

1.3.2.3 Adaptation de lien

Utilisée dans la couche PHY, l'adaptation de lien est une méthode qui permet de varier le taux de transmission des paquets en fonction des conditions du canal. Les différents taux de transmission sont obtenus en combinant différentes techniques de modulation et de codage. La plupart de ces algorithmes sont basés sur une ou des combinaisons des métriques suivantes : le rapport signal sur bruit (SNR), le taux d'erreur par paquet, le nombre d'ACK reçus [49]. Notons par ailleurs que certains auteurs [33] ont montré que l'utilisation des procédés d'adaptation de modulation et de codage améliore l'efficacité spectrale des réseaux sans fil et maximise le débit effectif. Nous reviendrons sur l'adaptation de lien dans le chapitre 2.

Dans ce chapitre, nous avons expliqué le concept des réseaux maillés sans fil et passé en revue les caractéristiques principales de la norme IEEE 802.11. Après avoir évoqué les différentes normes physiques, notre choix s'est porté sur le 802.11g. L'étude du fonctionnement de la couche MAC nous a permis de mettre en évidence l'overhead produit par les paquets de petite taille et la nécessité de respecter les besoins de qualité de service lorsque plusieurs applications convergent sur un même canal sans fil. Si les méthodes présentées dans ce chapitre permettent d'offrir de la qualité de service dans les réseaux 802.11, elles ne donnent cependant aucune garantie de qualité de service. Elles permettent seulement à chaque station d'offrir différents traitements aux applications. De plus, elles opèrent dans les couches MAC et PHY indépendamment des mécanismes des couches supérieures.

Puisque notre objectif est de garantir une bonne qualité de service dans les réseaux maillés 802.11, nous étudierons dans le chapitre 2 les mécanismes de qualité de service proposés par la couche IP puis nous présenterons le modèle utilisé dans notre projet.

CHAPITRE 2

DIFFÉRENCIATION DE SERVICES IP ET MODÈLE WIRELESS-DIFFSERV*

Mis à part les solutions présentées dans la section 1.3.2, il existe d'autres mécanismes, offerts par les couches Transport et IP, permettant de fournir de la qualité de service dans les réseaux IP. Étant donné que nous nous intéressons au transfert des applications multimédia en temps réel via un réseau sans fil et, comme le protocole Transmission Control Protocol (TCP) comporte des problèmes dans les environnements sans fil [7, 41], nous utilisons dans cette étude le protocole User Datagram Protocol (UDP) comme protocole de couche Transport. Le protocole UDP ne procure pas de mécanisme de fiabilité et s'appuie uniquement sur les solutions des couches IP et/ou MAC.

Nous présentons dans ce chapitre une brève revue des mécanismes de différenciation de services utilisés dans la couche IP puis nous expliquons le fonctionnement du modèle *DiffServ** ainsi que les composantes nécessaires à son intégration dans les réseaux sans fil. Originellement, *DiffServ** a été proposé pour les réseaux IP optiques. Étant donné que ce projet consiste à l'implémenter dans les réseaux sans fil, nous nommons ce modèle *Wireless-DiffServ** qui résulte de l'utilisation conjointe de la différenciation de service IP, du routage multi-trajets et de l'adaptation de liens.

2.1 Architecture DiffServ

On distingue deux mécanismes de couche IP permettant d'offrir de la qualité de service : les techniques de réservation de ressources (IntServ) et la différenciation de services (DiffServ). Bien que les protocoles soient différents, les techniques de réservation de ressources (IntServ) sont semblables aux méthodes de réservation de bande passante de la couche MAC (voir section 1.3.2). En effet, IntServ permet de réserver dynamiquement de la bande passante et de garantir une qualité de service pour les applications qui le demandent. Cependant, il fait face aux mêmes problèmes que les techniques de réservation de bande passante de la couche MAC. De plus [28] a montré que IntServ est une architecture très complexe et difficilement utilisable dans les grands réseaux. Dans cette section, nous nous attardons uniquement sur la différenciation de services (DiffServ).

2.1.1 Fonctionnement de DiffServ

DiffServ est un mécanisme de différenciation de la couche IP qui traite les paquets envoyés à travers un réseau IP de façon différente. L'architecture DiffServ utilise la notion de classe pour servir les flots de trafic. Une classe de service est le regroupement des flots de trafic ayant les mêmes exigences de qualité de service. Les flots appartenant à une même classe de service reçoivent le même traitement. Les applications en temps réel, comme la voix et la vidéo, sont celles qui bénéficient toujours d'un traitement préférentiel, tandis que les applications de transfert de données sont servies avec le *meilleur effort*. La différenciation de service est particulièrement efficace lorsqu'une congestion survient dans le réseau. En effet, en cas de congestion, DiffServ s'assure que les classes bénéficiant du traitement préférentiel sont toujours mieux servies que les paquets des autres classes. Les flots préférentiels ont donc une meilleure qualité de service que les autres flots.

Les principales parties du module DiffServ sont présentées sur la figure 2.1. On y retrouve le classificateur de paquets, les files d'attente et l'ordonnanceur ou scheduler. Le classificateur identifie la classe de chaque paquet entrant et le met dans la file d'attente virtuelle qui lui correspond. Il y a autant de files d'attente que de classes. L'ordonnanceur, quant à lui, utilise un algorithme de sélection, comme les priorités ou la pondération cyclique, pour sortir les paquets de leur file respective et s'assurer que chaque flot reçoit le traitement qui lui convient. Dès que les paquets sont sortis de leur file d'attente, ils sont encapsulés dans la couche MAC et envoyés sur le média physique. En général, DiffServ est implémenté séparément sur chaque interface de sortie des routeurs, c'est-à-dire entre les couches IP et MAC, et le principe de fonctionnement de DiffServ est le même peu importe le média physique utilisé : Ethernet, fibre optique, canal sans fil.

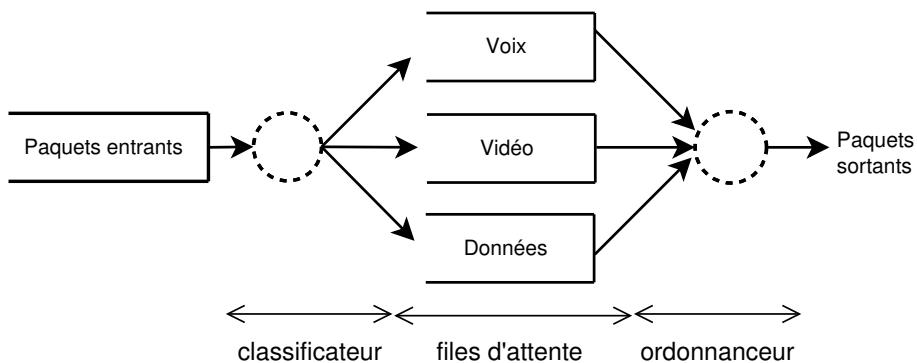


Figure 2.1 Fonctionnement standard de DiffServ dans un réseau IP

2.1.2 Politiques d'ordonnancement

Plusieurs politiques de service peuvent être utilisées par l'ordonnanceur DiffServ pour accorder à chaque flot le traitement qui lui convient [42]. Les files d'attente peuvent être servies selon un ordonnancement prioritaire (PRI), Weighted Round Robin (WRR) ou Round Robin (RR). Lorsque l'ordonnancement est de type PRI, toutes les files de haute priorité sont servies, et doivent être vides avant de servir les files de basse priorité. Les ordonnanceurs de type WRR affectent des poids aux files. Les files qui possèdent les poids les plus élevés reçoivent une portion de bande passante plus importante. Avec un ordonnanceur RR, toutes les files ont la même priorité et sont servies à tour de rôle ; c'est une variante du WRR où tous les poids sont égaux.

Dans cette étude, l'ordonnanceur DiffServ utilise le mécanisme de pondération cyclique ou WRR pour sélectionner les paquets à transmettre. Le WRR est un algorithme qui affecte différents poids aux files d'attente du module DiffServ. Les files correspondant aux classes qui doivent recevoir un traitement préférentiel se voient attribuer les poids les plus élevés et les paquets de ces files ont accès au média physique beaucoup plus souvent que les paquets des autres files. Ce qui permet de réduire leur temps d'attente.

Illustrons nos propos par un exemple. Considérons trois classes de trafic : la voix, la vidéo et les données dont les files d'attente respectives ont pour poids : 8, 4 et 1. Supposons aussi que tous les paquets sont de même longueur et que les files d'attente sont infinies. A la fin d'un cycle de service, l'ordonnanceur WRR aura transmis 13 paquets dont 8 paquets de voix, 4 paquets vidéo, et, 1 paquet de données. L'ordonnanceur passe donc 62% du temps à servir la file de voix, 31% du temps à servir la file de vidéo et 7% du temps à servir la file de données. Dans cet exemple, ce sont donc les trafics de voix et de vidéo qui reçoivent un traitement particulier. Soulignons que lorsque les paquets sont de différentes tailles, ce paramètre doit être pris en compte dans le calcul des proportions de temps allouées pour servir chaque file.

Des files d'attente désservies par une politique de service telle que le WRR peuvent être approximées par un système constitué de n files d'attente, où n est le nombre de classes de service définies. La capacité C_i du lien qui sert chaque file est donnée par :

$$C_i = \frac{\alpha_i L_i}{\sum_{k=1}^n \alpha_k L_k} C \quad (2.1)$$

où C est la capacité totale du lien, L_i est la longueur moyenne avec en-têtes des paquets servis par la file i et α_i est le poids assigné à la file i [40]. Lorsque la taille moyenne des paquets varie d'une classe à l'autre, la valeur de C_i indique alors la répartition réelle de la largeur de bande entre les différentes classes.

L'utilisation du WRR affecte aussi les conditions de stabilité des files d'attente lorsque le

buffer est infini. En effet, les files restent stables aussi longtemps que leur facteur d'utilisation est inférieur à l'unité. Soient λ_i et μ_i , respectivement les taux d'arrivée et de service (en paquet/s) des paquets de la classe i . Soit ρ_i le facteur d'utilisation du lien desservant la classe i .

$$\rho_i = \frac{\lambda_i}{\mu_i} = \frac{\lambda_i L_i}{C_i} \quad (2.2)$$

La file d'attente de la classe i est stable si

$$\rho_i < 1 \quad \Rightarrow \quad \frac{\lambda_i L_i}{C_i} < 1 \quad (2.3)$$

2.2 Modèle DiffServ*

Basé sur l'architecture DiffServ, *DiffServ** est un modèle proposé par [3] pour protéger les trafics de haute priorité contre les congestions et les pannes de la couche optique et garantir ainsi leur qualité de service dans les réseaux maillés optiques lorsqu'une rupture de câbles survient. L'objectif de ce projet est d'implémenter le modèle *DiffServ** dans les réseaux maillés sans fil. Dans cette section, nous expliquons le modèle *DiffServ** tel qu'il a été pensé pour les réseaux optiques puis nous présentons l'adaptation proposée pour les réseaux sans fil.

2.2.1 DiffServ* dans les réseaux IP/WDM

Dans les réseaux filaires et optiques, la qualité du canal de transmission est une variable à deux états. Lorsque le canal est bon, tous les paquets sont envoyés à débit constant. Lorsque le canal est mauvais, aucun paquet ne peut être transmis car le débit est nul. Ainsi, s'il existe un unique chemin entre une source de trafic et sa destination, la différenciation offerte par DiffServ prend fin lorsqu'une défaillance apparaît sur un des supports de transmission. Cela se traduit tout simplement par un arrêt complet du service. De plus, l'apparition d'une panne affecte de la même façon toutes les classes de service, de sorte qu'elles subissent toutes une dégradation de performance. Cependant, en utilisant plusieurs chemins disjoints entre la source et la destination, comme le fait *DiffServ**, l'apparition d'une panne sur l'une quelconque des routes n'entraîne pas la suspension totale du service mais simplement une diminution de la largeur de bande disponible. Dès lors, le mécanisme de contrôle de congestion de DiffServ* se déclenche et ajuste le débit de transmission par classe de façon à protéger les trafics prioritaires contre toute dégradation de performance. *DiffServ** permet ainsi de garantir la qualité de service des trafics prioritaires et ce, même en cas de panne dans la couche optique. Avec *DiffServ**, la dégradation de performance des trafics prioritaires peut seulement être causée par une interruption complète du service, interruption qui ne peut avoir lieu que si toutes

les routes vers la destination tombent en panne au même moment, chose assez rare dans les réseaux filaires. La mise en œuvre de *DiffServ** nécessite donc une répartition de la charge entre plusieurs canaux et un mécanisme de détection de panne.

Par ailleurs, il faut souligner que le fonctionnement de *DiffServ** repose entièrement sur le concept de *lien logique*. Un *lien logique* entre deux nœuds est constitué d'un regroupement de plusieurs liens physiques entre ces 2 nœuds. En outre, un *lien logique* peut être porté par un chemin de plusieurs liens physiques.

2.2.2 DiffServ* dans les réseaux 802.11

Dans les réseaux sans fil, la qualité du canal n'est plus une variable binaire mais elle dépend de la valeur du rapport signal sur bruit (SNR). Dans cet environnement, lorsque le canal est mauvais, le débit de transmission n'est pas nul mais diminue en fonction du SNR. Plus le SNR diminue, plus le débit de transmission diminue aussi. De plus, la réduction de bande passante n'est plus due à une rupture de câble mais à des phénomènes multiples tels l'interférence, le bruit, la distance entre émetteur et destination, ou encore la vitesse de déplacement des stations. L'adaptation de *DiffServ** dans les réseaux sans fil requiert donc une modélisation du canal sans fil pour mettre en évidence la réduction de bande passante engendrée par les "pannes" et la détermination de plusieurs chemins indépendants entre la source et la destination. Dans un environnement sans fil, les "pannes" représentent la qualité du canal.

Les modifications requises pour implémenter *DiffServ** dans les réseaux sans fil nous ont permis de proposer une nouvelle architecture pour chaque nœud. En effet, cette variante de *DiffServ**, que nous nommons *Wireless-DiffServ**, est un mécanisme multi-couche qui interagit avec les couches IP, MAC et PHY afin de garantir la qualité de service des trafics hautement prioritaires dans un environnement sans fil variable. Les mécanismes de différenciation de la couche IP doivent recevoir des couches MAC et PHY une information portant sur l'état du média sans fil afin de réajuster l'ordonnancement des paquets de manière à limiter les dégradations de qualité de service des trafics prioritaires.

L'architecture proposée sur la figure 2.2 montre les composants principaux de chaque couche. Ainsi, le classificateur de paquet, les files d'attente, de même que l'ordonnanceur DiffServ sont mises en place avant le bloc de routage. Le bloc de routage, quant à lui, choisit l'interface de sortie vers laquelle le paquet doit être envoyé en fonction du routage sur les différents chemins vers la destination. Chaque nœud est équipé d'au moins trois interfaces radio. Il est à noter que *Wireless-DiffServ** utilise le même principe de fonctionnement que DiffServ c'est-à-dire la même classification de trafic et les mêmes politiques d'ordonnancement que DiffServ. Remarquons que l'architecture proposée peut être déployée avec une seule

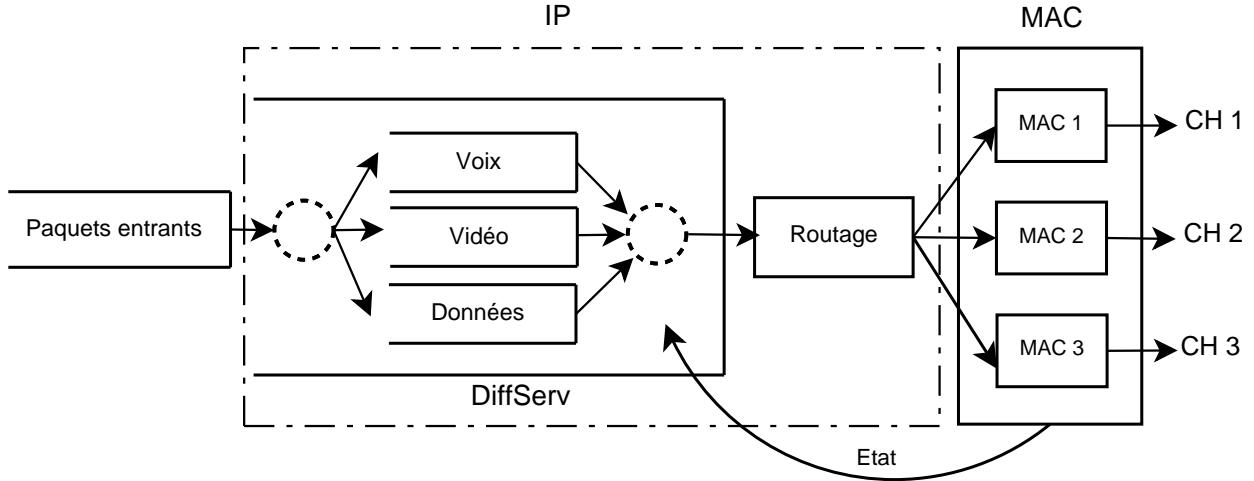


Figure 2.2 Protection DiffServ* dans un nœud mesh

interface. Dans ce cas, il n'y aurait aucune différence entre le modèle DiffServ standard et le *Wireless-DiffServ**. Dans les sections qui suivent, nous expliquons le modèle utilisé pour mettre en évidence la qualité du canal sans fil puis nous présentons le fonctionnement du module de routage.

2.3 Modélisation du canal sans fil

La modélisation du canal sans fil est la première étape du processus d'adaptation de DiffServ* dans les réseaux maillés sans fil. Dans les réseaux sans fil, il n'y a pas de "rupture de câble" à proprement parler puisque le support de transmission est l'air et ce dernier est toujours disponible. Une liaison sans fil peut cependant être interrompue lorsqu'il y a trop de bruit ou trop d'interférences entre l'origine et la destination. La modélisation du canal sans fil permet donc de trouver l'équivalence, dans les réseaux sans fil, à ce qui correspond à une "rupture de câble" dans un réseau filaire.

2.3.1 Description du canal sans fil

L'onde électromagnétique émise par une antenne s'atténue en fonction de la distance et est réfléchie ou réfractée par plusieurs surfaces avant d'être reçue par l'antenne de réception. On appelle *affaiblissement de parcours* (*path loss fading* en anglais), l'atténuation que subit le signal radio en fonction de la distance parcourue. Il se caractérise par une diminution de la puissance moyenne reçue en fonction de la distance entre les antennes d'émission et de réception. L'affaiblissement de parcours est également qualifié d'évanouissement à grande

échelle (*large scale fading* en anglais). Quelques modèles analytiques permettent d'évaluer cette atténuation. Le modèle le plus utilisé en pratique est le modèle de réflexion sur le sol [37] qui estime la puissance moyenne reçue P_r par l'équation :

$$P_r = P_t G_t G_r \left(\frac{h_t^2 h_r^2}{d^4} \right) . \quad (2.4)$$

où P_t est la puissance de transmission, G_t et G_r sont les gains des antennes émettrice et réceptrice, h_t et h_r la hauteur des antennes, d est la distance qui sépare les 2 antennes.

Lorsque l'onde est réfléchie ou réfractée, cela crée plusieurs copies du même signal qui arrivent toutes au récepteur. C'est le phénomène de multi-trajets (*multipath* en anglais). La somme vectorielle de toutes ces copies entraîne une variation temporelle de l'enveloppe du signal puisque toutes les répliques ne sont pas reçues au même instant. On l'appelle évanouissement à petite échelle (*small scale fading* en anglais). Les évanouissements à petite échelle sont généralement modélisés par le modèle de Rice ou celui de Rayleigh. Dans un canal de Rice, on suppose que l'onde reçue est la somme de toutes les composantes réfléchies et de la composante de ligne directe. Le modèle de Rayleigh quant à lui, considère qu'il n'y a pas de composante de ligne directe entre les antennes. La composante de ligne directe apparaît généralement lorsque les antennes d'émission et de réception sont distantes de quelques dizaines de mètres. Ainsi, le modèle de Rice convient mieux aux études menées dans des environnements internes (appartements, immeubles, bureaux, école, etc). Le canal de Rayleigh est très utilisé en pratique, notamment dans les environnements externes car les antennes d'émission et de réception sont assez éloignées l'une de l'autre, des centaines de mètres. Dans la suite, nous travaillons avec le canal sans fil de Rayleigh dont nous présentons les propriétés dans la section 2.3.3.

2.3.2 Paramètres physiques

Le canal sans fil variant dans le temps se caractérise par quelques paramètres :

La fréquence Doppler F_D

La mobilité des antennes d'émission et de réception d'une part, et d'autre part le phénomène d'évanouissement multi-trajets, engendrent l'étalement fréquentiel du signal reçu dans la bande $[f_c - F_D, f_c + F_D]$ où F_D est la fréquence Doppler maximale. La fréquence Doppler f_d est le décalage entre la fréquence émis et celle du signal reçu.

$$f_d = \left(\frac{v}{c} \right) f_c \cos \theta \quad (2.5)$$

où f_c est la fréquence porteuse du signal transmis, v est la vitesse du mobile, c est la vitesse de la lumière et θ est l'angle que fait le signal reçu avec la direction de déplacement du mobile. La fréquence Doppler est maximale lorsque $\cos \theta = 1$ et

$$F_D = \frac{f_c v}{c} \quad (2.6)$$

Le temps de cohérence T_c

Le temps de cohérence T_c d'un canal sans fil est l'intervalle de temps au cours duquel le canal change [46]. Il est inversement proportionnel à la fréquence Doppler maximale F_d

$$T_c \approx \frac{9}{16\pi F_d} \quad (2.7)$$

Par ailleurs, T_c permet de caractériser les évanouissements à petite échelle du canal. Un canal est dit à évanouissement lent si $T_s \ll T_c$ tandis qu'un canal est dit à évanouissement rapide si $T_s \gg T_c$, T_s étant la durée du signal émis. Pendant un intervalle de temps équivalent à T_c , les évanouissements sont fortement corrélés entre eux. Dans la pratique, il est cependant recommandé de réestimer le canal après un intervalle de temps équivalent à $T_c/10$ [46].

Bande de cohérence W_c

La bande de cohérence W_c est inversement proportionnelle à l'étalement temporel des retards des multi-trajets. C'est l'intervalle de fréquence dans lequel la réponse fréquentielle du canal change. Un canal est dit *plat* ou *non sélectif en fréquence* lorsque la largeur de bande W_s du signal est très petit par rapport à W_c . Dans le cas contraire, le canal est *sélectif en fréquence*. Puisque nous nous intéressons aux distorsions temporelles du signal reçu, nous considérons dans la suite du projet que le canal étudié est non sélectif en fréquence et à évanouissement lent.

2.3.3 Caractéristiques du Canal de Rayleigh

Un avantage du canal de Rayleigh est que la densité de probabilité de l'enveloppe du signal reçu et la densité de probabilité de la puissance instantanée reçue suivent des lois connues [46]. L'amplitude du signal instantané reçu suit la loi de Rayleigh

$$f(s) = \frac{s}{\sigma^2} \exp\left(-\frac{s^2}{2\sigma^2}\right), \quad s \geq 0, \quad (2.8)$$

où s est l'amplitude du signal reçu et σ^2 , la puissance moyenne du signal diffus reçu à travers le canal.

La puissance instantanée P du signal reçu, quant à elle, suit une distribution exponentielle $f(P)$ de moyenne σ^2

$$f(P) = \frac{1}{\sigma^2} \exp\left(-\frac{P}{\sigma^2}\right). \quad (2.9)$$

Soit γ le SNR instantané reçu défini par :

$$\gamma = \frac{P}{N_0 B} \quad (2.10)$$

où N_0 (en W/Hz) est la densité spectrale du bruit et B (en Hz) la bande passante du canal. On déduit de l'équation (2.9) que la densité de probabilité du SNR suit une distribution exponentielle de moyenne γ_0 .

$$f(\gamma) = \frac{1}{\gamma_0} \exp\left(-\frac{\gamma}{\gamma_0}\right), \quad \gamma \geq 0 \quad (2.11)$$

où $\gamma_0 = \sigma^2/N_0 B$ est le SNR moyen reçu.

2.3.4 Adaptation de lien

Comme nous l'avions dit dans la section 1.3.2.3, l'adaptation de lien est un procédé de la couche PHY qui permet de varier le taux de transmission des paquets en fonction des conditions du canal. La variation du taux de transmission suppose aussi une variation de la bande passante disponible.

2.3.4.1 Principe de l'AMC

L'AMC adapte dynamiquement le débit de transmission des paquets en fonction de la qualité du canal. Cette dernière est évaluée par le SNR instantané reçu. La figure 2.3 montre que pour obtenir une probabilité d'erreur par bit (BER) inférieure à 10^{-4} le SNR reçu doit être supérieur à 8 dB avec la modulation BPSK, ou à 12 dB avec la 16-QAM ou encore à 16 dB avec la 64-QAM. Si le système n'utilise qu'une seule modulation, par exemple 64-QAM, le BER sera trop élevé ($> 10^{-4}$) en dessous du seuil de 16 dB. Cependant, en découplant l'axe des SNR en intervalles disjoints, il est possible de maintenir la probabilité d'erreur par bit constante en utilisant différentes modulations. C'est le principe de l'AMC.

Nous utilisons, dans ce projet, la norme 802.11g qui définit 8 taux de transmission obtenus en combinant divers schémas de modulation et de codage (voir tableau 1.2). Le principe de l'AMC peut aussi être étendu à la probabilité d'erreur par paquet (PER) de façon à obtenir

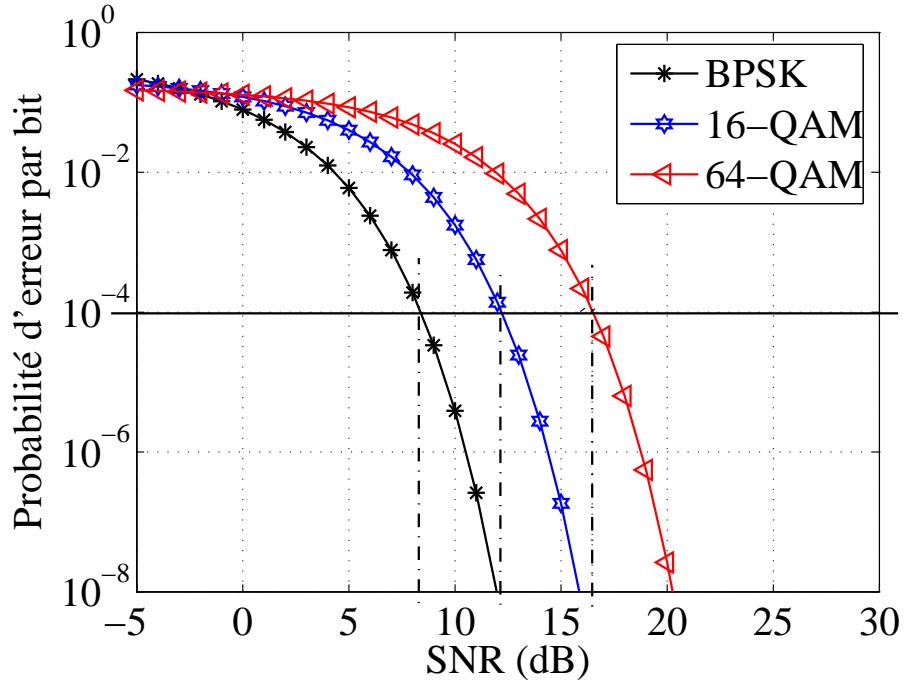


Figure 2.3 BER en fonction du SNR pour différentes techniques de modulation

un PER constant en faisant varier la modulation et le codage en fonction du SNR reçu et de la taille du paquet à transmettre.

Nous expliquons dans la section suivante les combinaisons retenues pour l'AMC et nous invitons le lecteur à se référer aux travaux de Gnassou [15] pour comprendre la procédure utilisée pour créer les plages de SNR disjointes.

2.3.4.2 Partitionnement du SNR

Dans cette étude, le canal de Rayleigh est supposé non sélectif en fréquence et à évanouissement lent. La variation du SNR instantané reçu γ est modélisée par une chaîne de Markov discrète [15]. Chaque état de la chaîne correspond à une plage de valeurs de SNR à l'intérieur de laquelle la probabilité d'erreur par paquet (PER) est constante et égale à un taux d'erreur cible P_0 . Dans les travaux de Gnassou [15], le SNR est partitionné en 7 intervalles correspondant aux états de la chaîne de Markov (Figure 2.4). Le tableau 2.1 montre la modulation et le codage utilisés dans chaque état ainsi que le taux de transmission correspondant.

Nous pouvons observer que les états 0 et 1 de la chaîne de Markov utilisent les mêmes modulation et codage. Cependant ces états sont différents puisque la probabilité d'erreur dans l'état 0 est élevée et n'est pas égale à P_0 . L'état 0 correspond donc au cas où le canal est très mauvais.

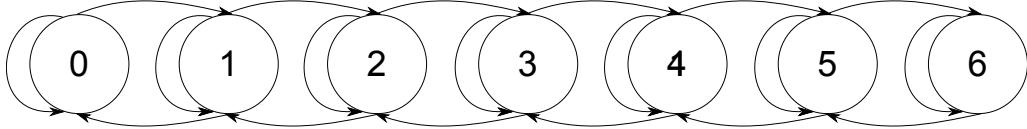


Figure 2.4 Chaîne de Markov à états finis modélisant les conditions du canal

Tableau 2.1 Correspondance entre états de la chaîne de Markov et taux de transmission

État de la CM	Modulation et codage	Débit (Mbps)
0	BPSK 1/2	6
1	BPSK 1/2	6
2	QPSK 1/2	12
3	QPSK 3/4	18
4	16QAM 1/2	24
5	16QAM 3/4	36
6	64QAM 3/4	54

Comme nous le disions plus haut, la chaîne de Markov est composée de 7 états. L'état k est choisi si le SNR instantané $\gamma(t) \in [\gamma_k, \gamma_{k+1}]$. Ainsi, la probabilité d'être dans l'état k est donnée par l'équation :

$$\pi_k = \int_{\gamma_k}^{\gamma_{k+1}} \gamma(t) dt \quad (2.12)$$

Sachant que le SNR est exponentiellement distribué (équation 2.11), on en déduit que :

$$\pi_k = \exp\left(-\frac{\gamma_k}{\gamma_0}\right) - \exp\left(-\frac{\gamma_{k+1}}{\gamma_0}\right). \quad (2.13)$$

Le vecteur $\boldsymbol{\pi}$ est le vecteur de probabilité stationnaire de la chaîne de Markov. Nous utilisons ce vecteur pour générer l'état du médium sans fil à un instant quelconque au cours des simulations.

Les plages de SNR pour chaque état ont été calculées à partir de la longueur des paquets, du SNR moyen reçu, de la fréquence Doppler et du taux d'erreur moyen acceptable par paquet. La procédure est détaillée dans [15].

2.4 Sélection de l'interface de sortie

Dans l'architecture que nous proposons sur la figure 2.2, l'opération de routage doit non seulement choisir la route logique entre deux noeuds mais aussi, l'interface de sortie à utiliser. Puisque l'utilisation de *DiffServ** est fondée sur l'utilisation de routes physiquement disjointes

et indépendantes [39], l'algorithme de routage doit permettre de trouver plusieurs routes logiques disjointes entre deux nœuds. À cet effet, l'algorithme de routage multi-trajets *Multi-Path Optimized Link State Routing* (MP-OLSR) est utilisé. Lorsque l'ordonnanceur DiffServ retire un paquet d'une file d'attente et que la route vers la destination est calculée par le protocole de routage, il faut ensuite déterminer l'interface vers laquelle le paquet doit être envoyé puisque chaque nœud est équipé d'au moins trois interfaces. Plusieurs critères peuvent être utilisés pour choisir l'interface de sortie du paquet. Par exemple, nous pourrions envoyer aléatoirement les paquets sur les interfaces, ou bien choisir l'interface qui a le moins de paquets en attente, etc. Étant donné que les canaux vus par chaque interface sont variables, le critère de sélection doit donc, autant que possible, tenir compte de l'état du canal.

Ainsi, lorsqu'au moins deux interfaces sont vides et prêtes à transmettre des paquets, le paquet choisi par l'ordonnanceur est envoyé à l'interface disponible qui a le meilleur canal. Ensuite, lorsqu'une interface finit de transmettre un paquet, elle demande directement un paquet à l'ordonnanceur *Wireless-DiffServ*^{*}, sans tenir compte de l'état de son canal ; cela garantit que toutes les interfaces envoient des paquets pendant qu'il y en a en attente.

Dans ce chapitre, nous avons présenté le modèle *DiffServ*^{*} sur lequel se base ce projet. Nous avons mis en relief les aspects à prendre en considération en vue de l'adapter aux réseaux maillés sans fil. Il s'agissait notamment de modéliser les pannes dans un environnement sans fil, d'expliquer le principe du routage multi-trajets puis de décrire l'interaction entre la couche logique et la couche physique. Dans le prochain chapitre, nous présenterons la plate-forme de simulation et les résultats permettant de valider les implémentations du procédé d'adaptation de lien et du module DiffServ.

CHAPITRE 3

IMPLÉMENTATIONS ET PERFORMANCES DE DIFFSERV ET DE L'AMC

La simulation du modèle *Wireless-DiffServ** nécessite l’implémentation du module DiffServ et d’un “générateur de pannes”. Rappelons que le module DiffServ comprend un classificateur de paquets, des files d’attente et un ordonnanceur utilisant la politique de la pondération cyclique ou WRR. Les pannes du canal sans fil sont caractérisées par ses différents états d’évanouissements. L’utilisation d’un procédé d’adaptation de modulation et de codage (AMC) permet donc de connaître la capacité d’un lien dans chaque état d’évanouissement et de savoir si ce dernier est en panne ou non.

Dans ce chapitre, nous utilisons le simulateur de réseau à évènements discrets NS-2 (*Network Simulator* version 2.34) [13] pour vérifier, par simulation, l’implémentation du module DiffServ puis celle de l’AMC. Après avoir défini les classes de trafic prises en charge par DiffServ, la validation portera sur l’étude de leurs performances sur un lien.

3.1 Modèles de trafic et métriques de performance

Dans cette étude, nous avons défini trois classes de service dans le module DiffServ :

- la classe EF : Elle regroupe les trafics de voix sur IP (VoIP). C’est la classe qui a la plus grande priorité ;
- la classe AF : Moins prioritaire que la classe EF, elle regroupe les trafics de vidéo sur demande, et
- la classe BE est la moins prioritaire de tous. Elle comprend le trafic des applications de transfert de données.

Tous les trafics sont de type ON-OFF. Les périodes moyennes d’activité et de silence d’une source de trafic sont exponentiellement distribuées de moyenne 500 msec. Les caractéristiques d’une source de chaque trafic sont présentées dans le tableau 3.1.

Tableau 3.1 Paramètres des 3 classes de trafic.

Paramètres	VoIP	Vidéo	Données
Débit par source (Kbps)		64	
Temps d’activité (ms)		500	
Cycle d’activité (%)		50	
Taille des paquets (octets)	80	variable	1440

Les paramètres du trafic de VoIP correspondent à ceux du codec G.711 [31]. La taille des paquets vidéo est uniformément distribuée entre 100 octets et 1500 octets. Soulignons que tous ces trafics sont modélisés par des processus ON-OFF de mêmes caractéristiques afin de mesurer les performances du réseau en fonction du nombre de sources actives, donc en fonction de la charge.

Au cours des simulations, nous évaluons les métriques de qualité de service suivantes :

- le taux de paquets perdus : c'est le rapport du nombre de paquets rejetés et perdus, après un nombre maximal de retransmission, divisé par le nombre total de paquets envoyés par le nœud source.
- le délai moyen par paquet : c'est le temps moyen requis pour que la destination reçoive le paquet transmis par la source. Ceci inclut le temps de séjour dans les files d'attente et les temps d'overhead.
- la gigue moyenne : Selon la définition de l'IETF, la gigue est la variation du délai de deux paquets consécutifs. Elle se calcule par l'équation (3.1) dans laquelle d_i est le délai du $i^{\text{ème}}$ paquet et n le nombre total de paquets reçus.

$$\frac{\sum_{i=2}^n \|d_i - d_{i-1}\|}{n - 1} \quad (3.1)$$

La durée d'une simulation est 300 sec. Les sources commencent à envoyer du trafic à partir de la 20^{ème} sec et s'arrêtent à la 120^{ème} sec. Ceci permet de s'assurer que les nœuds ont bien établi leur relation de voisinage et que tous les paquets générés sont reçus par la destination avant la fin de la simulation. Chaque simulation est répétée trente (30) fois afin de calculer un intervalle de confiance à 95%.

3.2 Topologie étudiée

Nous utilisons le réseau de la figure 3.1 pour vérifier le fonctionnement du module DiffServ et l'implémentation de l'AMC dans le canal sans fil de Rayleigh. Sur cette figure, la source et la destination communiquent directement par le biais d'une seule interface. Il existe entre les deux nœuds un lien logique constitué d'une seule route physique d'une longueur de un saut. La structure interne de l'interface de sortie du nœud source est présentée sur la figure 3.2 qui représente aussi l'implémentation standard de DiffServ dans un nœud. Implémenté entre les couches IP et MAC, l'ordonnanceur DiffServ contrôle les trois (3) files d'attente virtuelles de l'interface physique. Les files virtuelles sont mesurées en octets et ont, chacune, une capacité maximale de 50 paquets de 1520 octets.

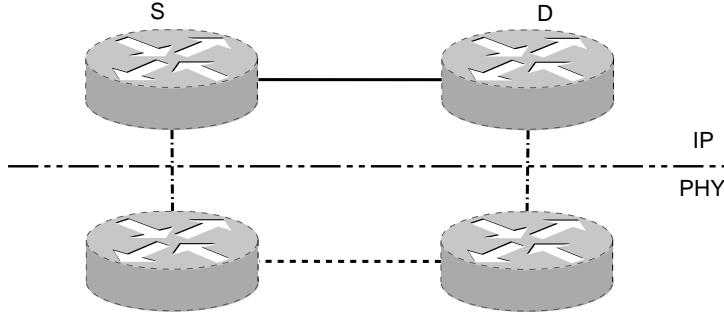


Figure 3.1 Topologie étudiée

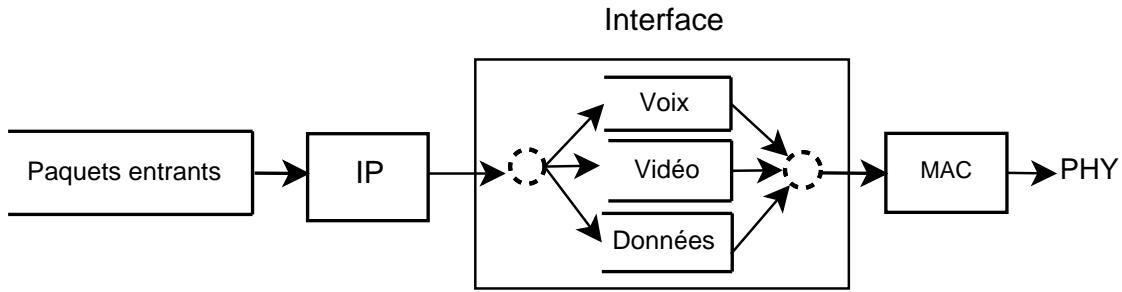


Figure 3.2 Déploiement standard de DiffServ dans un noeud sans fil

3.3 Vérification de l'implémentation de DiffServ

Nous avons mentionné dans la section 2.1.2 que la taille des paquets doit être prise en compte lors de l'étude de l'ordonnanceur WRR. Afin de vérifier l'implémentation de l'ordonnateur DiffServ, nous allons déterminer les rapports de proportionnalités qui doivent exister entre les poids des différentes classes afin de respecter la priorité de service de chacune d'elle ; ceci revient donc à déterminer la manière d'attribuer les poids de chaque classe de façon à garantir un traitement prioritaire au trafic VoIP, assigné à la classe EF. À cet effet, nous procédons par simulation en analysant les performances observées lorsqu'on fait varier les poids du WRR. Toutes les mesures, perte, délai et gigue, sont réalisées dans un canal supposé constant et transmettant les paquets au taux fixe de 54 Mb/s. C'est donc un canal ayant un seul état car les variations du canal sans fil ne sont pas encore modélisées dans le simulateur donc l'AMC n'est pas encore utilisée.

3.3.1 Règle d'attribution des poids

L'attribution des poids du WRR dépend de la taille moyenne des paquets par trafic. Soit L_i et \mathcal{L}_i la taille moyenne avec et sans en-têtes d'un paquet de la classe i . Dans le standard

IEEE 802.11g, la somme de toutes les en-têtes d'un paquet est 69 octets. Le tableau 3.2 rappelle les valeurs de \mathcal{L}_i et L_i par classe de trafic.

Tableau 3.2 Taille moyenne des paquets (avec et sans en-têtes) par classe

Classe	EF	AF	BE
\mathcal{L}_i	80	800	1440
L_i	149	869	1509

Soit α_i le poids assigné au WRR pour servir la classe i . Soit θ_i le pourcentage de bits utiles servi par classe :

$$\theta_i = \frac{B_i}{B} \quad (3.2)$$

où B_i le nombre de bits utiles servis par classe pendant un cycle de WRR et B est le nombre total de bits utiles servis pendant un cycle de WRR. En supposant qu'il y a n classes de trafic, il s'ensuit que :

$$\theta_i = \frac{\alpha_i \mathcal{L}_i}{\sum_{k=1}^n \alpha_k \mathcal{L}_k} \quad (3.3)$$

Considérant les classes EF, AF et BE, pour avoir la priorisation désirée, il faut que :

$$\theta_{EF} > \theta_{AF} > \theta_{BE}$$

Autrement dit, il faut que :

$$\alpha_{EF} \mathcal{L}_{EF} > \alpha_{AF} \mathcal{L}_{AF} > \alpha_{BE} \mathcal{L}_{BE} \quad (3.4)$$

L'inéquation (3.4) revient donc à résoudre le système d'inéquations suivant :

$$\begin{cases} \frac{\alpha_{EF}}{\alpha_{AF}} > \frac{\mathcal{L}_{AF}}{\mathcal{L}_{EF}} \\ \frac{\alpha_{AF}}{\alpha_{BE}} > \frac{\mathcal{L}_{BE}}{\mathcal{L}_{AF}} \end{cases} \quad (3.5)$$

En remplaçant la longueur des paquets par leurs valeurs numériques, l'équation (3.5) devient

$$\begin{cases} \frac{\alpha_{EF}}{\alpha_{AF}} > 10 \\ \frac{\alpha_{AF}}{\alpha_{BE}} > 1.8 \end{cases} \Rightarrow \begin{cases} \alpha_{EF} > 10 \alpha_{AF} \\ \alpha_{AF} > 1,8 \alpha_{BE} \end{cases} \quad (3.6)$$

Puisque les poids α_i sont des entiers positifs non nuls, et vu qu'il n'y a aucune exigence de qualité de service pour la classe BE, nous choisissons d'attribuer à cette classe un poids unitaire en vue de simplifier la résolution du système d'inéquations (3.6). Ceci étant, l'équation (3.6) conduit à :

$$\begin{cases} \alpha_{EF} \geq 20 \\ \alpha_{AF} \geq 2 \\ \alpha_{BE} = 1 \end{cases} \quad (3.7)$$

Étant donné que la taille des paquets de la classe AF est variable, l'équation (3.7) ne fournit pas les bornes exactes des poids du WRR mais les valeurs qui nous rapprochent le plus des bornes recherchées. Nous ne pouvons que procéder par simulation pour déterminer les valeurs minimales exactes des poids du WRR.

3.3.2 Étude des systèmes 20-2-1 et 20-3-1

Le premier système de valeurs produit par l'équation (3.7) est le triplet ($\alpha_{EF} = 20$, $\alpha_{AF} = 2$, $\alpha_{BE} = 1$). En observant par simulation le fonctionnement de l'ordonnanceur Diff-Serv lorsqu'on utilise cette pondération, les figures 3.3 et 3.4 révèlent que c'est bien le trafic de voix qui est prioritaire. Cependant, il est suivi du trafic de données puis de vidéo. Rappelons que le trafic vidéo génère des paquets de taille variable d'où les valeurs minimales trouvées par le système d'équations (3.7) peuvent ne pas correspondre à la priorisation désirée. Par contre, la figure 3.5, qui présente l'évolution de la gigue en fonction du nombre de sources actives, montre une différenciation telle que souhaitée entre les trois classes. La pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$ n'est donc pas le système de valeurs minimales assurant la priorité de service du trafic vidéo par rapport au trafic de données.

Puisque le système 20-2-1 offre une meilleure qualité de service au trafic de données plutôt qu'au trafic vidéo, en augmentant d'une unité le poids du trafic vidéo, nous étudions à présent le triplet (α_{EF} , α_{AF} , α_{BE}) = (20, 3, 1). Les figures 3.6 à 3.8 sont les résultats issus de l'utilisation de cette pondération. La figure 3.6 montre une différenciation assez prononcée entre les pertes enregistrées par les trois classes de services (en particulier à partir de 80 sources par classe) tandis que la figure 3.7 montre une différenciation très faible entre les délais de ces trois classes. Bien que la différenciation soit minimale, elle montre tout de même que c'est le trafic de VoIP qui est prioritaire, suivi du trafic vidéo et enfin les données. Comme sur la figure 3.5, la figure 3.8 montre, encore une fois, une différenciation favorable à la voix, puis la vidéo et les données.

Il ressort de ces observations que le choix des poids du WRR est très déterminant dans l'évaluation des performances du lien. Par conséquent, nous pouvons dire que les valeurs

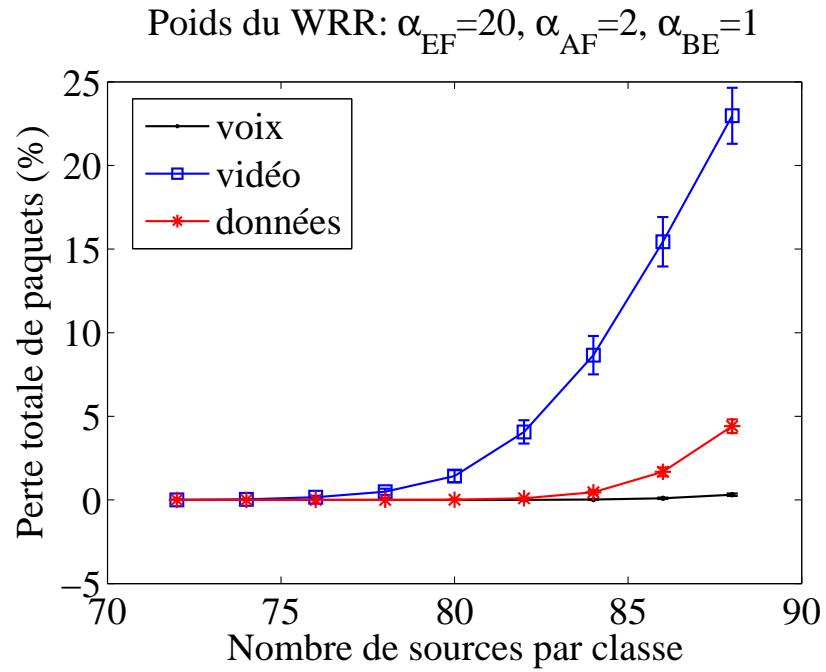


Figure 3.3 Taux de perte dans avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$

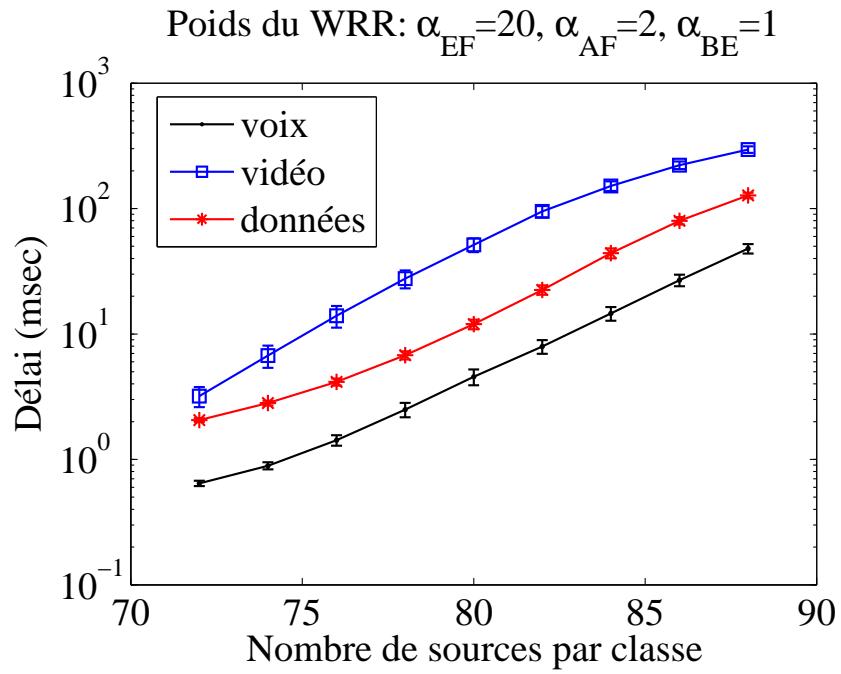


Figure 3.4 Délai moyen avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$

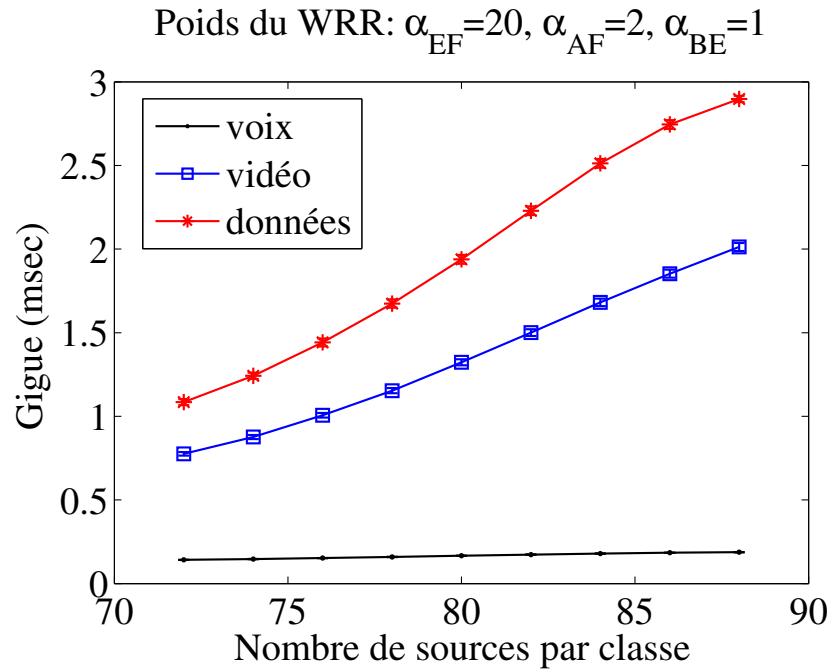


Figure 3.5 Gigue moyenne avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 2$ et $\alpha_{BE} = 1$

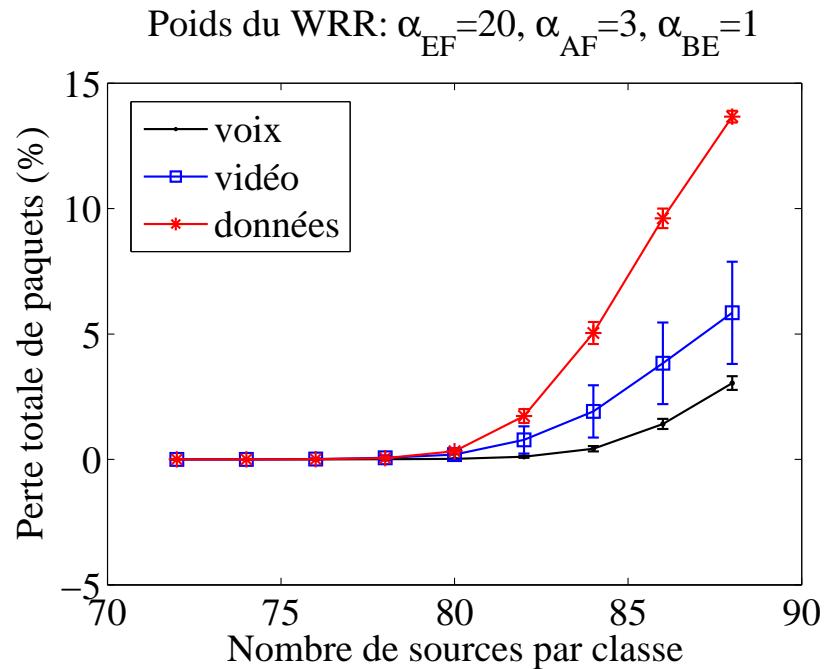


Figure 3.6 Taux de perte avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$

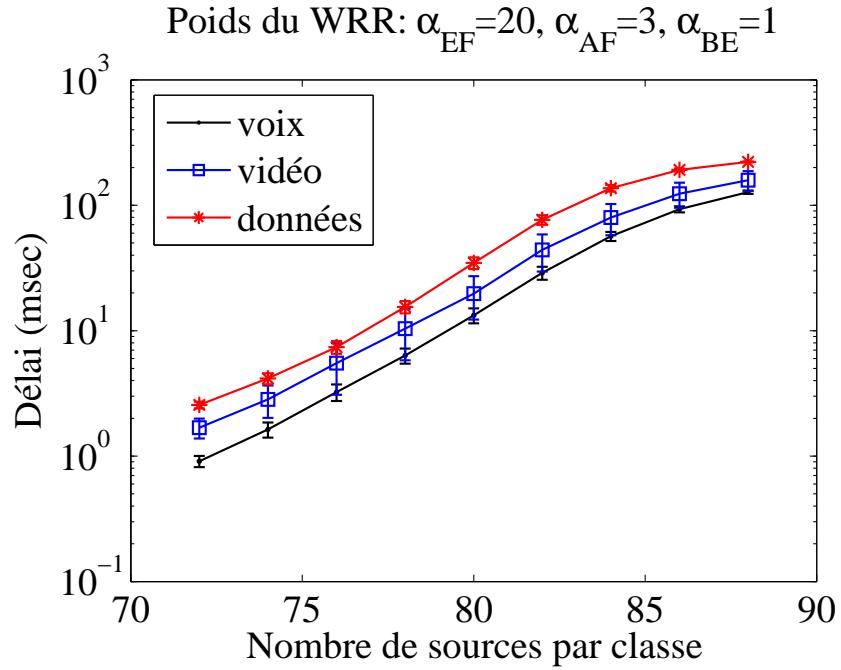


Figure 3.7 Délai moyen avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$

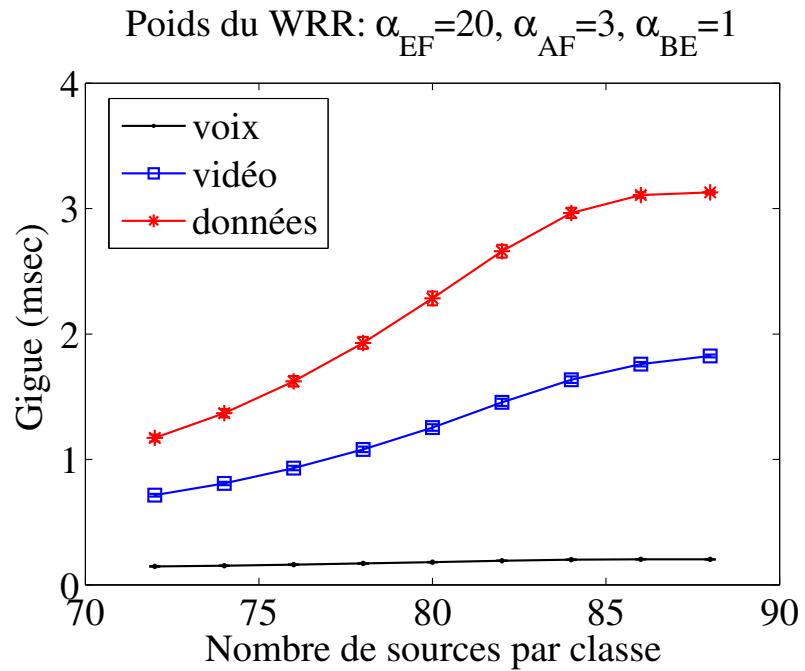


Figure 3.8 Gigue moyenne avec la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$

minimales permettant de prioriser d'abord le trafic VoIP, puis le trafic vidéo et enfin, les données doivent vérifier le système d'inéquations défini comme suit :

$$\begin{cases} \frac{\alpha_{EF}}{\alpha_{AF}} \geq \frac{20}{3} \\ \frac{\alpha_{AF}}{\alpha_{BE}} \geq 3 \\ \alpha_{BE} = 1 \end{cases} \quad (3.8)$$

3.3.3 Étude du système 8-4-1

Dans cette section, nous voulons vérifier que le système réagit comme prévu à un changement de poids. Pour cela, nous changeons la pondération des files d'attente de façon à ne pas respecter au moins une des conditions imposées par le système d'inéquations (3.8) et nous observons de nouveau la qualité de service offerte par DiffServ. Les poids du WRR sont $\alpha_{EF} = 8$, $\alpha_{AF} = 4$ et $\alpha_{BE} = 1$ donc :

$$\begin{aligned} \frac{\alpha_{EF}}{\alpha_{AF}} &= \frac{8}{4} = 2 < \frac{20}{3} \\ \frac{\alpha_{AF}}{\alpha_{BE}} &= \frac{4}{1} = 4 > 3 \end{aligned} \quad (3.9)$$

Suite aux observations de la section 3.3.2, nous pouvons dire que, dans le cas actuel, la mesure du taux de perte et du délai moyen par classe sont suffisantes pour évaluer la qualité de la différenciation obtenue entre les trois classes.

Les figures 3.9 et 3.10 montrent la qualité de service des trois classes de trafic dans un canal sans fil constant d'une capacité de 54 Mb/s. Comme attendu, le trafic de voix est le moins favorisé des trois trafics tandis que le trafic vidéo a un taux de perte et un délai bien inférieurs aux performances du trafic de données. Avec cette pondération, le réseau est donc à dominance vidéo (AF), puis données (BE) et enfin voix (EF). En effet, après un cycle du WRR, l'ordonnanceur aura transmis 8 paquets de voix, 4 paquets de vidéo et 1 paquet de données soit :

- Pour EF : $8 \times 80 = 640$ octets
- Pour AF : $4 \times 800 = 3200$ octets
- Pour BE : $1 \times 1440 = 1440$ octets

En d'autres termes, le pourcentage de la bande passante utile allouée à chaque trafic est :

- Pour EF :

$$\frac{640 \times 100}{640 + 3200 + 1440} = 12,12\%$$

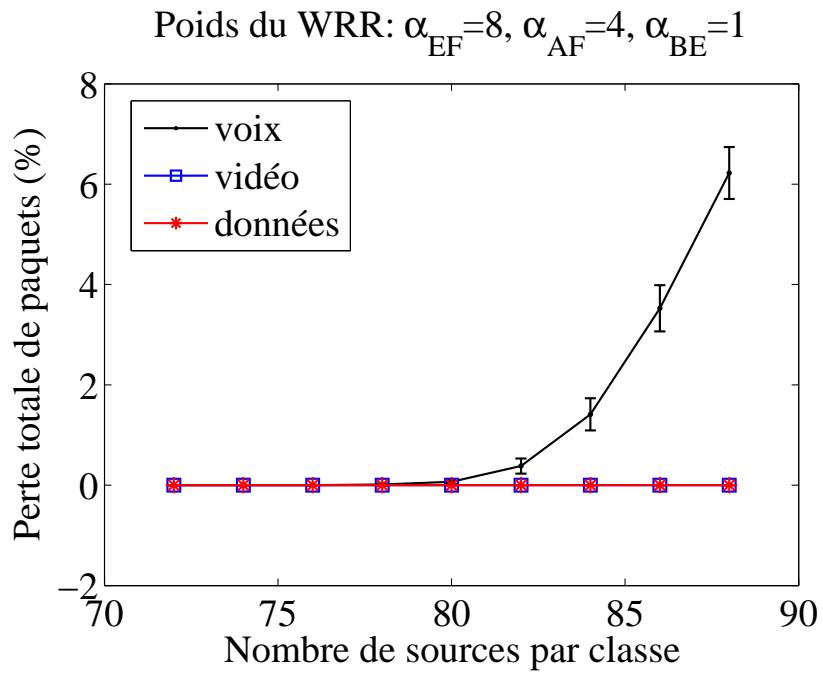


Figure 3.9 Perte moyenne avec la pondération $\alpha_{EF} = 8, \alpha_{AF} = 4$ et $\alpha_{BE} = 1$

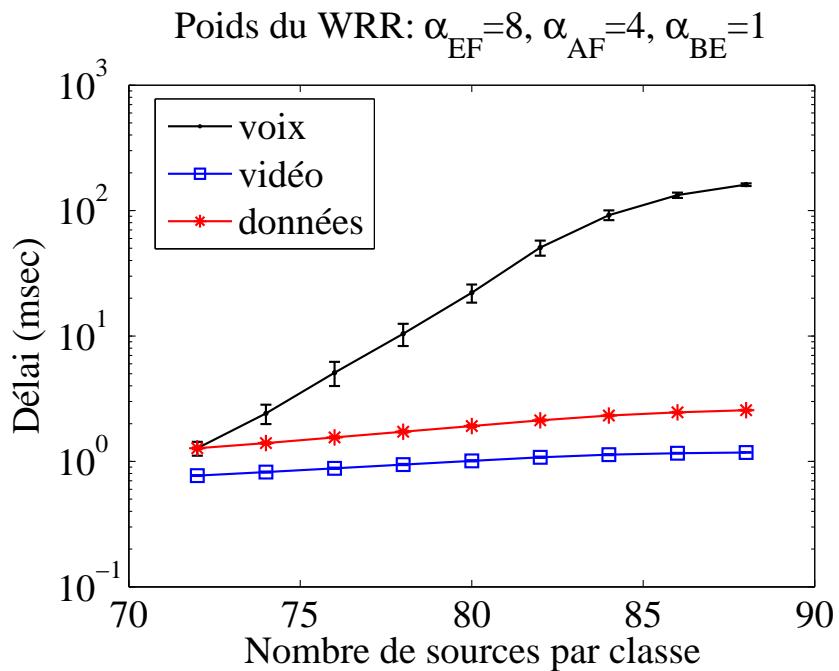


Figure 3.10 Délai moyen avec la pondération $\alpha_{EF} = 8, \alpha_{AF} = 4$ et $\alpha_{BE} = 1$

– Pour AF :

$$\frac{3200 \times 100}{640 + 3200 + 1440} = 60,61\%$$

– Pour BE :

$$\frac{1440 \times 100}{640 + 3200 + 1440} = 27,27\%$$

Cette répartition du débit utile explique alors le fort taux de perte et l'augmentation du délai observés pour le trafic de voix puisque la file d'attente de ce dernier est très peu servie par l'ordonnanceur.

Ces tests nous ont permis de mieux comprendre la manière dont l'ordonnanceur WRR gère les paquets de différentes tailles et d'observer la qualité de service de chaque classe de trafic. Aussi, l'analyse des résultats issus de ces simulations permettent de valider notre implémentation de l'ordonnanceur. Dans la suite de ce chapitre, nous utilisons la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$, $\alpha_{BE} = 1$.

3.4 Implémentation du module d'AMC dans un canal de Rayleigh

La section précédente a permis de valider l'implémentation de l'ordonnanceur DiffServ dans le simulateur NS-2 avec un débit de transmission constant. Dans cette section, nous allons vérifier l'implémentation du module d'AMC puis analyser la protection offerte par DiffServ en présence d'un canal variable. À cet effet, nous supposons dans un premier temps que le canal sans fil de Rayleigh est modélisé par une chaîne de Markov discrète à deux états. Nous faisons ensuite varier les probabilités d'état afin d'observer si le changement d'état correspond effectivement à un changement de débit de transmission : c'est le principe de fonctionnement de l'AMC. Dans un second temps, nous examinons le fonctionnement du module d'AMC dans un canal variable ayant les 7 états décrits à la section 2.1.

3.4.1 Validation d'un canal à 2 états

Pour valider l'implémentation du procédé d'adaptation de lien dans un canal sans fil de Rayleigh, nous allons considérer un canal à 2 états correspondant aux taux de transmission de 6 Mb/s et 54 Mb/s. Le taux d'erreur maximal par paquet est 10^{-4} . Nous supposons que dans l'état 0, le taux de transmission dans le canal est 6 Mb/s parce que le SNR instantané reçu est inférieur ou égal à γ_0 tandis que dans l'état 1, le SNR instantané reçu est supérieur à γ_0 et le taux de transmission est 54 Mb/s. Le débit moyen \bar{R} du canal peut être calculé par l'équation (3.10) dans laquelle R_i est le taux de transmission lorsque le canal se trouve dans l'état i et π la probabilité stationnaire d'être dans l'état 1 et de transmettre les paquets à 54

Mb/s.

$$\bar{R} = (1 - \pi)R_0 + \pi R_1 \quad (3.10)$$

$$\pi = P[\gamma > \gamma_0] \quad (3.11)$$

Le tableau 3.3 montre l'évolution du débit moyen du canal en fonction des probabilités

Tableau 3.3 Probabilité d'état et débit moyen

π	\bar{R} (Mb/s)
0	6
0,25	18
0,50	30
0,75	42
1,0	54

d'état. Comme on le constate, le débit moyen du canal augmente — donc la qualité du canal s'améliore — lorsque π augmente. Cette situation doit également se répercuter sur toutes les performances du réseau et doit s'observer par une diminution des pertes, du délai et de la gigue. Pour cela, nous mesurons les performances du réseau en fonction de la probabilité stationnaire π . Un tirage aléatoire est réalisé avant la transmission de chaque paquet afin de connaître l'état du canal et le taux auquel le paquet doit être envoyé. Le trafic total offert est égal à 3 Mb/s soit 32 sources par classe.

La figure 3.11 montre qu'il n'y a aucune perte dans le réseau car le volume de trafic que l'on désire écouler est très faible, 3 Mb/s. Cette valeur est choisie à dessein parce que si le lien envoie les paquets au taux de 6 Mb/s et qu'il est saturé, on noterait des pertes en majorité dues au remplissage des files d'attente puisque le taux d'erreur souhaité est très faible. Afin d'observer uniquement l'effet du canal sur les performances du réseau, nous avons donc choisi d'activer peu de sources. Les figures 3.12 et 3.13 confirment nos attentes et montrent la décroissance du délai et de la gigue lorsque la qualité du canal devient meilleure. De plus, quel que soit l'état du canal, le trafic de la classe EF est le mieux protégé de tous, suivi du trafic de la classe AF.

3.4.2 Effet d'un canal à plusieurs états

Puisque la sous-section précédente nous a permis de valider l'implémentation de l'AMC dans le canal sans fil, nous étudions ici le fonctionnement de DiffServ dans un canal à 7 états. Les différents états du canal sont présentés dans le tableau 2.1. Le débit moyen du canal est donné par l'expression (3.12) dans laquelle R_i est le taux de transmission des paquets lorsque

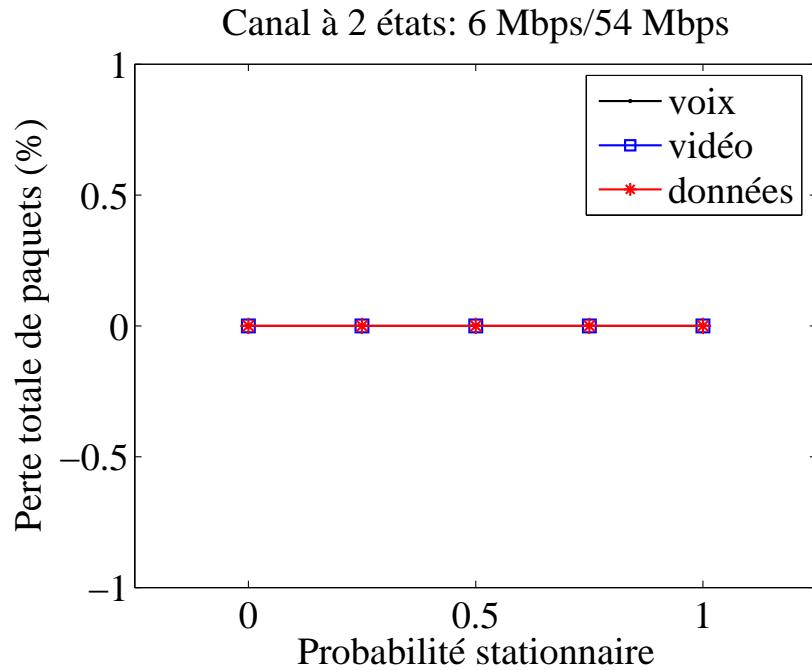


Figure 3.11 Variation des pertes avec DiffServ dans un canal à 2 états

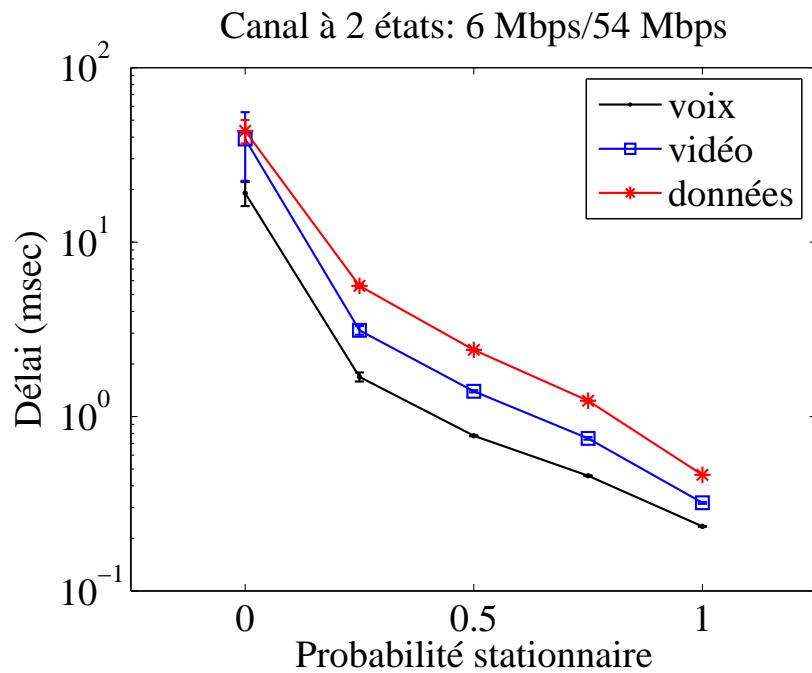


Figure 3.12 Variation du délai avec DiffServ dans un canal à 2 états

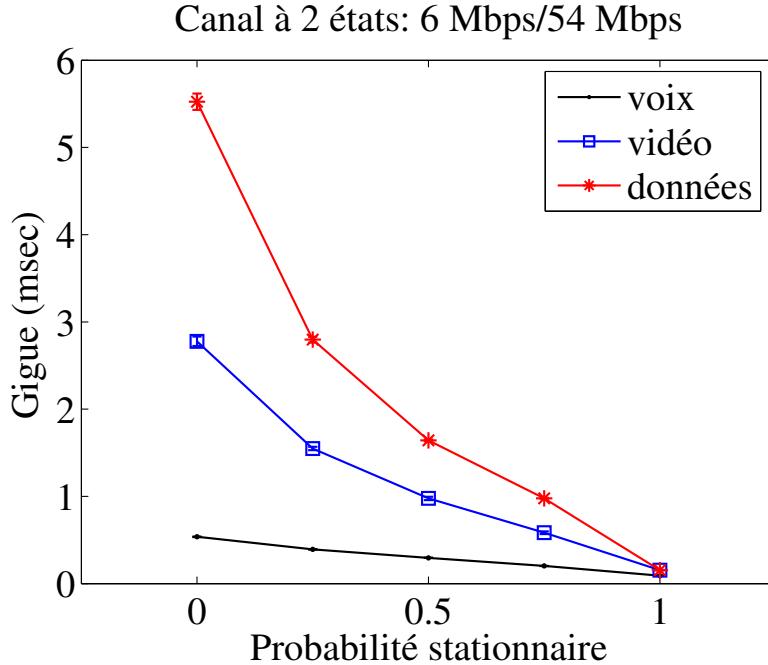


Figure 3.13 Variation de la gigue avec DiffServ dans un canal à 2 états

le canal est dans l'état i et π_i la probabilité stationnaire de l'état de i .

$$\bar{R} = \sum_{i=0}^6 R_i \pi_i \quad (3.12)$$

Les probabilités stationnaires utilisées ici correspondent à celles d'un canal de Rayleigh ayant un SNR moyen égal à 15 dB, un PER égal à 10^{-4} et une fréquence Doppler de 5 Hz. Les paquets peuvent être envoyés à des taux différents puisqu'un tirage aléatoire est fait avant chaque nouvelle transmission pour connaître l'état du canal. Pour un SNR moyen égal à 15 dB et un PER égal à 10^{-4} , le débit moyen du canal s'estime à environ 27,8 Mb/s.

Les figures 3.14, 3.15 et 3.16 montrent respectivement le taux de perte, le délai moyen et la gigue mesurés dans un canal variable de 7 états. L'observation de ces 3 figures révèle une diminution du nombre de sources actives par classe, diminution due à la réduction du débit moyen. En effet, on remarque que le réseau est saturé au delà de 64 sources par classe soit un débit total de 6 Mb/s alors que comparativement aux résultats obtenus dans un canal constant de 54 Mb/s (figures 3.6 et 3.7) le réseau n'est pas encore saturé avec 88 sources par classe soit un débit total de 8,45 Mb/s. Toutefois, quel que soit le débit offert et quel que soit l'état du canal, le trafic de la classe EF est le mieux protégé face aux 2 autres classes.

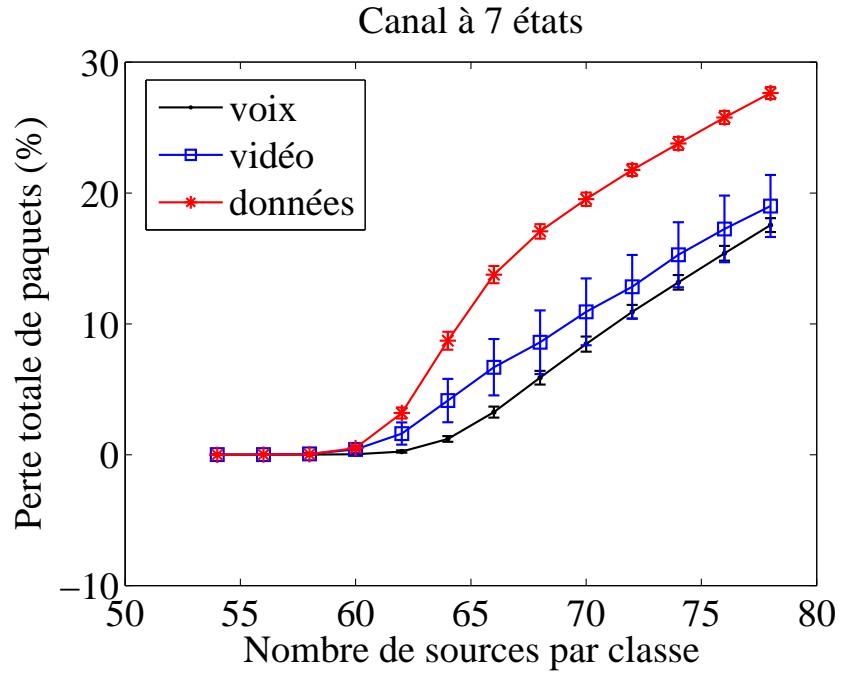


Figure 3.14 Taux de paquets perdus avec DiffServ dans un canal à 7 états

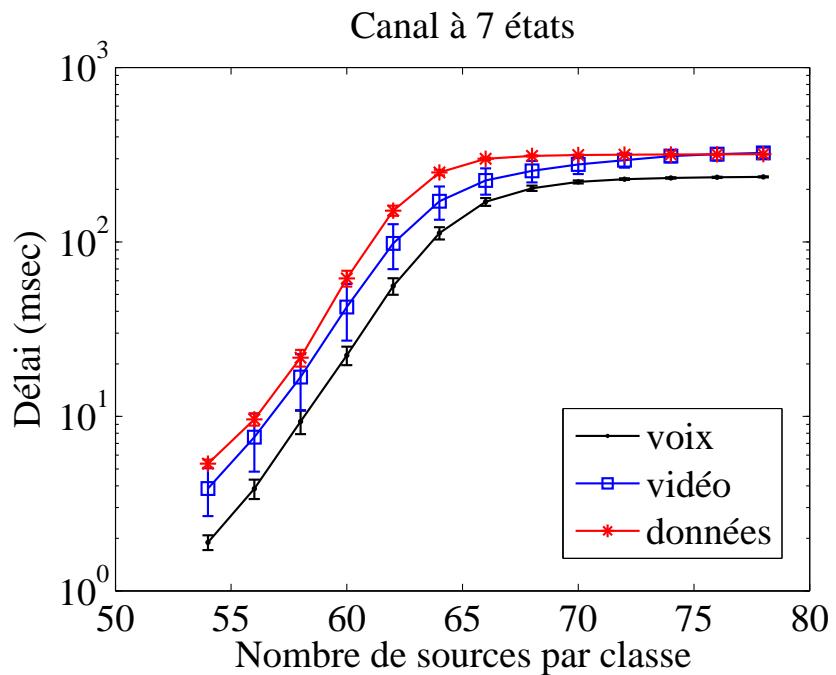


Figure 3.15 Délai moyen avec DiffServ dans un canal à 7 états

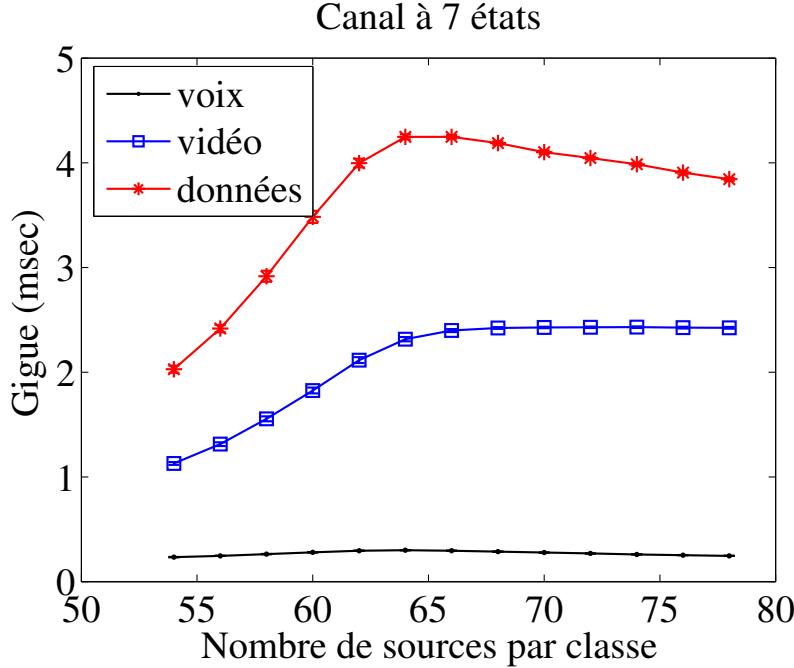


Figure 3.16 Gigue moyenne avec DiffServ dans un canal à 7 états

3.4.3 Impact des paramètres physiques

Dans cette section, nous étudions l'influence des paramètres tels que la fréquence Doppler, le SNR moyen et le taux d'erreur par paquet sur les performances du réseau. La topologie étudiée est toujours celle de la figure 3.1. Le canal sans fil est variable. Son état est estimé avant l'envoi de chaque paquet.

3.4.3.1 Incidence du SNR moyen

Dans le modèle étudié, la probabilité que le canal soit dans un état d'évanouissement est donnée par la probabilité stationnaire π_k d'être dans l'état k (équation 2.13). La probabilité d'état π_k dépend uniquement du SNR moyen γ_0 et des bornes de SNR γ_k . Les bornes γ_k quant à elles, sont fonctions de la taille des paquets à transmettre, du taux d'erreur par trame et du SNR moyen [15].

Afin d'étudier l'impact du SNR moyen sur les performances du réseau, nous évaluons le débit moyen du lien lorsqu'on fait varier le SNR moyen de 5 dB à 20 dB. Ceci permet de changer également les bornes γ_k et les probabilités d'état π_k . Le taux d'erreur souhaité par paquet est fixé à 10^{-4} . La figure 3.17 montre que l'augmentation du SNR moyen se traduit par une augmentation du débit moyen. Cela conduit donc à un canal de meilleur qualité,

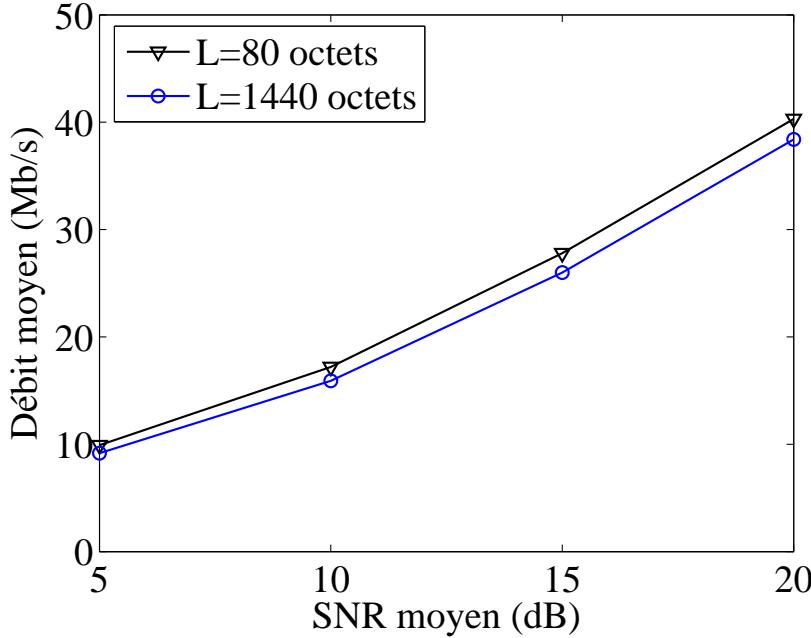


Figure 3.17 Débit moyen par canal en fonction du SNR moyen.

offrant la possibilité de transmettre les paquets à des taux élevés, réduisant ainsi le délai de transmission et le taux de perte.

Pour corroborer nos propos, nous mesurons alors les performances du réseau lorsqu'on fait varier le SNR moyen de 5 dB à 20 dB. Les probabilités d'état changent en fonction du SNR et le débit moyen peut augmenter ou diminuer. Le taux d'erreur désiré par paquet est 10^{-4} . Le nombre de sources actives par classe est fixé à 62 sources soit une charge de 1,98 Mb/s par classe. C'est donc un réseau moyennement chargé mais non saturé (voir Figures 3.14 et 3.15). Les poids des classes EF, AF et BE sont respectivement $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. Les figures 3.18 à 3.20 montrent respectivement l'évolution du taux de perte, du délai et de la gigue en fonction du SNR moyen. Comme on peut s'y attendre, les performances du réseau décroissent presque linéairement quand le SNR moyen augmente. Ces résultats, appuyés par ceux de Gnassou [15], confirment que l'augmentation du SNR moyen reflète un canal de meilleur qualité, donnant la possibilité de transmettre les paquets à des taux élevés.

3.4.3.2 Incidence du taux d'erreur par trame

Nous faisons varier le taux d'erreur souhaité par paquet de 10^{-4} à 10^{-1} . Le SNR moyen est 15 dB. La charge moyenne du réseau est fixée à 1,98 Mb/s par classe soit 62 sources actives par classe (réseau non saturé d'après la figure 3.15). La pondération du WRR est $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. Les figures 3.21 à 3.23 montrent l'évolution du taux de perte,

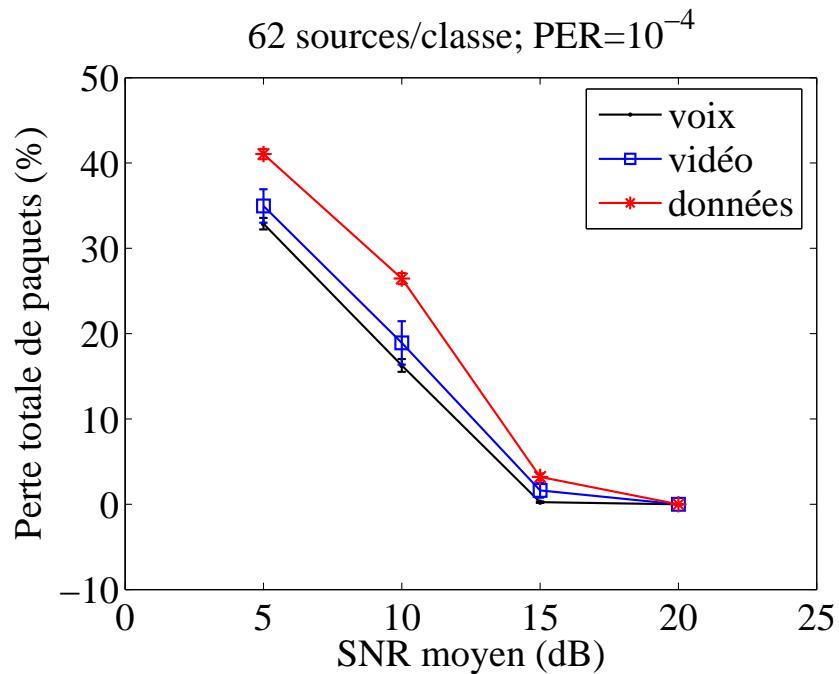


Figure 3.18 Perte moyenne en fonction du SNR moyen

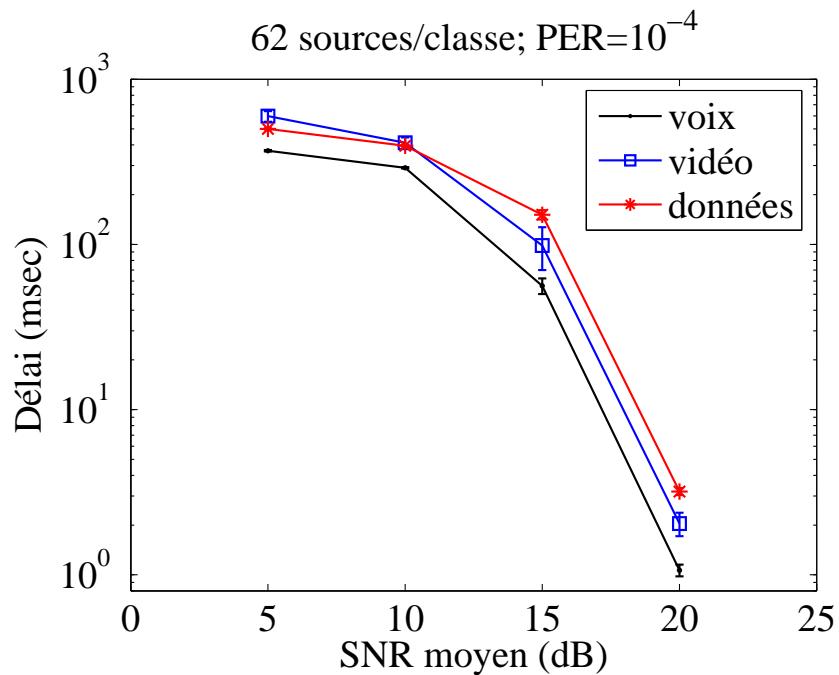


Figure 3.19 Délai moyen en fonction du SNR moyen

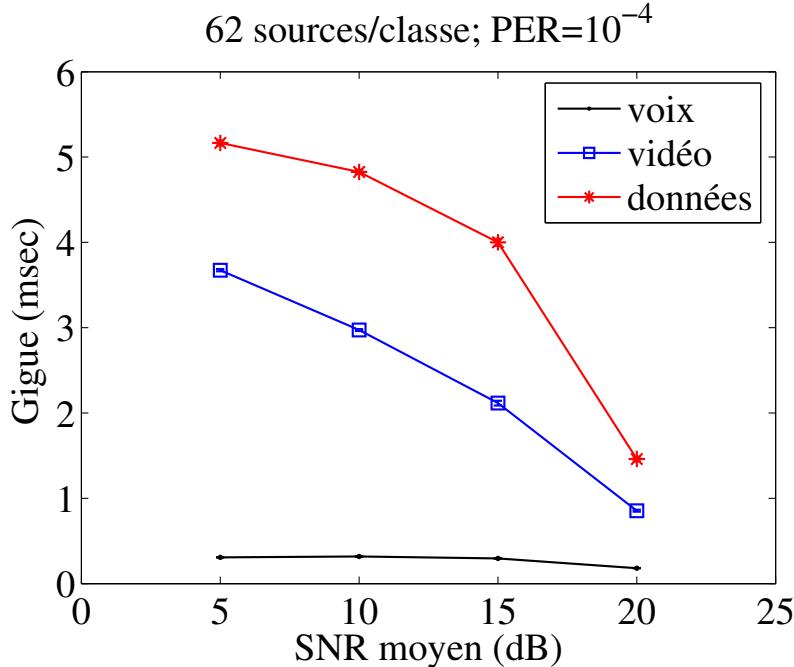


Figure 3.20 Gigue moyenne en fonction du SNR moyen.

du délai et de la gigue en fonction du PER. Il est intéressant de remarquer sur ces figures que les métriques décroissent lentement de 10^{-4} à 10^{-2} avant d'augmenter brusquement lorsque le PER est fixé à 10^{-1} . Ceci peut s'expliquer par le fait qu'en imposant un PER très faible (10^{-4} par exemple), la détection des paquets corrompus devient beaucoup plus sélective. Dans ce cas, l'AMC privilégie l'utilisation des schémas de modulation les plus robustes (BPSK et QPSK), ce qui entraîne l'utilisation de faibles taux de transmission, et par conséquent une augmentation du délai moyen. À faible PER, l'augmentation du taux de perte est principalement due aux pertes liées au remplissage des files d'attente, étant donné que les paquets sont envoyés à des taux de transmission plus faibles. Lorsque le PER = 10^{-1} , le risque de corruption des paquets est plus élevé. Ceci entraîne comme conséquence immédiate l'augmentation des retransmissions et du délai moyen (voir section 1.3.1.2). Les paquets sont définitivement perdus lorsqu'ils ne sont pas correctement reçus au bout de n retransmissions.

Cette étude révèle d'une part qu'un compromis doit être fait entre la probabilité d'erreur désirée par trame (PER) et les performances du réseau et d'autre part que l'utilisation d'un PER = 10^{-2} serait un bon compromis dans ce cas. Soulignons que ces résultats concordent avec ceux de [15].

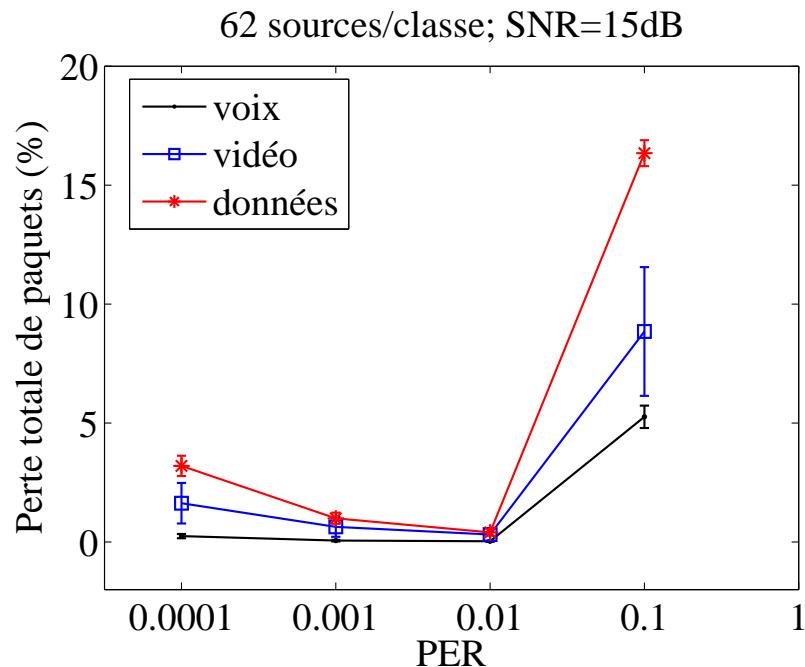


Figure 3.21 Perte moyenne en fonction du PER

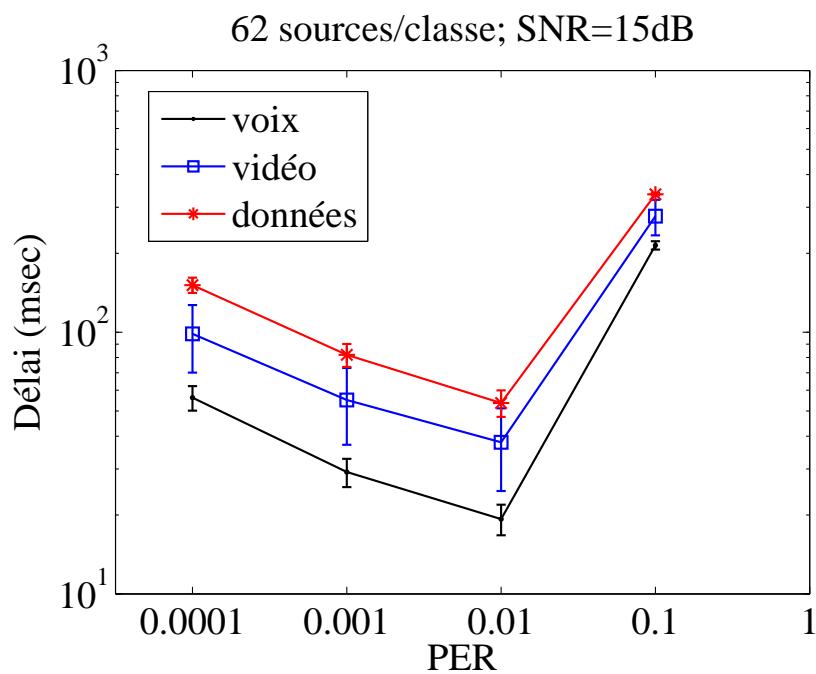


Figure 3.22 Délai moyen en fonction du PER

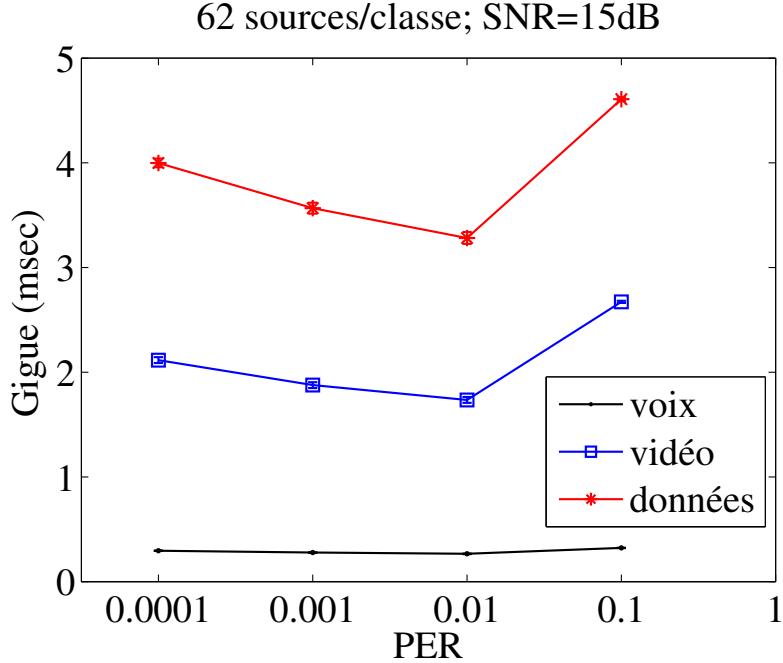


Figure 3.23 Gigue moyenne en fonction du PER

3.4.3.3 Impact de la fréquence Doppler

Nous avons vu dans la section 3.4.3.1 que la probabilité d'état π_k dépend uniquement du SNR moyen et du taux d'erreur par trame donc l'impact de la fréquence Doppler n'est pas aussi évident à observer que celui des autres paramètres.

La fréquence Doppler est fortement liée à la vitesse de variation du canal. Une fréquence Doppler faible signifie que le canal peut passer beaucoup de temps dans un état, par exemple un état d'évanouissement profond, tandis qu'une fréquence Doppler élevée signifie que le canal varie très rapidement et passe très peu de temps dans un état d'évanouissement profond. Rappelons que ces états se caractérisent par un faible taux de transmission, donc une augmentation du délai de transmission, et dans le pire des cas, une augmentation du taux de perte, lorsque le nombre maximal de retransmissions est atteint. Une fréquence Doppler trop faible diminue donc la qualité du lien et dégrade les performances du réseau alors qu'une fréquence Doppler élevée les améliore [9, 15, 18].

Puisque l'augmentation de la fréquence Doppler diminue la durée des évanouissements, il est donc très probable que les paquets soient transmis avec des taux de transmission différents, qui peuvent être élevés ou faibles. Cette situation est très semblable à notre utilisation du modèle des chaînes de Markov à états discrets. En effet, dans les résultats des sections précédentes, une estimation du canal est requise pour chaque paquet à transmettre. Cette

utilisation se rapproche alors de la modélisation d'un canal ayant une fréquence Doppler élevée. Il en découle que la variation de la fréquence Doppler n'a aucun impact sur ce modèle de canal.

Dans ce chapitre, nous avons implémenté l'ordonnanceur DiffServ entre les couches IP et MAC et nous avons observé la qualité de service offerte par ce dernier dans un canal sans fil variable. Les simulations réalisées ont permis de vérifier les implémentations des modules DiffServ et AMC dans le simulateur NS-2. Il s'agissait principalement d'observer le fonctionnement de l'ordonnanceur DiffServ sur un lien physique dont le taux de transmission peut être constant ou variable. Les variations de ce taux sont introduites par la mise en place du procédé d'adaptation de modulation et de codage.

Que le canal soit variable ou non, la priorité de service est toujours accordée au trafic de voix, suivi du trafic vidéo lorsque les poids de l'ordonnanceur WRR vérifient la règle d'attribution proposée dans ce chapitre car ces derniers (les poids) dépendent uniquement du nombre de bits transmis. Les résultats obtenus dans un canal variable ont montré d'une part que l'augmentation du SNR diminue les métriques de performance et améliore ainsi les performances du réseau et d'autre part qu'il y a un compromis à faire entre le taux d'erreur souhaité par paquet et les performances du réseau notamment en termes de perte et délai moyen. Il est important de remarquer que ces conclusions concordent parfaitement avec les résultats de [15]. Cependant, l'utilisation des probabilités stationnaires pour connaître l'état du canal est comparable à l'utilisation d'une fréquence Doppler très élevée. Il ne nous a donc pas été possible de vérifier l'impact d'une fréquence Doppler faible sur les performances du réseau. Le lecteur est invité à lire [15, 18] pour obtenir de plus amples informations. Dans les chapitres suivants, nous utiliserons un minuteur (ou "timer") pour simuler la vitesse de variation du canal sans fil étant donné que le modèle étudié est indépendant de la fréquence Doppler.

Le prochain chapitre présente l'implémentation de *Wireless-DiffServ** dans le simulateur NS-2. Cette fois-ci, l'ordonnanceur DiffServ est implanté dans la couche IP et le concept de *lien logique* est réintroduit.

CHAPITRE 4

PERFORMANCES DE WIRELESS-DIFFSERV*

L'implémentation du modèle *Wireless-DiffServ** requiert la mise en place du module Diff-Serv dans la couche IP, en plus de l'utilisation conjointe de l'AMC et des routes disjointes. Ce modèle utilise aussi le concept de *lien logique*. Un *lien logique* est constitué de 3 liens physiques entre deux noeuds adjacents. Afin de rendre la disjonction de routes possible en tout temps (quel que soit le nombre de sauts entre l'origine et la destination), et d'assurer l'indépendance des canaux utilisés, nous équipons chaque noeud de trois (3) interfaces configurées sur des fréquences non-chevauchantes. Chaque interface voit un canal sans fil variable. Les variations sont indépendantes d'un canal à l'autre. L'objectif principal de ce chapitre est de vérifier l'implémentation du modèle *Wireless-DiffServ*¹ sans l'adaptation des poids et de comparer ses performances à celles de quelques autres politiques de service. La première partie du chapitre porte sur l'étude de performance d'un réseau de deux (2) noeuds desservi par trois politiques d'ordonnancement : *Wireless-DiffServ**, DiffServ et FIFO tandis que la deuxième partie porte sur l'étude des performances d'un réseau de trois (3) noeuds et d'un réseau maillé sans fil de cinq (5) noeuds. Les poids de l'ordonnanceur de paquets du *Wireless-DiffServ** sont fixés à $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$ et restent constants pendant toute la durée des simulations.

4.1 Temps d'échantillonnage

Nous avons montré dans le chapitre précédent que la variation de la fréquence Doppler n'a aucune incidence sur le modèle du canal étant donné qu'un tirage aléatoire sur les probabilités stationnaires est effectué avant la transmission ou la retransmission d'un paquet. En conséquence, un tel procédé est équivalent à l'utilisation d'une fréquence Doppler très élevée. Comme les routeurs mesh sont fixes, la plage de variation de la fréquence Doppler demeure faible. On peut supposer que dans les réseaux maillés sans fil, la fréquence Doppler varie entre 5 Hz et 20 Hz [19].

Pour ralentir la vitesse de variation du canal dans notre étude, nous utiliserons une horloge, ou timer, dont la période déterminera l'intervalle de temps pendant lequel le canal sera supposé constant. Les propriétés temporelles du canal sans fil sont utilisées pour déterminer la période du timer. Rappelons d'abord ce qu'est le temps de cohérence d'un canal sans fil.

1. À partir de ce chapitre, le terme *DiffServ** est mis pour *Wireless-DiffServ**

Le *temps de cohérence*, noté T_c , est l'intervalle de temps au cours duquel l'amplitude du canal change considérablement [46]. Le temps de cohérence T_c est inversement proportionnel à la fréquence Doppler maximale F_d et se définit par

$$T_c \approx \frac{9}{16\pi F_d} \quad (4.1)$$

Les réalisations du canal à intervalle de T_c sont indépendantes. Cependant, après un intervalle de temps équivalent à $T_c/10$, l'amplitude du canal peut avoir suffisamment changé et doit être ré-évaluée [46]. C'est donc la fréquence à laquelle nous ferons l'échantillonnage du canal et plus tard la détection de pannes. Le canal est supposé constant pour tous les paquets transmis pendant une période de temps égale à $T_c/10$. De plus, on s'assure que pour des fréquences inférieures à 20 Hz, la période du timer ($T_c/10$) est inférieure au temps de transmission d'un paquet.

Dans le modèle *Wireless-DiffServ**, les nœuds sont équipés de trois (3) cartes réseau sans fil. Chaque carte voit un canal de Rayleigh plat à évanouissements lents. Toutefois, les propriétés statistiques du canal sont les mêmes pour toutes les interfaces c'est-à-dire que tous les canaux ont la même fréquence d'échantillonnage, le même SNR moyen et le même taux d'erreur cible. Nous utilisons dans la suite une fréquence de variation de 5 Hz, un SNR moyen de 15 dB et un PER moyen égal à 10^{-4} . Une fréquence de variation de 5 Hz signifie que l'état des canaux change toutes les 3.581 ms.

4.2 Performance sur un lien

La topologie présentée sur la figure 4.1 est constituée de deux routeurs mesh, une source (S) et une destination (D), équipés chacun de trois interfaces. Trois (3) configurations internes ressortent d'une telle architecture.

Dans un premier temps, nous voulons vérifier l'implémentation de *Wireless-DiffServ**. Le fonctionnement interne du nœud source est donc identique à ce qui est présenté dans la section 2.2.2 du chapitre 2. Tous les paquets sont mis en attente dans des files virtuelles situées dans la couche IP. Il y a une file d'attente virtuelle par type de trafic. L'ordonnanceur *DiffServ** décide la classe du paquet à servir mais c'est le module de routage qui décide l'interface de sortie vers laquelle ce paquet doit être envoyé en tenant compte de l'interface qui a le meilleur canal. Cette configuration est nommée *DiffServ** ou DS*. La capacité maximale de chaque file virtuelle est 150 paquets de 1520 octets.

La deuxième configuration possible est présentée sur la figure 4.2. Comme on peut le voir, la couche IP s'occupe uniquement de l'opération de routage puis choisit l'interface de sortie. Cette couche détermine les routes vers la destination puis envoie aléatoirement les paquets

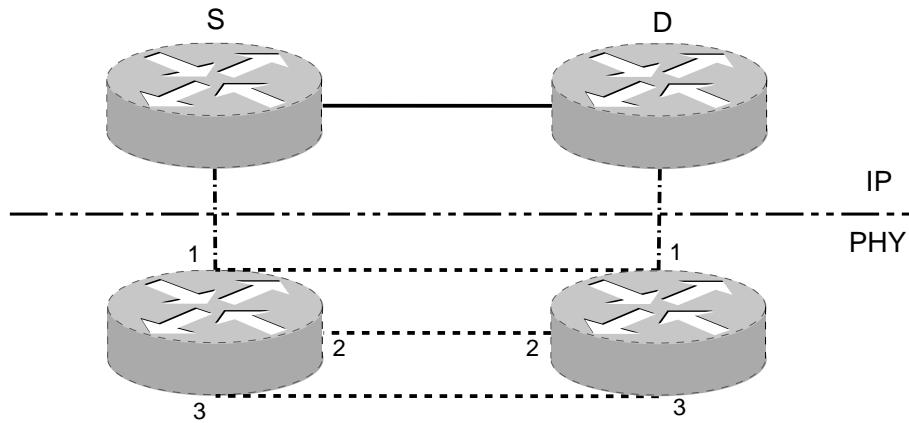


Figure 4.1 Lien logique et liens physiques.

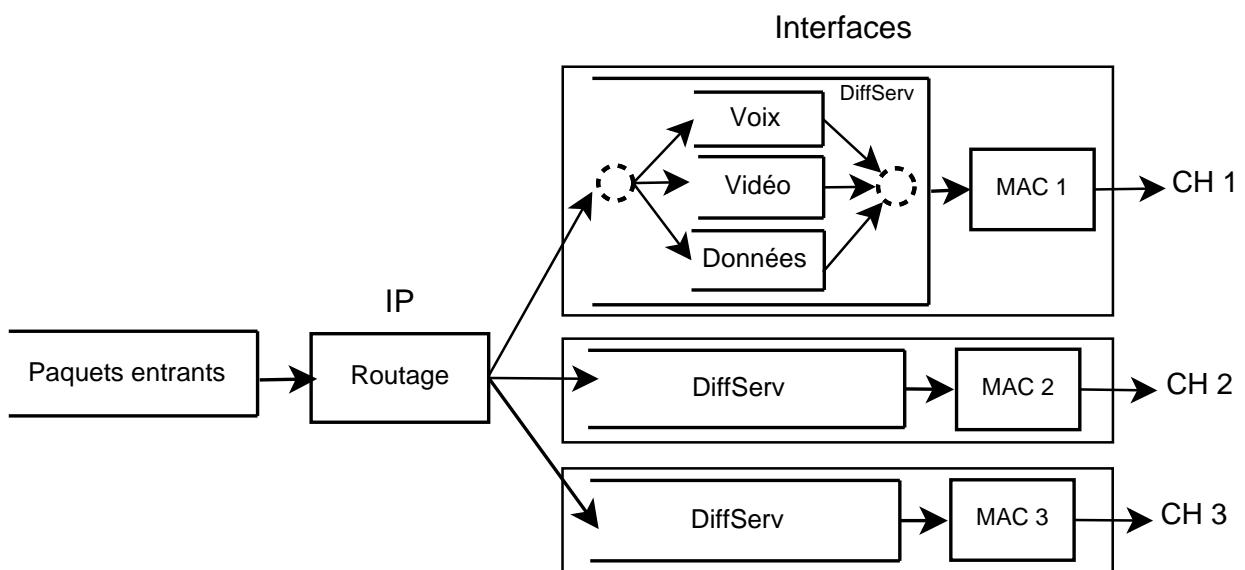


Figure 4.2 Protection DiffServ Standard dans un nœud mesh

vers les interfaces. Chaque interface a une chance sur trois (1/3) de recevoir un nouveau paquet. Dans cette configuration, le module DiffServ est implémenté sur chaque interface. Il y a alors autant d'ordonnanceurs DiffServ que d'interfaces. Il n'est donc pas nécessaire de faire une estimation du canal puisque tous les canaux seront utilisés avec la même probabilité. Cette configuration est celle du *DiffServ* classique ou DS. Chaque file virtuelle du module DiffServ compte au maximum 50 paquets de 1520 octets. Chaque interface peut donc recevoir au maximum 150 paquets de 1520 octets. Comme pour l'ordonnanceur *DiffServ**, la pondération utilisée par chaque ordonnanceur DiffServ est $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$.

La troisième configuration consiste à ne pas faire de différenciation de service. Il n'y a donc pas d'ordonnanceur DiffServ mais uniquement des files d'attente de type Premier Arrivé Premier Servi (FIFO). Les paquets sont servis par ordre d'arrivée à l'interface. Les interfaces sont choisies de manière équiprobable. De façon schématique, sa représentation est présentée sur la figure 4.3. Mesurée en octets, la capacité maximale d'une file FIFO est 150 paquets de 1520 octets. Ce cas est traité uniquement à titre comparatif.

L'objectif de cette section est de faire varier la charge de trafic envoyée dans le réseau et de comparer les performances offertes par ces trois configurations. Les caractéristiques des trafics (taille des paquets, taux d'arrivée, cycle d'activité) sont les mêmes que celles de la sous-section 3.1.

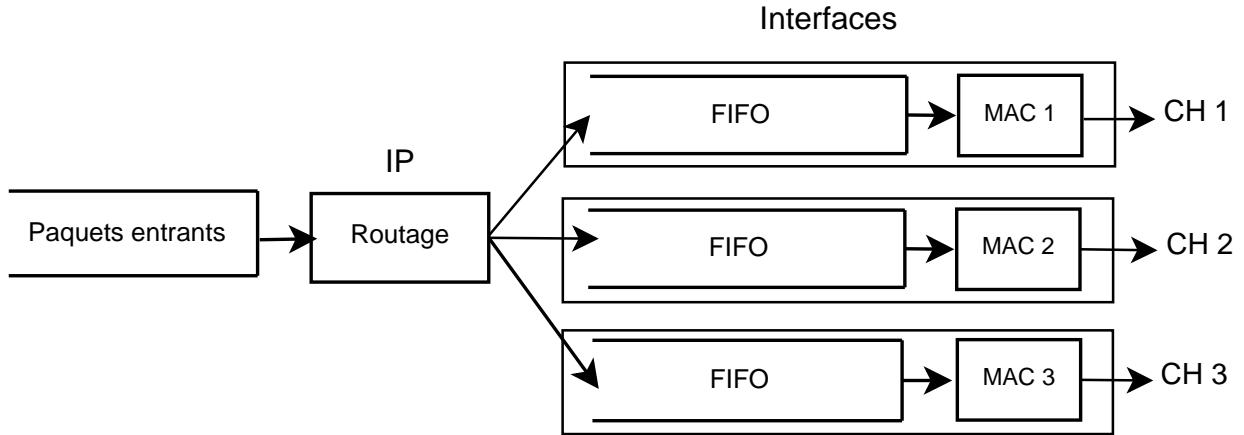


Figure 4.3 Files d'attente FIFO dans un noeud mesh

4.2.1 Classe EF – Trafic VoIP

Nous comparons, dans cette section, l'effet des trois politiques de service sur les performances du trafic VoIP.

4.2.1.1 Mesure des pertes

La figure 4.4 montre l'augmentation des pertes en fonction du volume de trafic envoyé et de la politique de service. Les différences entre le taux de perte des mécanismes DiffServ et DiffServ* ne sont pas très significatives étant donné que ces deux méthodes utilisent la même pondération pour faire la différenciation de service et que la capacité totale de tous les buffers est la même en passant d'une configuration à l'autre. D'où les pertes engendrées par ces deux méthodes sont représentées par une même courbe. Cette figure montre également que la politique FIFO semble offrir, dans ce cas, une meilleure qualité de service au trafic VoIP. En effet, jusqu'à environ 200 sources, il n'y a quasiment aucune perte dans le réseau. Cependant, au delà de 200 sources, les pertes DiffServ et DiffServ* augmentent beaucoup plus vite que les pertes FIFO. Et pour preuve, avec 212 sources de voix soit une charge approximative de 6.8 Mb/s, le trafic VoIP enregistre à peine 1% de perte avec FIFO contre 5% avec DiffServ ou DiffServ*. Toutefois, il n'est pas encore possible d'affirmer que FIFO garantit une meilleure qualité de service au trafic VoIP sans avoir observé les autres métriques de qualité de service.

4.2.1.2 Mesure du délai moyen

La figure 4.5 montre la variation du délai moyen du trafic VoIP en fonction du volume de trafic envoyé et de la politique de service. Cette figure montre un renversement de tendance par rapport aux observations de la figure 4.4. On constate d'abord que, quelle que soit la charge de trafic, DiffServ et DiffServ* engendrent moins de délai que FIFO. Par ailleurs, le gain en perte (4%) obtenu grâce à FIFO se traduit par une augmentation de délai d'environ 70 ms. Bien que cette augmentation du délai semble être acceptable, elle pourrait ne pas être tolérée par certaines applications de VoIP. Dans ce cas, les besoins spécifiques de chaque application doivent être pris en compte afin d'étudier la possibilité d'un éventuel compromis.

D'autre part, à faible charge, la diminution de délai avec DiffServ* s'explique par le fait que *Wireless-DiffServ** envoie préférentiellement les paquets vers l'interface qui a un canal de meilleure qualité c'est-à-dire le canal qui envoie les paquets à un taux plus élevé réduisant de ce fait le délai de transmission. Lorsque le réseau est saturé, les paquets arrivent très vite et il faut donc les servir le plus vite possible en utilisant toutes les ressources disponibles. Dans de telles circonstances, toutes les interfaces doivent être utilisées indépendamment de l'état de leur canal. C'est ce qui explique qu'à forte charge les performances de *DiffServ** tendent vers celles de DiffServ.

À ce stade de l'analyse, il est encore difficile de se prononcer par rapport à la meilleure politique de service, puisque nous n'avons pas encore observé l'impact de la politique de service sur la gigue, ni mesuré les performances des autres classes de trafic.

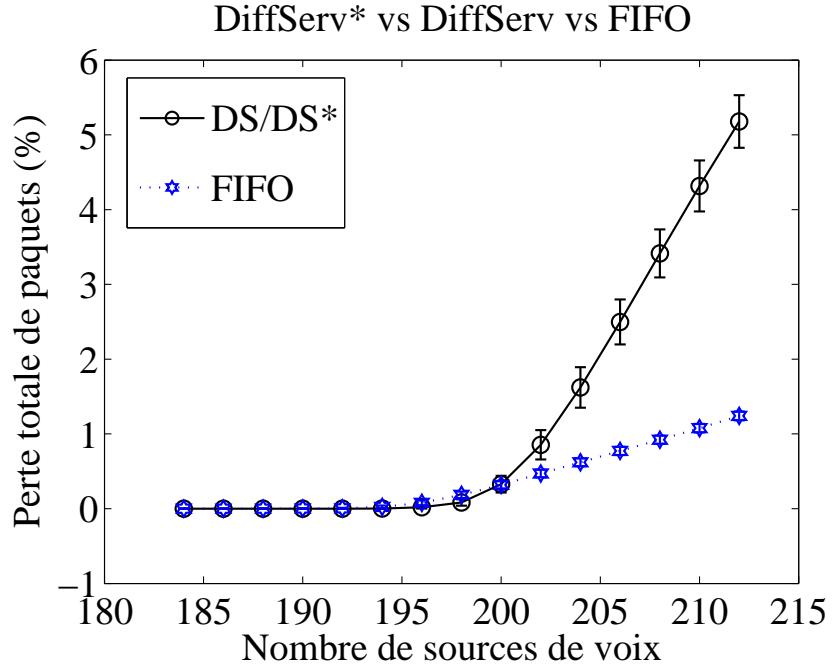


Figure 4.4 Réseau de 2 nœuds : Pertes du trafic VoIP en fonction de la politique de service.

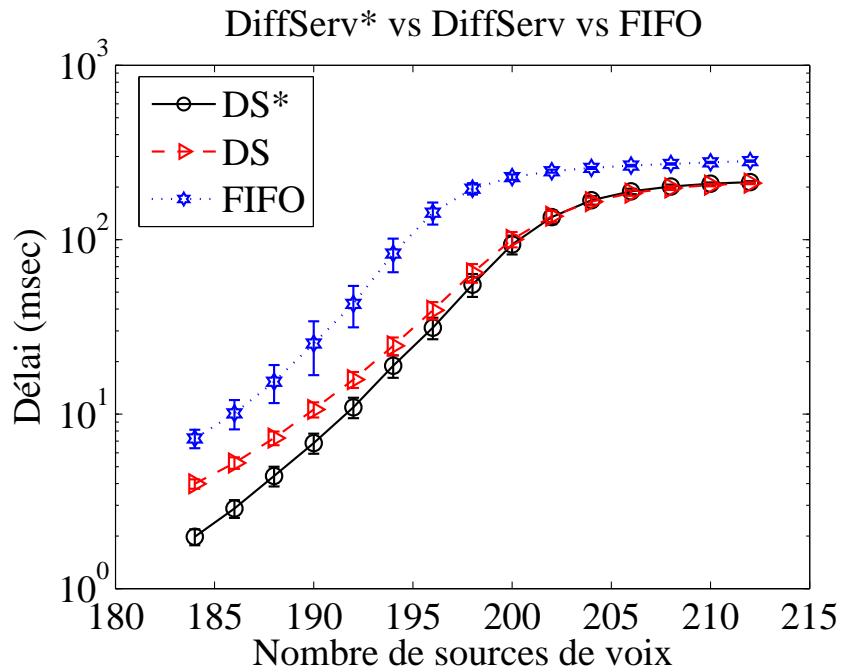


Figure 4.5 Réseau de 2 nœuds : Délai moyen du trafic VoIP en fonction de la politique de service.

4.2.1.3 Mesure de la gigue

La figure 4.6 montre la variation de la gigue du trafic VoIP en fonction du volume de trafic envoyé et de la politique de service. La politique de service des files d'attente affecte considérablement la gigue des paquets. Contrairement aux observations précédentes, on observe une forte différenciation de la gigue en fonction de la politique de service. Tandis que DiffServ et FIFO présente une gigue croissant en fonction du volume de trafic, la gigue mesurée avec DiffServ* ne fluctue presque pas.

La forte diminution de la gigue peut être due au fait que, lorsque les paquets d'une même rafale sont servis par l'ordonnanceur *DiffServ** situé dans la couche IP, ils sont presqu'aussitôt transmis dès qu'ils sont envoyés vers l'interface de sortie parce que *DiffServ** envoie les paquets vers une interface "disponible" c'est-à-dire prête à débuter une transmission. Le temps d'attente est donc le temps mis par l'ordonnanceur pour finir le cycle entamé avant l'arrivée de ces paquets plus le temps mis pour servir les paquets situés en tête de file. Lorsqu'il y a trois ordonnanceurs DiffServ, les paquets d'une même rafale sont distribués sur les trois interfaces. Le temps d'attente sur chaque interface est variable et dépend du nombre de paquets déjà en attente dans la file virtuelle à laquelle appartient le paquet et de la position des ordonnanceurs dans les cycles de WRR en cours. En conséquence, on voit une augmentation de la gigue avec la configuration DiffServ. De même, avec une configuration FIFO, lorsque les paquets d'une même rafale sont distribués sur les trois interfaces, le temps d'attente sur les interfaces est variable et dépend uniquement du nombre de paquets déjà en attente dans la file FIFO. C'est ce qui a pour conséquence d'augmenter la gigue moyenne lorsqu'on passe d'une configuration *DiffServ** à une configuration "non-*DiffServ**".

Eu égard à tout ce qui précède, la politique FIFO réduit les pertes du trafic VoIP tandis que DiffServ et DiffServ* en diminuent le délai moyen. De plus, seul DiffServ* en réduit la gigue moyenne.

4.2.2 Classe AF – Trafic Vidéo

Nous comparons, dans cette section, l'effet des trois politiques de service sur les performances du trafic Vidéo.

4.2.2.1 Mesure des pertes

La figure 4.7 montre l'augmentation des pertes du trafic vidéo en fonction du volume de trafic envoyé et de la politique de service. Les pertes des mécanismes DiffServ et DiffServ* sont combinées sur une même courbe étant donné que leurs différences sont presque négligeables. L'impact de la politique de service sur les pertes du trafic vidéo est contraire à ce qui a été

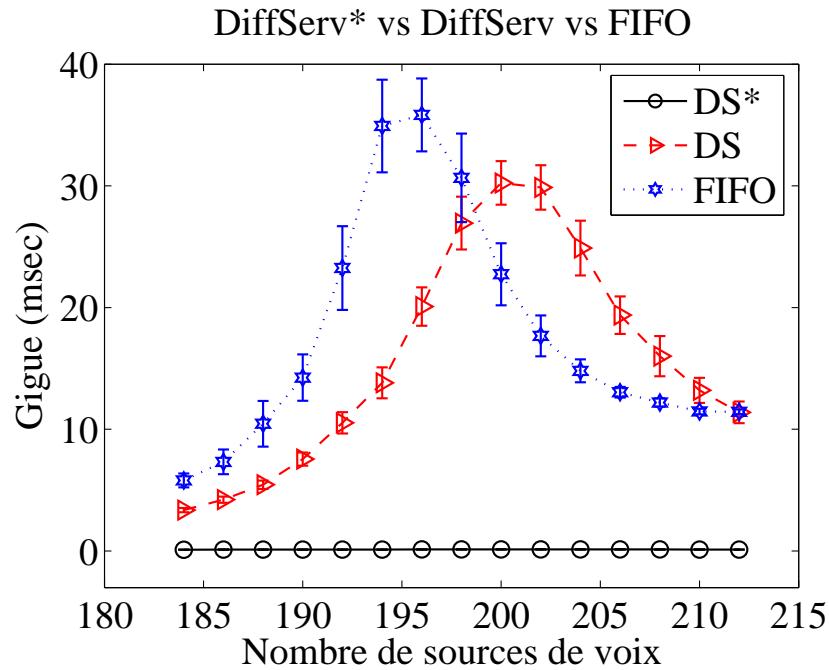


Figure 4.6 Réseau de 2 nœuds : Gigue du trafic VoIP en fonction de la politique de service.

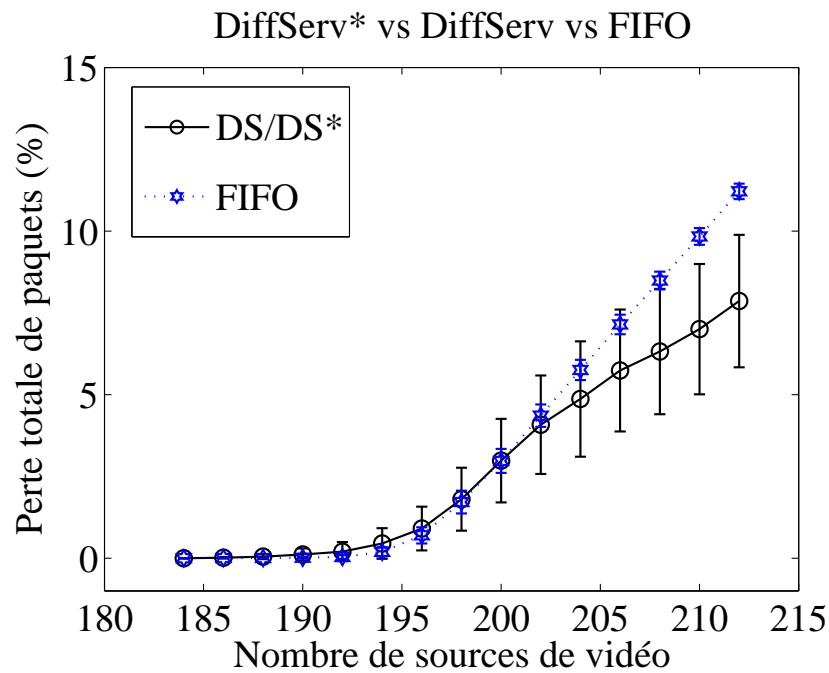


Figure 4.7 Réseau de 2 nœuds : Pertes du trafic vidéo en fonction de la politique de service.

observé sur la figure 4.4 montrant les pertes du trafic VoIP. En effet, avec moins de 200 sources actives, la politique FIFO engendre moins de perte que DiffServ ou DiffServ*. Toutefois cette différence n'est que peu significative. Au delà de 200 sources, les pertes DiffServ et DiffServ* sont nettement inférieures aux pertes avec FIFO. Avec 212 sources de vidéo, le trafic vidéo enregistre 8% avec DiffServ ou DiffServ* contre 11% de perte avec FIFO. Il en résulte que la différenciation de service (DiffServ ou DiffServ*) semble, a priori, offrir au trafic vidéo une meilleure qualité de service que la politique FIFO. Il convient alors d'observer les autres métriques afin de vérifier cette hypothèse.

4.2.2.2 Mesure du délai moyen

La figure 4.8 montre la variation du délai moyen du trafic vidéo en fonction du volume de trafic envoyé et de la politique de service. L'analyse de cette figure révèle un comportement identique à celui de la figure 4.7. En effet, quel que soit le volume de trafic, DiffServ* engendre moins de délai que FIFO. De plus, à faible charge, DiffServ* engendre également moins de délai que DiffServ compte tenu du fait que DiffServ* envoie les paquets vers l'interface qui est immédiatement prête à débuter la transmission et qui a le canal de meilleure qualité.

À ce niveau de l'analyse des performances, il est clair que la différenciation des services, DiffServ et DiffServ*, protège mieux le trafic vidéo que la politique de service FIFO. Observons enfin, l'effet de ces trois politiques sur la gigue du trafic vidéo.

4.2.2.3 Mesure de la gigue

La figure 4.9 montre la variation de la gigue du trafic vidéo en fonction du volume de trafic envoyé et de la politique de service. Comme sur la figure 4.6 qui présentait la gigue du trafic VoIP, on observe aussi sur la figure 4.9 que la gigue mesurée avec DiffServ* ne fluctue presque pas tandis que celle de DiffServ et FIFO varie considérablement en fonction du volume de trafic. La forte diminution de la gigue s'explique, ici encore, par les arguments avancés dans la sous-section 4.2.1.3. Avec DiffServ*, le temps d'attente des paquets dépend de la position de l'ordonnanceur dans le cycle de WRR en cours et du nombre de paquets déjà en attente avant l'arrivée de ces paquets. Avec DiffServ ou FIFO, le temps d'attente des paquets dépend de l'interface qui reçoit le paquet et ce temps peut varier d'une interface à l'autre. Sur chaque interface, le temps d'attente dépend du nombre de paquets déjà en attente et éventuellement de la position de l'ordonnanceur dans le cycle de WRR en cours (cas de DiffServ).

En conséquence, il ressort de tout ce qui précède que la politique *DiffServ** protège mieux le trafic vidéo du point de vue des trois métriques de qualité de service (perte, délai et gigue) que les politiques DiffServ et FIFO.

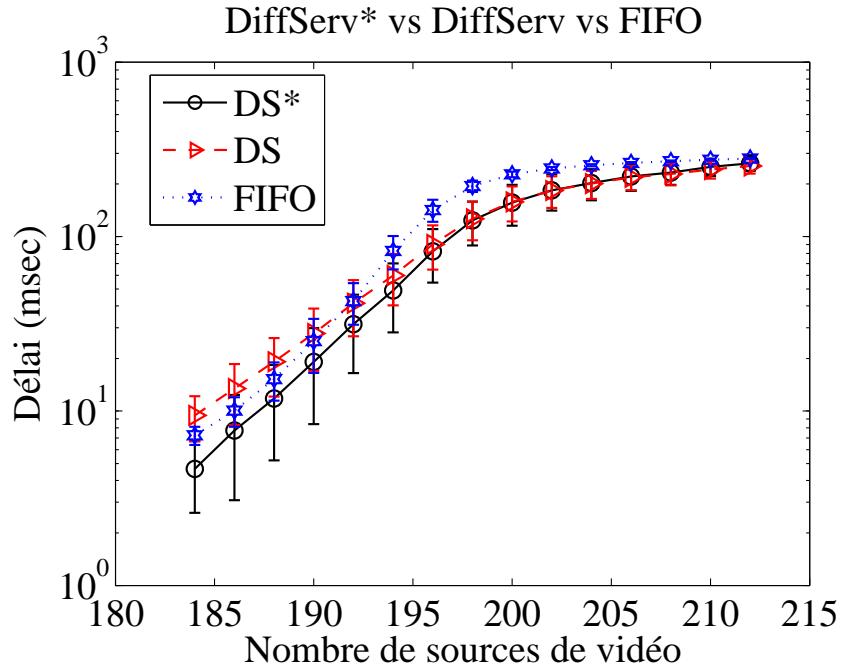


Figure 4.8 Réseau de 2 noeuds : Délai moyen du trafic vidéo en fonction de la politique de service.

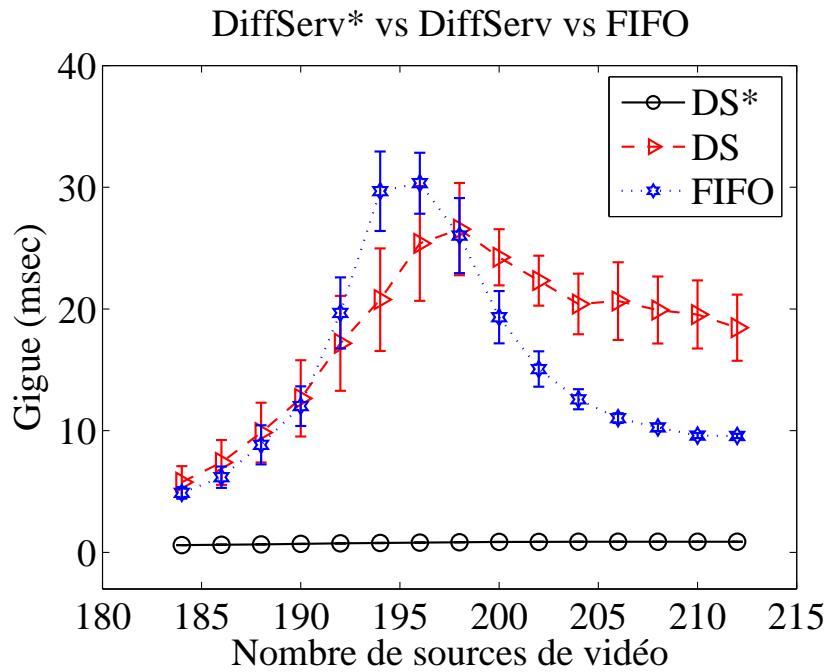


Figure 4.9 Réseau de 2 nœuds : Gigue du trafic vidéo en fonction de la politique de service.

4.2.3 Classe BE – Trafic Données

Nous comparons, dans cette section, l'effet des trois politiques de service sur les performances du trafic données.

4.2.3.1 Mesure des pertes

La figure 4.10 montre l'augmentation des pertes du trafic données en fonction du volume de trafic envoyé et de la politique de service. Encore une fois, les pertes des mécanismes DiffServ et DiffServ* sont combinées sur une même courbe étant donné que leurs différences sont presque négligeables. L'impact de la politique de service sur les pertes du trafic données est identique à ce qui a été observé sur la figure 4.7 qui révélait les pertes du trafic vidéo. En effet, avec moins de 200 sources actives, la politique FIFO engendre moins de perte que DiffServ ou DiffServ*. Toutefois, cette différence est très peu significative. Au delà de 200 sources, les pertes DiffServ et DiffServ* sont nettement inférieures aux pertes avec FIFO. Avec 212 sources de données, on enregistre 17% avec DiffServ ou DiffServ* contre 33% de perte avec FIFO, soit environ le double de pertes DiffServ*. La différenciation de service semble donc, a priori, offrir au trafic données une meilleure qualité de service que la politique FIFO. Observons alors l'effet de la politique de service sur le délai moyen et la gigue afin de vérifier cette hypothèse.

4.2.3.2 Mesure du délai moyen

La figure 4.11 montre la variation du délai moyen du trafic données en fonction du volume de trafic et de la politique de service. Contrairement à la figure 4.10, la figure 4.11 montre que c'est avec la politique FIFO que l'on note le moins de délai. Ce comportement est tout à fait normal étant donné qu'en tolérant plus de perte dans un réseau, on s'attend à obtenir moins de délai. C'est un compromis couramment observé entre le taux de perte et le délai moyen d'un trafic. Cependant, dans le cas présent, ce compromis ne semble pas très avantageux pour le trafic données.

En effet, si avec moins de 200 sources actives, FIFO procure moins de perte et moins de délai au trafic données, au delà de 200 sources, les pertes FIFO augmentent beaucoup plus vite que celles des deux autres politiques tandis que le délai moyen sature approximativement à 276 ms pour FIFO contre 304 ms pour DiffServ et DiffServ* soit une différence d'environ 30 ms. En conséquence, en tolérant le double des pertes mesurées avec DS/DS*, on gagnerait seulement 30 ms de délai. Sachant que la classe BE n'a aucune exigence de qualité de service, surtout pas en ce qui concerne le délai de ses paquets, il est préférable, dans la situation

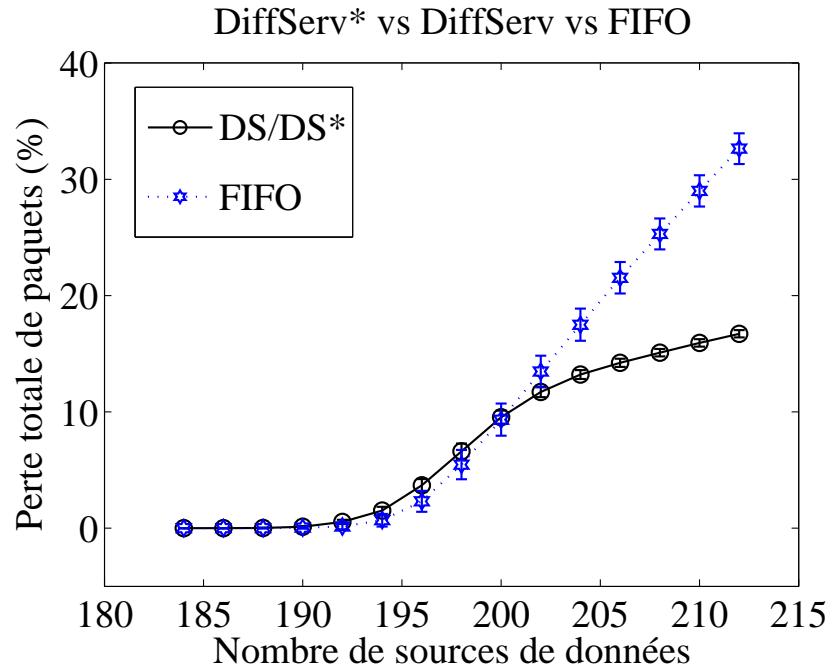


Figure 4.10 Réseau de 2 noeuds : Pertes du trafic données en fonction de la politique de service.

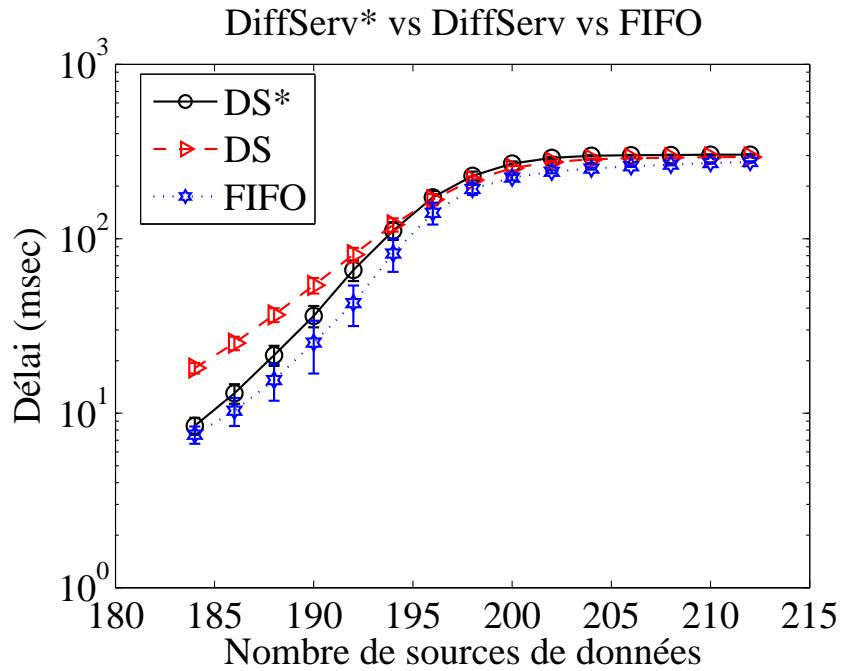


Figure 4.11 Réseau de 2 noeuds : Délai moyen du trafic données en fonction de la politique de service.

actuelle, d'avoir un délai supplémentaire de 30 ms par paquet que de perdre 16% des paquets juste pour gagner ces 30 ms.

L'analyse des figures 4.10 et 4.11 montre que la différenciation des services, DiffServ et DiffServ*, protège mieux le trafic données que la politique de service FIFO. L'effet de ces trois politiques sur la gigue du trafic données nous permettra de départager DiffServ et DiffServ*.

4.2.3.3 Mesure de la gigue

La figure 4.12 montre la variation de la gigue du trafic données en fonction du volume de trafic envoyé et de la politique de service. Comme sur les figures 4.6 et 4.9, la figure 4.12 montre que la gigue mesurée avec DiffServ* ne varie presque pas tandis que celle de DiffServ et FIFO évolue considérablement en fonction du volume de trafic. En effet, les courbes de DiffServ et FIFO sont en forme de cloche tandis que celle de DiffServ* est presque constante. La forte diminution de la gigue s'explique, ici encore, par les arguments avancés dans les sous-sections 4.2.1.3 et 4.2.2.3.

En conséquence, il ressort de tout ce qui précède que la politique *DiffServ** protège mieux le trafic données que les politiques DiffServ et FIFO.

Nous venons d'étudier, séparément, l'impact de la politique de service sur la qualité de service de chaque classe de trafic. Les résultats montrent clairement que *DiffServ** diminue la gigue pour toutes les classes, mais protège également les trafics vidéo et données des pertes importantes engendrées par FIFO. En ce qui concerne le trafic VoIP, la conclusion n'est pas aussi évidente puisque nous avons montré que FIFO réduit de 4% les pertes de trafic VoIP mais augmente son délai d'environ 70 ms. L'analyse des pertes de paquet par classe nous amène à penser que la politique FIFO fait apparaître une certaine différenciation de service en faveur du trafic VoIP. Cette hypothèse mérite d'être vérifiée par une étude synchrone des performances de toutes les classes. Cette étude permettra également de vérifier ou non l'efficacité de DiffServ* dans un réseau de deux noeuds.

4.2.4 Étude du compromis entre DiffServ* et FIFO

Dans cette section, nous comparons les performances du module DiffServ* avec celles des modules FIFO par interface.

La figure 4.13 montre l'évolution du taux de perte des deux mécanismes lorsque la charge de trafic augmente dans le réseau. On constate, bel et bien, sur cette figure que les courbes illustrant le fonctionnement des files FIFO font apparaître une certaine différenciation entre les trois classes de trafic. En effet, à partir de 192 sources par classe, les pertes des files FIFO se démarquent des pertes *DiffServ**. Comparées aux pertes *DiffServ**, les pertes des trafics

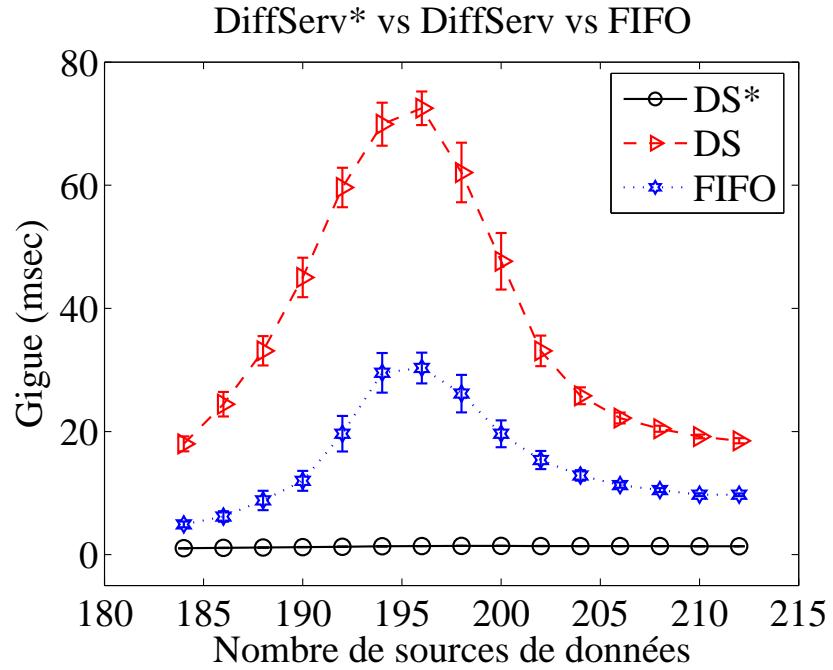


Figure 4.12 Réseau de 2 noeuds : Gigue du trafic données en fonction de la politique de service.

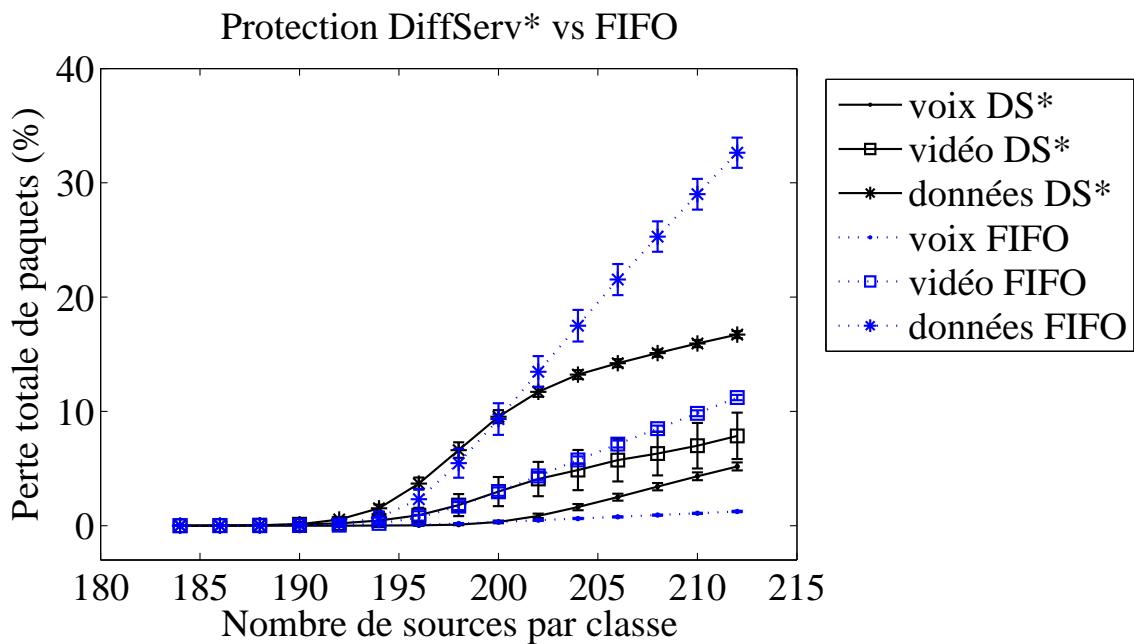


Figure 4.13 DiffServ* vs FIFO : Perte moyenne par classe dans un réseau de 2 noeuds

vidéo et données augmentent rapidement tandis que les pertes du trafic de voix diminuent par rapport aux pertes *DiffServ**. Pour un volume de trafic équivalent à 212 sources/classe soit environ 6.8 Mb/s par classe, le trafic de voix n'enregistre presque aucune perte alors que les pertes des trafics de vidéo et de données atteignent respectivement 17% et 33%. Le fort taux de perte observé pour ces deux trafics (vidéo et données) est lié à la capacité des files d'attente. Les buffers étant mesurés en octets, lorsque ceux-ci commencent à se remplir, ils n'acceptent de gros paquets que lorsque l'espace disponible le permet. Expliquons nos propos par un exemple.

En incluant l'en-tête IP, la taille moyenne des paquets de voix est 100 octets, celle des paquets vidéo est 820 octets et celle des données 1460 octets. Supposons que le buffer peut recevoir un dernier paquet de 1520 octets pour se remplir. Cela équivaut à :

- $1520/100 = 15$ paquets de voix ou,
- $1520/820 = 1$ paquet vidéo et 7 paquets de voix ou,
- $1520/1460 = 1$ paquet de données.

Dans ces conditions, si le premier paquet qui arrive est un paquet de voix, il est évident que tous les paquets de données qui viendront après seront rejetés. De même si les 8 premiers paquets arrivés sont des paquets de voix, les paquets vidéo qui suivront seront aussi rejetés. Ainsi la mesure des files d'attente en octets est davantage favorable aux paquets de petite taille qu'aux paquets plus gros. Dans ce projet, les paquets de petite taille appartiennent à la classe EF et les gros paquets appartiennent aux classes AF et BE. C'est ce qui justifie que la différenciation de service apparue sur la figure 4.13 soit favorable aux paquets de voix. Cependant, la situation serait certainement très différente si les gros paquets appartenaient à la classe EF.

L'analyse de la figure 4.13 à elle seule, montre que l'utilisation des files FIFO mesurées en octets donne davantage de priorité à la VoIP, réduisant ainsi de 4% le taux de perte de ce trafic par rapport aux pertes mesurées pour ce même trafic en utilisant *DiffServ**. Ce gain engendre, cependant, une augmentation des pertes du trafic vidéo d'à peu près 4% et une augmentation des pertes du trafic données d'à peu près 17%. Il s'agit alors d'un compromis entre le taux de perte toléré pour le trafic voix et les pertes des deux autres classes de trafic. Dans le cas actuel, c'est le trafic données qui subit presque tout l'impact d'un tel compromis étant donné qu'il perd deux fois plus de paquets que *DiffServ**, soit environ le quadruple du gain réalisé pour le trafic VoIP. Vu comme cela, il est plus intéressant d'admettre une perte supplémentaire de 4% du trafic VoIP en utilisant *DiffServ** afin de récupérer en contrepartie environ 4% du trafic vidéo mais surtout près de 17% du trafic données.

Sur les figures 4.14 et 4.15, les courbes illustrant les performances des files FIFO montrent que le délai moyen et la gigue par classe ne sont pas différenciés. C'est ce qui caractérise

normalement l'utilisation de files FIFO dans un réseau. Ainsi, la figure 4.14 montre que le

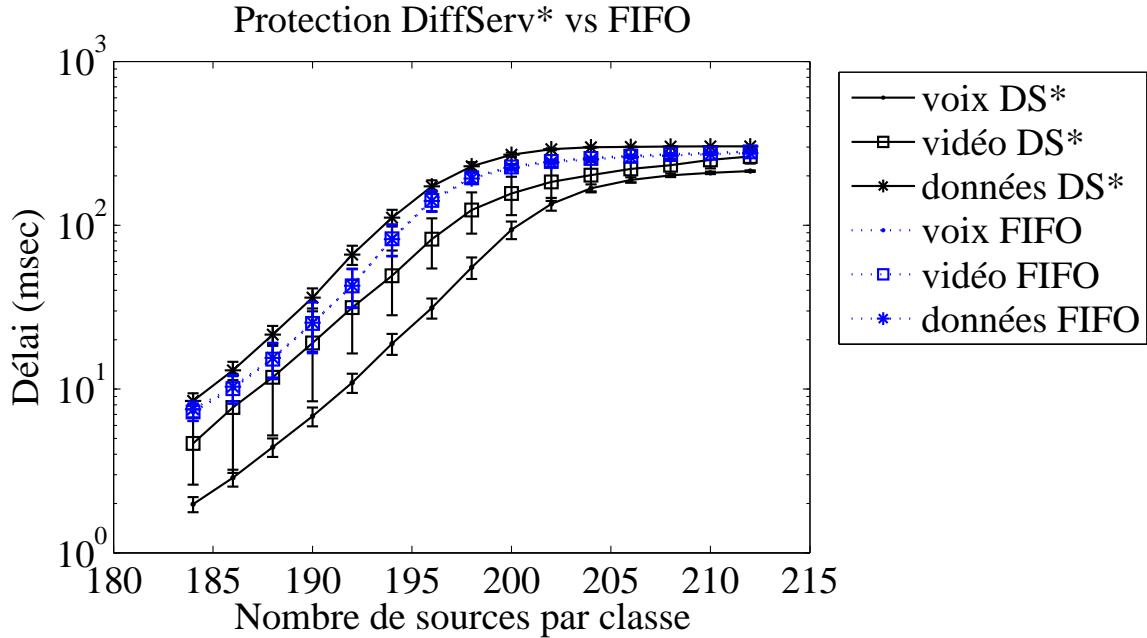


Figure 4.14 DiffServ* vs FIFO : Délai moyen par classe dans un réseau de 2 nœuds

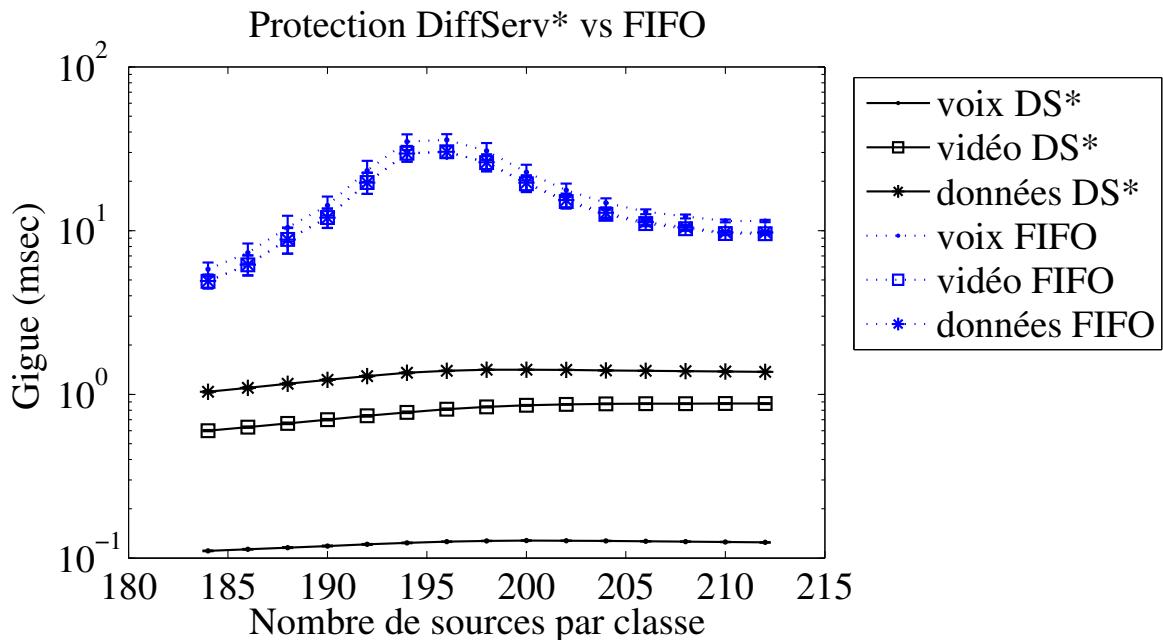


Figure 4.15 DiffServ* vs FIFO : Gigue moyenne par classe dans un réseau de 2 nœuds

compromis autorisé pour les pertes n'a aucune incidence sur le délai mesuré avec la politique

FIFO tandis que *DiffServ** permet de réduire, notamment à faible charge, le délai moyen des classes EF et AF par rapport aux valeurs mesurées lorsque nous utilisons des files FIFO.

La figure 4.15, quant à elle, indique encore une fois que, comparé au mécanisme FIFO, *DiffServ** réduit la gigue moyenne par paquet. La justification est la même que celle des sous-sections précédentes. Lorsque les paquets d'une même rafale sont distribués sur les trois interfaces, le temps d'attente sur chaque interface est variable et dépend uniquement du nombre de paquets déjà en attente dans la file FIFO. C'est ce qui a pour conséquence d'augmenter la gigue moyenne lorsqu'on passe d'une configuration *DiffServ** à une configuration "non-*DiffServ**".

Au regard de tout ce qui précède, l'utilisation de *DiffServ** s'avère être l'option la plus équitable dans ce réseau d'une part parce qu'elle garantit une différenciation de service au niveau des trois métriques de performance (perte, délai, gigue) et d'autre part, parce qu'elle régule également les pertes des trafics de vidéo et de données comparativement aux pertes observées lors de l'utilisation des files FIFO.

Les observations des sous-sections précédentes, 4.2.1, 4.2.2, 4.2.3 et 4.2.4, montrent l'efficacité du mécanisme *DiffServ** dans un réseau sans fil multi-radios composé de 2 nœuds. Il en ressort essentiellement que *DiffServ** permet de réduire la gigue des paquets et qu'à faible charge, il réduit également le délai moyen par paquet.

Dans la section suivante, nous comparerons à nouveau les trois (3) configurations étudiées dans cette section (*DiffServ**, DiffServ et FIFO) dans un réseau de cinq (5) nœuds. Le but de cette étude étant de valider ou d'infirmer la conclusion selon laquelle *DiffServ** offre une meilleure qualité de service que les deux autres politiques étudiées (DiffServ et FIFO).

4.3 Performance dans un réseau

Nous reprenons l'étude réalisée à la section 4.2 en considérant cette fois-ci un réseau maillé de 5 nœuds dont la topologie est présentée sur la figure 4.16.

Il existe trois (3) routes logiques entre la source *S* et la destination *D* (*S* – *A* – *D*, *S* – *B* – *D* et *S* – *C* – *D*) et un lien logique entre chaque paire de nœuds appartenant à une route logique.

Lorsque la source *S* génère un paquet, le protocole de routage MPOLSR lui assigne la route à suivre en utilisant l'algorithme de "round robin". Ainsi, si le paquet 1 est envoyé par la route *S* – *A* – *D*, le paquet 2 sera envoyé par la route *S* – *B* – *D*, le paquet 3 par la route *S* – *C* – *D*, le paquet 4 par la route *S* – *A* – *D* et ainsi de suite. Après avoir choisi la route logique par laquelle le paquet sera acheminé, il faut choisir l'interface par laquelle il doit être envoyé. L'assignation de l'interface de sortie dépend de la configuration interne étudiée. Lorsqu'un paquet arrive à un nœud intermédiaire, il n'y a plus de route à sélectionner. Le

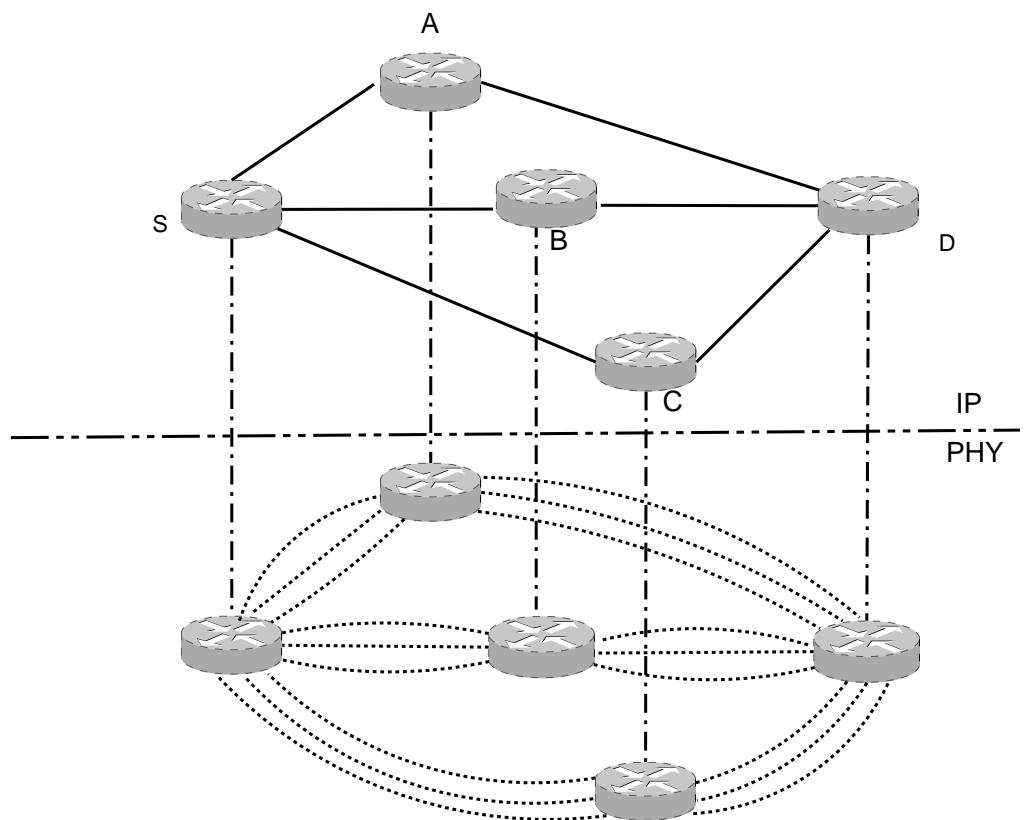


Figure 4.16 Réseau sans fil à 5 nœuds

protocole MPOLSR qui s'exécute aussi au nœud intermédiaire vérifie que le prochain nœud auquel le paquet sera envoyé est bien un voisin du nœud actuel. À partir de ce point, chaque paire de nœuds successifs est comparable à un réseau de 2 nœuds reliés par un lien logique. La description de cette architecture a été faite dans la section 4.2. Si le modèle *DiffServ** est utilisé, soit, le paquet est directement envoyé vers l'interface disponible qui a le meilleur canal, soit, il est placé dans un buffer en attendant qu'une interface se libère. Si les modèles DiffServ ou FIFO sont utilisés, MPOLSR choisit aléatoirement l'interface par laquelle le paquet doit être envoyé au prochain nœud. Ensuite le paquet est soit mis en attente dans une file virtuelle de l'interface, soit envoyé directement.

Les autres paramètres de simulation restent inchangés c'est-à-dire les poids du WRR, la capacité maximale des files d'attente, les propriétés statistiques du canal et les caractéristiques du trafic.

4.3.1 Classe EF – Trafic VoIP

Nous comparons, dans cette section, l'effet des trois politiques de service sur les performances du trafic VoIP.

4.3.1.1 Mesure des pertes

La figure 4.17 montre l'augmentation des pertes en fonction du volume de trafic envoyé et de la politique de service. Les pertes occasionnées par les deux méthodes de différenciation sont du même ordre de grandeur mais cette fois-ci, les différences sont plus significatives et on constate que par rapport à DiffServ, *DiffServ** réduit de 1% les pertes du trafic VoIP. C'est l'un des avantages de l'utilisation de plusieurs routes disjointes entre deux nœuds. En effet, la trifurcation du trafic permet de répartir la charge totale du trafic sur plusieurs routes et de réduire les pertes dues aux évanouissements du canal sans fil ou à la congestion des liens.

En comparant les politiques DiffServ et *DiffServ** à la politique FIFO, les résultats sont semblables à ceux d'un réseau de 2 nœuds. En effet, à faible charge c'est-à-dire jusqu'à 96 sources, les trois politiques produisent presqu'un même taux de perte avec un léger avantage pour *DiffServ**. Cependant, au delà de 96 sources ou à forte charge, la politique FIFO génère moins de perte que les mécanismes DiffServ et *DiffServ**. Avec 106 sources actives, les files FIFO réduisent approximativement de 5.5% les pertes mesurées avec DS*. Comme nous l'avons montré dans la section 4.2, ce gain en perte est lié au fait que la capacité des files d'attente FIFO est mesurée en octets et que les paquets les plus petits appartiennent à la classe EF.

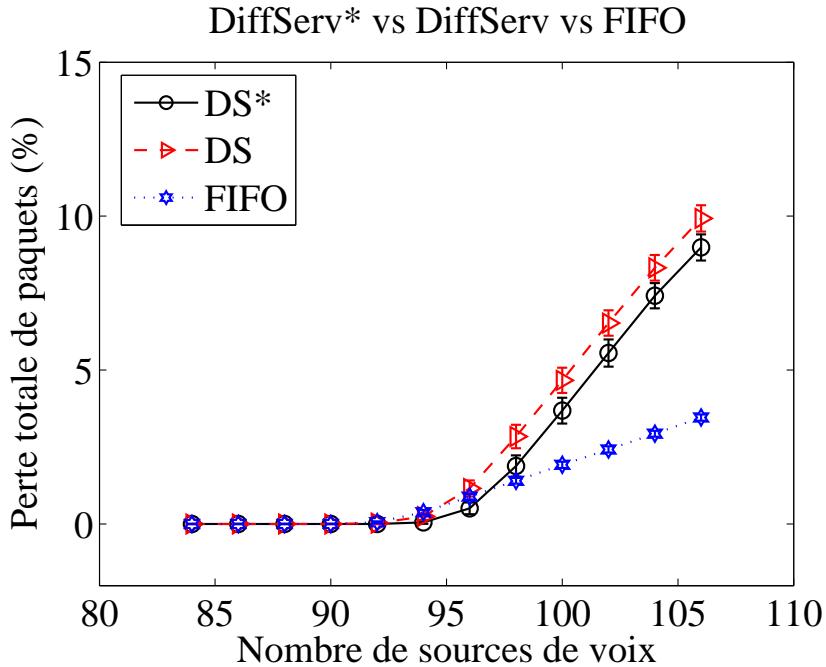


Figure 4.17 Réseau de 5 nœuds : Pertes du trafic VoIP en fonction de la politique de service.

4.3.1.2 Mesure du délai moyen

La figure 4.18 montre la variation du délai moyen du trafic VoIP en fonction du volume de trafic envoyé et de la politique de service. Les conclusions sont les mêmes que celles de la sous-section 4.2.1.2. Quel que soit le volume de trafic, et en particulier à faible charge, DiffServ et DiffServ* engendrent nettement moins de délai que FIFO. Par ailleurs, le gain en perte (5.5%) obtenu grâce à FIFO se traduit par une augmentation de délai d'environ 160 ms. Ainsi, en augmentant la complexité du réseau, le gain en perte issu de l'utilisation des files FIFO passe de 4% à 5.5% tandis que l'augmentation de délai passe de 70 ms à 160 ms. Un délai supplémentaire de 160 ms pourrait s'avérer intolérable pour certaines applications. Dans une telle situation, les besoins spécifiques de chaque application doivent être pris en compte afin d'étudier la possibilité d'un éventuel compromis entre le taux de perte et le délai moyen. Dans le cas où un tel compromis ne serait pas envisageable, il conviendra d'utiliser DiffServ ou DiffServ* pour que le réseau fonctionne de façon optimale. Rappelons, néanmoins, que si les gros paquets appartenaient à la classe EF, les files FIFO ne feraient pas apparaître cette différenciation en faveur du trafic VoIP, et aucun compromis ne serait nécessaire.

Par ailleurs, *DiffServ** améliore considérablement le délai moyen lorsque le réseau fonctionne en dessous de la zone de saturation étant donné que *DiffServ** n'envoie pas les paquets au hasard mais les envoie préférentiellement vers l'interface de sortie qui a le canal de

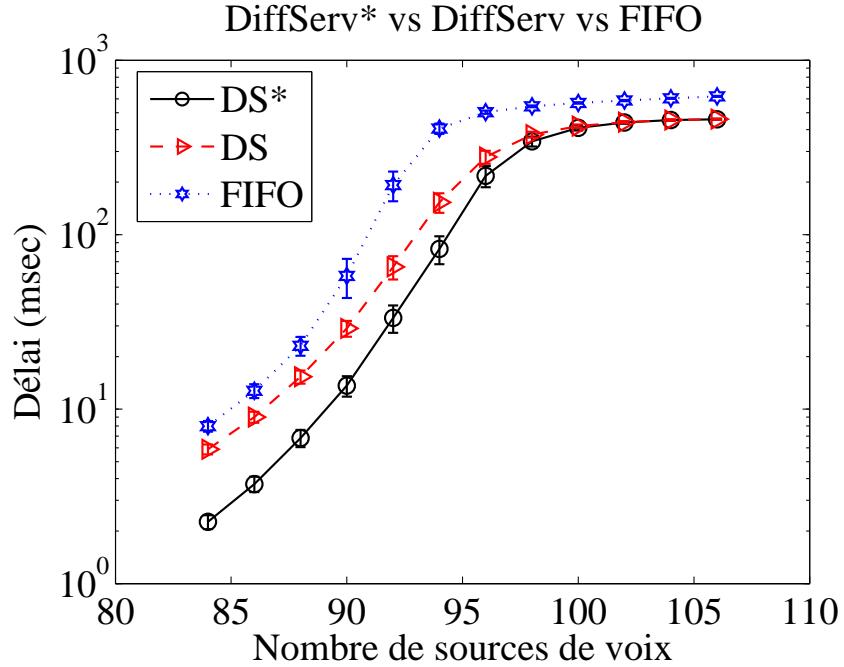


Figure 4.18 Réseau de 5 noeuds : Délai moyen du trafic VoIP en fonction de la politique de service.

meilleure qualité c'est-à-dire que ce canal envoie les paquets au taux de transmission le plus élevé réduisant ainsi le délai de transmission. À forte charge, il est nécessaire d'utiliser toutes les interfaces afin de ne pas augmenter le temps de séjour des paquets dans les buffers. En régime saturé, on mesure une augmentation d'environ 10 ms par rapport à DiffServ. Cette légère augmentation du délai est due à la diminution du taux de perte, gain de 1% par rapport à DiffServ.

4.3.1.3 Mesure de la gigue

La figure 4.19 montre la variation de la gigue du trafic VoIP en fonction du volume de trafic envoyé et de la politique de service. Cette figure confirme que *DiffServ** réduit effectivement la gigue moyenne des paquets pour les mêmes raisons que celles évoquées dans la section 4.2. La gigue est la variation du délai entre deux paquets consécutifs. Dans l'analyse qui suit, nous faisons l'hypothèse que deux paquets consécutifs appartiennent à une même rafale. Avec le mécanisme *DiffServ**, le temps de séjour, dans les buffers, des paquets d'une même rafale dépend uniquement du nombre de paquets déjà en attente et de la position de l'ordonnanceur dans le cycle de WRR en cours. En utilisant des files d'attente par interface (DiffServ ou FIFO), les paquets de la même rafale sont distribués aléatoirement sur toutes les interfaces. Le temps d'attente sur chaque interface dépend des deux faits sus-cités dans

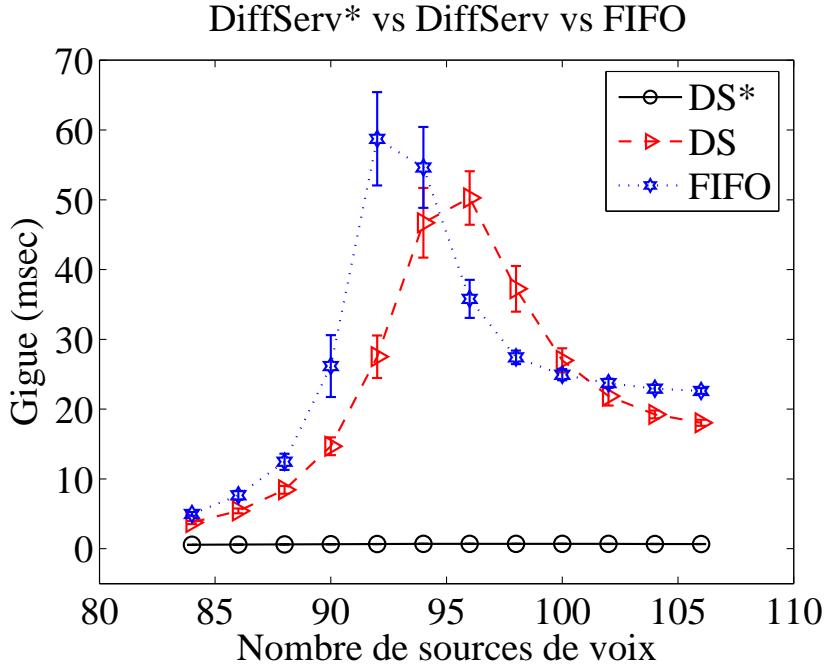


Figure 4.19 Réseau de 5 nœuds : Gigue du trafic VoIP en fonction de la politique de service.

le cas de DiffServ, et uniquement du nombre de paquets déjà en attente dans le cas FIFO. Cependant, ce temps varie d'une interface à l'autre et lorsque deux paquets consécutifs d'une même rafale se retrouvent sur des interfaces différentes, ils subissent des délais d'attente différents et cela se traduit par une augmentation de la gigue.

Eu égard à tout ce qui précède, la politique FIFO réduit les pertes du trafic VoIP tandis que DiffServ et DiffServ* en diminuent le délai moyen. De plus, seul DiffServ* en réduit la gigue moyenne.

4.3.2 Classe AF – Trafic Vidéo

Nous comparons, dans cette section, l'effet des trois politiques de service sur les performances du trafic Vidéo.

4.3.2.1 Mesure des pertes

La figure 4.20 montre l'augmentation des pertes du trafic vidéo en fonction du volume de trafic envoyé et de la politique de service. DiffServ* occasionne moins de perte que les deux autres politiques. En effet, on note une diminution de 1% en faveur de DiffServ* entre DS et DS*, et avec 106 sources actives, une diminution d'environ 6% entre DS* et FIFO. L'augmentation de 6% est à peu près égale à la diminution des pertes du trafic VoIP de 5.5%.

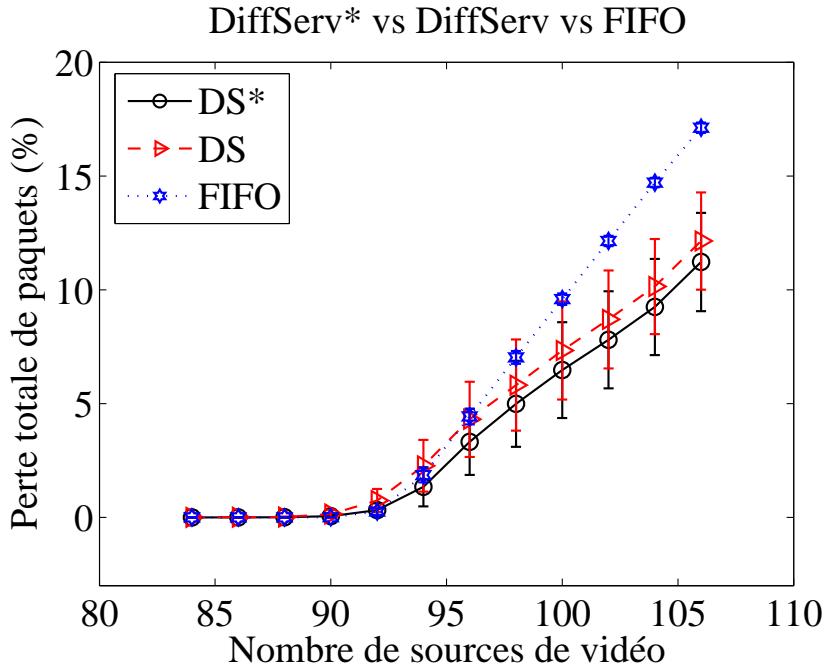


Figure 4.20 Réseau de 5 nœuds : Pertes du trafic vidéo en fonction de la politique de service.

Il en ressort que c'est donc le prix à payer pour obtenir de la différenciation avec l'usage des files FIFO, bien qu'elles n'aient pas été conçues à cet effet. Ainsi, du point de vue des pertes, c'est la différenciation de service (DiffServ ou DiffServ*) qui offre au trafic vidéo une meilleure qualité de service que la politique FIFO.

4.3.2.2 Mesure du délai moyen

La figure 4.21 montre la variation du délai moyen du trafic vidéo en fonction du volume de trafic envoyé et de la politique de service. L'analyse de cette figure révèle un comportement identique à celui de la figure 4.20. En effet, quelle que soit la charge de trafic, DiffServ* engendre moins de délai que FIFO. De plus, à faible charge, DiffServ* engendre également moins de délai que DiffServ compte tenu du fait que DiffServ* envoie les paquets vers l'interface qui est immédiatement prête à débuter la transmission et qui a le canal de meilleure qualité. Cependant, en régime saturé, on mesure une augmentation d'environ 10 ms par rapport à DiffServ. Cette légère augmentation du délai est due à la réduction de 1% des pertes causées par DS*.

En conséquence, du point de vue des pertes et du délai, DiffServ* offre au trafic vidéo une meilleure qualité de service que les politiques DiffServ et FIFO.

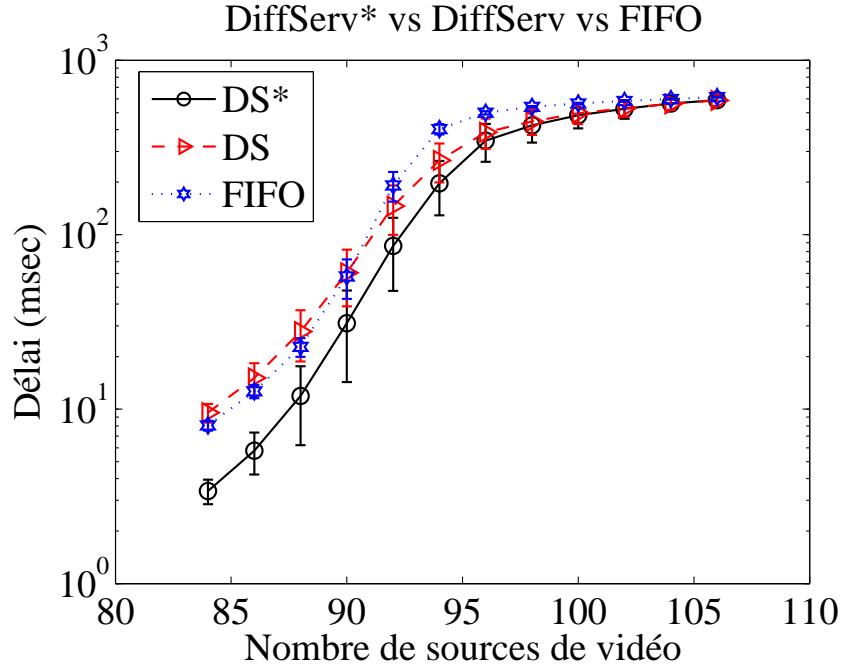


Figure 4.21 Réseau de 5 noeuds : Délai moyen du trafic vidéo en fonction de la politique de service.

4.3.2.3 Mesure de la gigue

La figure 4.22 montre la variation de la gigue du trafic vidéo en fonction du volume de trafic envoyé et de la politique de service. Comme attendu, DiffServ* maintient la gigue constante tandis que les gigues mesurées avec DiffServ et FIFO sont variables en fonction du volume de trafic. DiffServ* a donc un effet régulateur sur le trafic du réseau. La diminution de la gigue s'explique par les mêmes raisons que celles avancées dans la section 4.2.

En conséquence, il ressort de tout ce qui précède que la politique *DiffServ** protège mieux le trafic vidéo que les politiques DiffServ et FIFO puisqu'elle produit notamment moins de perte, moins de délai et moins de gigue.

4.3.3 Classe BE – Trafic Données

Nous comparons, dans cette section, l'effet des trois politiques de service sur les performances du trafic données.

4.3.3.1 Mesure des pertes

La figure 4.23 montre l'augmentation des pertes du trafic données en fonction du volume de trafic envoyé et de la politique de service. Comme pour le trafic vidéo, DiffServ* occasionne

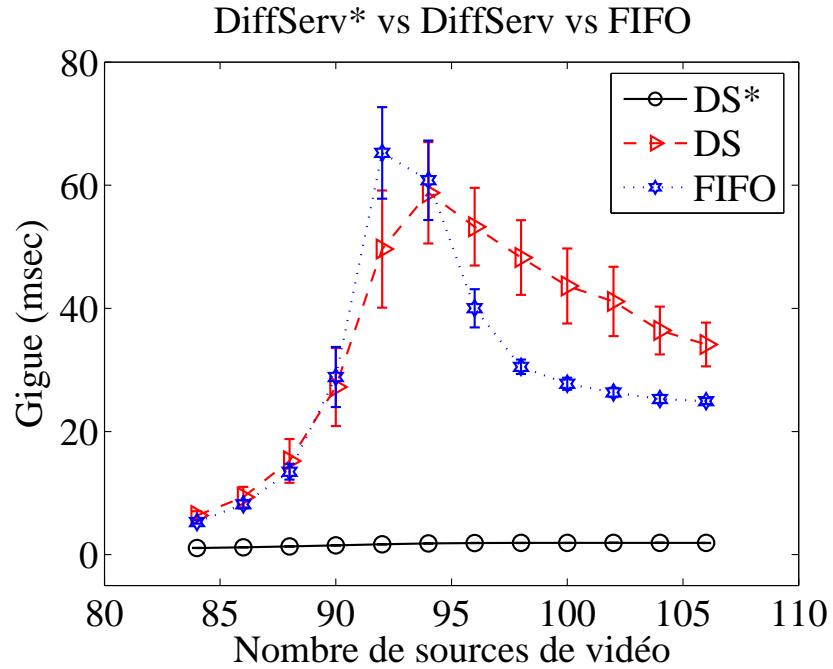


Figure 4.22 Réseau de 5 nœuds : Gigue du trafic vidéo en fonction de la politique de service.

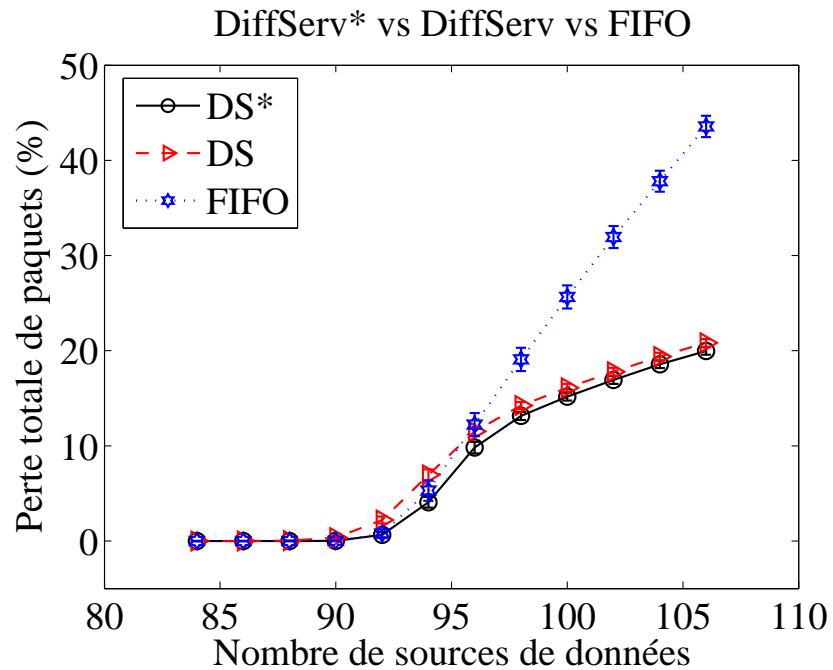


Figure 4.23 Réseau de 5 nœuds : Pertes du trafic données en fonction de la politique de service.

moins de perte que les deux autres politiques. En effet, on note, ici aussi, une diminution des pertes de 1% en faveur de DiffServ*, entre DiffServ et DiffServ*. Jusqu'à 96 sources actives, la politique FIFO engendre moins de perte que DiffServ mais plus de perte DiffServ*. Toutefois ces différences sont très peu significatives. Au delà de 96 sources, les pertes DiffServ et DiffServ* sont nettement inférieures aux pertes avec FIFO. Avec 106 sources de données, on enregistre approximativement 20% de perte avec DiffServ*, avec 21% DiffServ et 44% de perte avec FIFO, soit environ le double de pertes DiffServ/DiffServ*.

4.3.3.2 Mesure du délai moyen

La figure 4.24 montre la variation du délai moyen du trafic données en fonction du volume de trafic et de la politique de service. Les avantages de la trifurcation du trafic se font ressentir également sur cette figure. En effet, contrairement à la figure 4.2.3.2 les politiques DiffServ* et FIFO produisent pratiquement le même délai moyen tant que le réseau n'est pas saturé. La politique DiffServ, quant à elle, produit légèrement plus de délai que DiffServ* et FIFO lorsque le réseau est exploité en dessous de la saturation. Lorsque le réseau est saturé, nous mesurons approximativement 637 ms avec DiffServ*, 628 ms avec DiffServ et 602 ms avec FIFO. Cela équivaut à une augmentation d'environ 10 ms de DiffServ à DiffServ* due à la diminution des pertes de 1% et une augmentation d'environ 35 ms de FIFO à DiffServ* due à la diminution des pertes de 24%. Le compromis entre DiffServ et DiffServ* est tout à fait concevable dans la mesure où on gagne 1% de perte tandis que le délai moyen de tous les paquets augmente de 10 ms avec DiffServ*. Cependant le compromis entre DiffServ/DiffServ* et FIFO semble moins payant car on gagne seulement 30 ms ou 35 ms de délai avec les files FIFO si on accepte de perdre 24% des paquets, soit environ le double des pertes DiffServ/DiffServ*. Sachant que la classe BE n'a aucune exigence de qualité de service, surtout pas en ce qui concerne le délai de ses paquets, il est plus satisfaisant, dans la situation actuelle, d'avoir un délai supplémentaire de 30 ms ou 35 ms par paquet que de perdre la moitié des paquets envoyés. L'utilisation de DiffServ ou DiffServ* est donc préférable. La mesure de la gigue nous permettra certainement d'affirmer que DiffServ* est la politique qui garantit, au trafic données, des performances optimales.

4.3.3.3 Mesure de la gigue

La figure 4.25 montre la variation de la gigue du trafic données en fonction du volume de trafic envoyé et de la politique de service. Cette figure confirme nos attentes en prouvant que la qualité de la protection offerte par DiffServ* se mesure sur la gigue des paquets. Les raisons de cet état de chose sont expliquées dans la section 4.2 et les sous-sections précédentes.

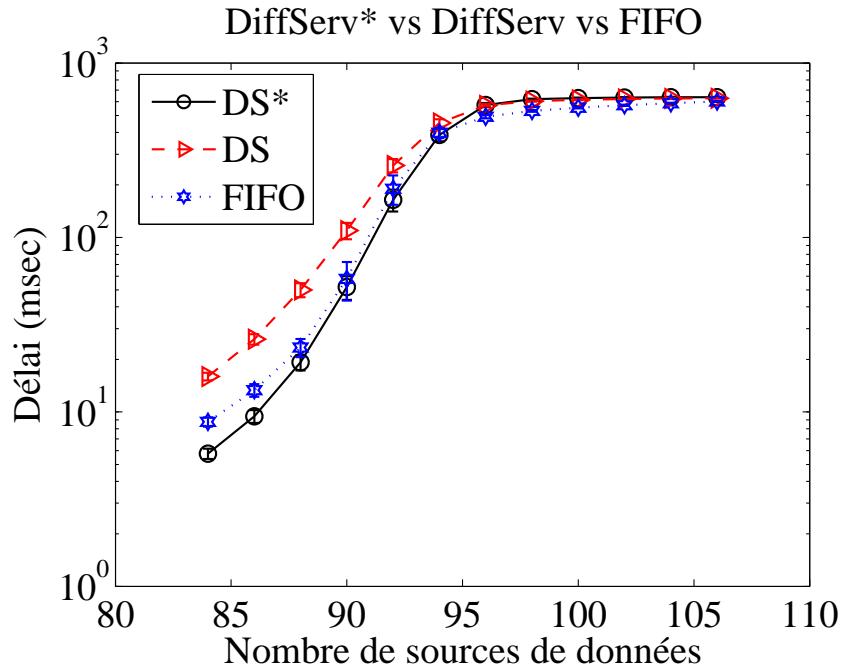


Figure 4.24 Réseau de 5 noeuds : Délai moyen du trafic données en fonction de la politique de service.

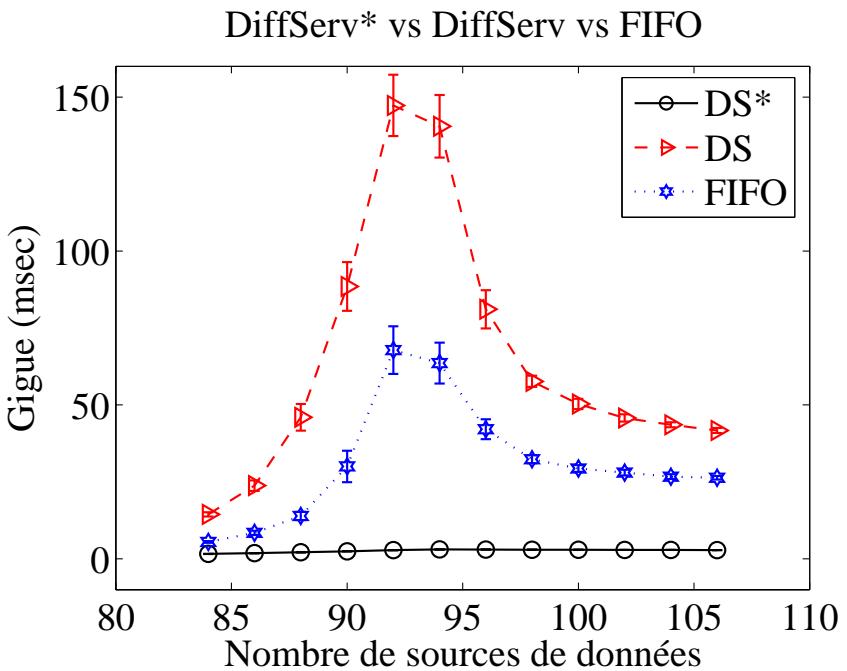


Figure 4.25 Réseau de 5 noeuds : Gigue du trafic données en fonction de la politique de service.

4.3.4 Étude du compromis entre DiffServ* et FIFO

L'avantage de l'utilisation de *DiffServ** par rapport aux files FIFO apparaît encore sur les figures 4.26 à 4.28.

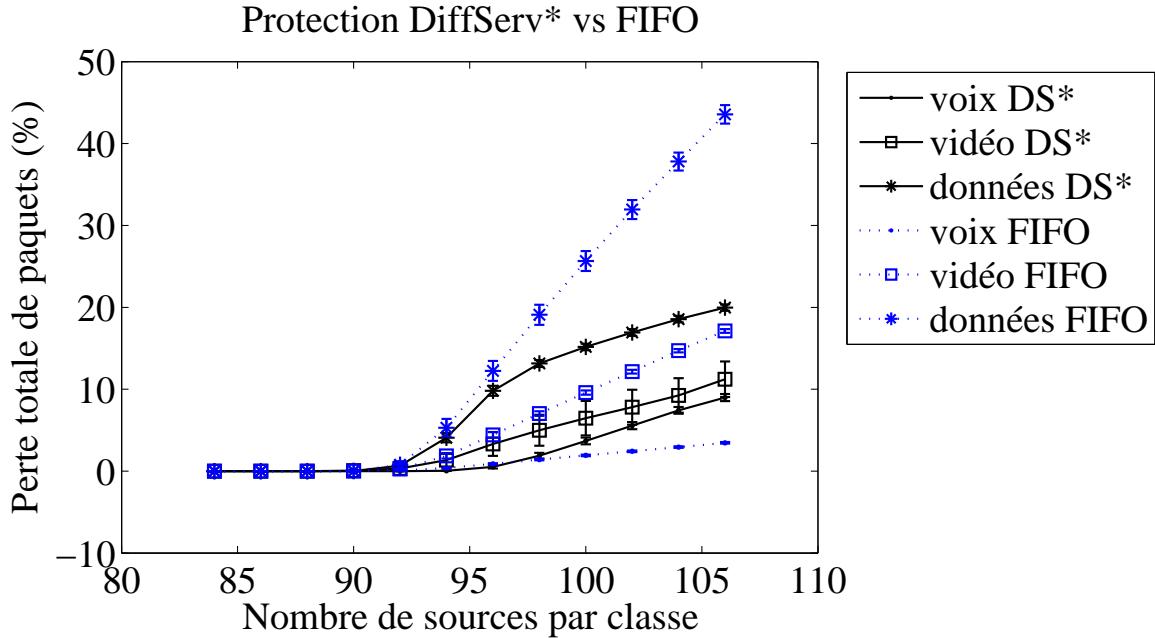


Figure 4.26 DiffServ* vs FIFO : Perte moyenne dans un réseau de 5 nœuds

L'effet de l'utilisation de plusieurs chemins n'est pas évident puisqu'on compare, ici, des politiques de gestion de file d'attente totalement différentes. Les conclusions issues de l'analyse de ces figures sont donc semblables à celles des sections précédentes.

Dans le présent cas, l'utilisation des files FIFO mesurées en octets semble offrir une meilleure protection au trafic de voix car les paquets de cette classe sont de très petite taille et peuvent être facilement stockées dans les buffers. Les paquets de vidéo et de données (classes AF et BE respectivement), plus gros, sont par contre plus difficiles à insérer dans les buffers lorsque le trafic augmente. Cela donne lieu aux forts taux de perte observés pour les trafics de vidéo et de données sur la figure 4.26. Par ailleurs, ces forts taux de perte pour la vidéo et les données sont le prix à payer pour offrir de la différenciation de service en utilisant des files FIFO. Ainsi pour réduire de 6% les pertes du trafic VoIP, celles du trafic vidéo ont augmenté de presque 6% et celles du trafic données d'à peu près 24% soit environ le quadruple du gain réalisé. Ce compromis coûte donc très cher au trafic données.

D'autre part, la différenciation apparente que l'on observe sur la figure 4.26 disparaît sur les figures 4.27 et 4.28 étant donné que la politique FIFO n'est pas conçue pour faire de la différenciation mais simplement pour servir les paquets selon leur ordre d'arrivée. Puisque le

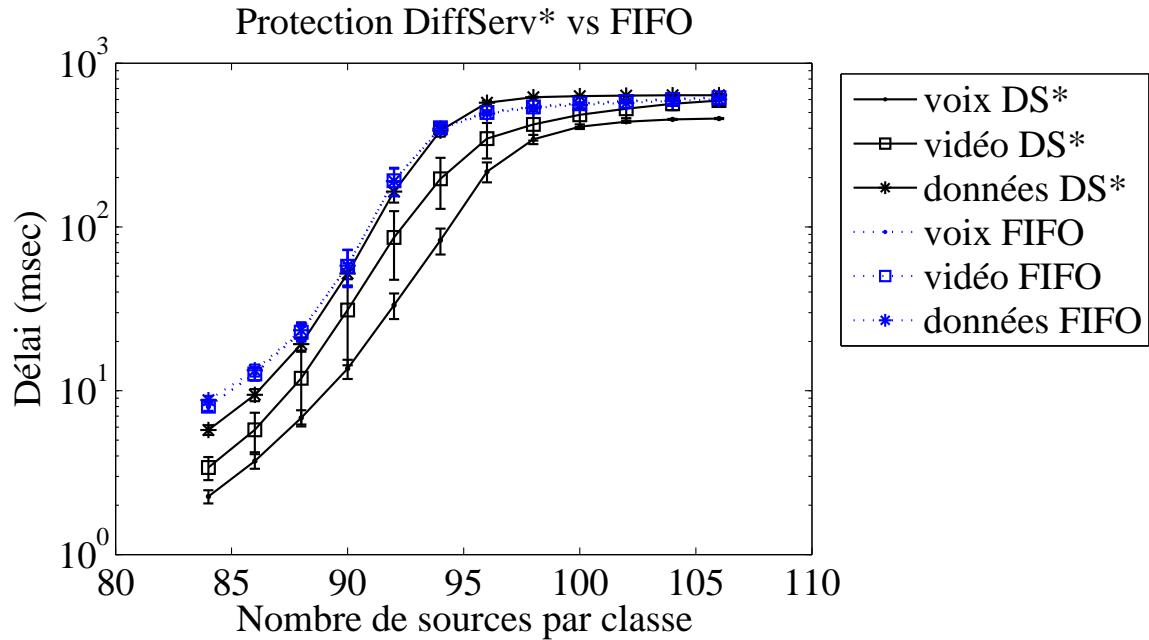


Figure 4.27 DiffServ* vs FIFO : Délai moyen dans un réseau de 5 noeuds

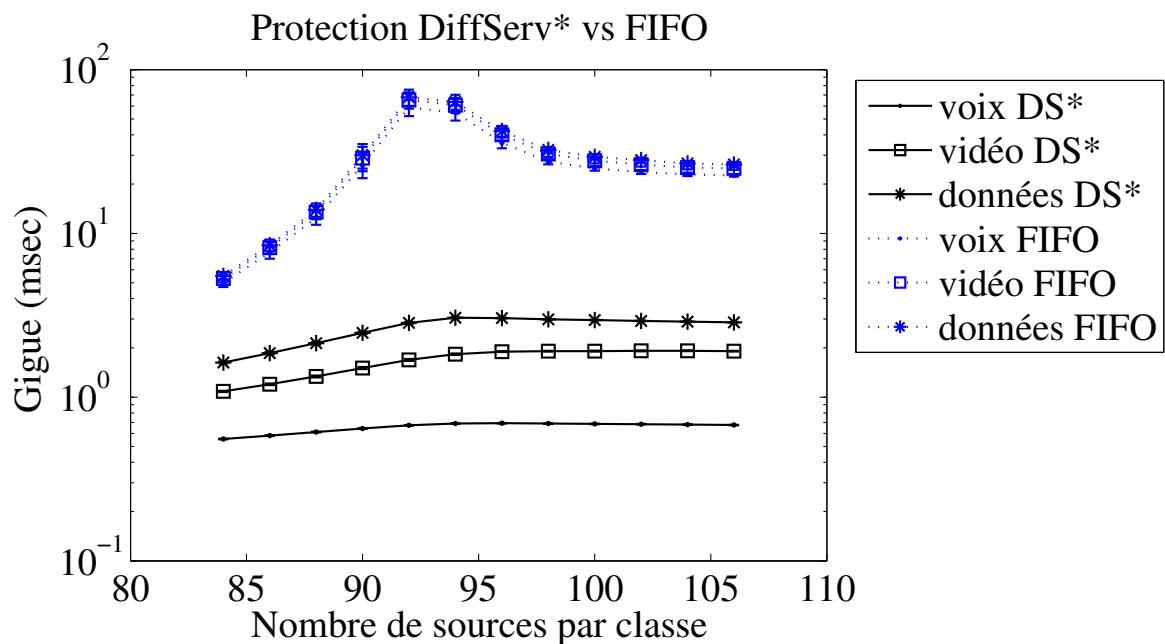


Figure 4.28 DiffServ* vs FIFO : Gigue moyenne dans un réseau de 5 noeuds

compromis réalisé entre les pertes occasionnées par le mécanisme FIFO coûte beaucoup trop cher au trafic données et que ce compromis n'a aucune incidence sur les autres métriques, l'utilisation de DiffServ* paraît l'option la plus juste dans le réseau. Bien que les pertes du trafic VoIP augmente de 6%, DiffServ*, réduit de 6% les pertes du trafic vidéo, de 24% les pertes du trafic données et garantit une différenciation du délai moyen et de la gigue des trois classes de service.

Comme dans la sous-section 4.2.4, nous pouvons conclure que la différenciation de services permet une répartition équitable des ressources du réseau et de toujours assurer au trafic EF de meilleures performances (perte, délai et gigue) par rapport aux deux autres trafics.

4.4 Sommaire

Dans ce chapitre, nous avons expliqué le fonctionnement et l'implémentation de *DiffServ** puis procédé à des tests par simulation. Nous avons montré qu'en déployant l'ordonnanceur DiffServ* dans la couche IP de façon à interagir avec le module de routage et en choisissant l'interface de sortie des paquets en tenant compte de l'état du canal, *DiffServ** améliore les performances d'un réseau en diminuant notamment la gigue moyenne par paquet mais aussi le taux de perte des paquets. De plus, lorsque le réseau n'est pas saturé, *DiffServ** réduit aussi le délai moyen par paquet, comparativement au modèle DiffServ standard. Par ailleurs, l'utilisation combinée de *DiffServ** et du routage par plusieurs chemins physiques réduit d'environ 1% les pertes causées par le modèle DiffServ standard.

D'un autre côté, en comparant les performances de *DiffServ** avec les performances des files d'attente FIFO, nous avons vu que la différenciation permet aussi de diminuer le délai moyen du trafic des classes EF et AF, d'assurer à toutes les classes de trafic l'accès aux ressources et de réguler les pertes par classe tout en priorisant le trafic EF.

Dans ce chapitre, les poids de l'ordonnanceur *DiffServ** ont été maintenus constants pendant toute la durée des simulations, et ce, malgré les variations du canal. Dans le prochain chapitre, nous introduisons la notion de détection de pannes dans un environnement sans fil et la solution que nous proposons pour réduire l'impact des pannes du canal sans fil.

CHAPITRE 5

WIRELESS-DIFFSERV* ET ADAPTATION DE POIDS

Au cours des tests réalisés dans les chapitres précédents, les poids de l'ordonnanceur WRR ont été choisis de façon à fournir une différenciation minimale entre les trois classes de service. En effet, pendant un cycle du WRR, l'ordonnanceur transmet 20 paquets de voix, 3 paquets vidéo et 1 paquet de données, ce qui représente une occupation de bande passante d'environ 42% pour la voix, 37% pour la vidéo et 21% pour les données. De plus, ces poids ont été maintenus constants malgré les variations du canal sans fil. Ainsi, les dégradations du canal affectent toutes les classes de trafic sans aucune distinction. Cependant, l'avantage principal du modèle *Wireless-DiffServ** est le contrôle de canal qui permet d'assurer en permanence l'acheminement du trafic de haute priorité et d'en garantir la qualité de service. Dans ce chapitre, nous proposons un mécanisme de contrôle de la qualité du canal intégré aux modèles *Wireless-DiffServ** et DiffServ. À cet effet, le regroupement des états de l'AMC permet de définir trois niveaux de qualité et, pour chaque niveau, la bande passante occupée par chaque classe de trafic est fixe. La répartition de la bande passante est telle qu'une différenciation quasi-minimale est assurée uniquement lorsque la qualité du canal est excellente. Lorsque la qualité du canal est mauvaise, le trafic prioritaire reçoit au moins 80% de la bande passante tandis que le trafic le moins prioritaire en reçoit juste 5%. Ainsi, la bande passante allouée au trafic prioritaire diminue progressivement lorsque la qualité du canal s'améliore.

L'objectif principal de ce chapitre est de montrer que le contrôle de qualité des liens permet non seulement d'améliorer la qualité de service du trafic VoIP mais aussi que le déploiement de ce mécanisme avec DiffServ* ajoute un supplément de protection par rapport au déploiement avec DiffServ. La section 5.1 décrit le fonctionnement du mécanisme de contrôle de qualité et les caractéristiques des trois niveaux de qualité. La section 5.2, quant à elle, présente une expression analytique qui permet de calculer les poids du WRR associés à chaque niveau de qualité puis les résultats expérimentaux issus de l'implémentation de cette expression. Ces résultats permettent de vérifier que les proportions de bande passante mesurées expérimentalement sont conformes aux valeurs attendues. Par la suite, dans les sections 5.3 et 5.4 nous évaluons les performances des trois classes de trafic avec un ou des ordonnanceurs à poids variables. La section 5.3 montre les résultats de l'étude d'un lien logique tandis que la section 5.4 montre ceux d'un réseau maillé. Dans la première partie de ces deux sections, une étude comparative est faite entre la technique DiffServ* combinée avec la variation des poids et le mécanisme DiffServ lui aussi combiné avec la méthode de variation des poids. Par la

suite, nous comparons les performances de DiffServ* obtenues avec une pondération statique et les performances obtenues avec une pondération dynamique. Dans la dernière partie des sections 5.3 et 5.4, nous présentons une étude qui montre l'avantage des poids dynamiques dans un environnement où le SNR moyen est variable.

5.1 Contrôle de qualité du canal

L'utilisation du procédé d'AMC dans la couche PHY améliore non seulement l'efficacité spectrale des réseaux sans fil mais nous permet également de définir des niveaux de qualité dans le canal. Il est possible de définir autant de niveaux qu'il y a d'intervalles de SNR ou d'états dans la chaîne de Markov. Cependant, afin de limiter la complexité du problème, nous regroupons certains états du canal de manière à définir seulement trois niveaux de qualité. Chaque niveau est caractérisé par deux seuils de taux de transmission : un seuil minimal et un seuil maximal. Étant donné qu'il nous revient de choisir les valeurs des seuils et les pourcentages de bande passante par classe, nous avons choisi des valeurs qui priorisent le plus possible le trafic VoIP à mesure que le canal se dégrade.

La qualité du canal est déterminée par le taux de transmission offert sur chaque lien physique. Avec l'adaptation de modulation et de codage, ce taux varie dans le temps puisque le signal porté par une onde radioélectrique se dégrade au cours du temps. Rappelons qu'avec le standard 802.11g, le taux de transmission le plus faible est égal à 6 Mb/s et est obtenu avec la modulation BPSK. Ainsi, le canal est considéré "mauvais" lorsque son taux de transmission est supérieur ou égal à 6 Mb/s mais ne dépasse pas 18 Mb/s. Cela correspond aux états 0, 1, 2 et 3 du canal. Lorsque le taux de transmission du lien est supérieur à 18 Mb/s mais au maximum égal à 36 Mb/s, nous supposerons que le canal est "bon". Cela correspond aux états 4 et 5 du canal. Le canal est considéré "très bon" lorsque le taux de transmission du lien est supérieur à 36 Mb/s. Cela veut dire que le canal se trouve dans l'état 6 du canal. Quelle que soit la qualité du lien, nous allons montrer qu'il est possible de toujours garantir au trafic VoIP au moins 50% de la capacité totale disponible en choisissant convenablement les poids du WRR. Ainsi, nous proposons une répartition arbitraire de la bande passante entre les trois classes de trafic en fonction de la qualité mesurée. Lorsque le canal est "très bon", nous garantissons environ 50% de la bande passante totale à la VoIP, la vidéo en reçoit environ 35% et les données 15%. Lorsque le canal est "bon", nous proposons de garantir environ 60% de la bande passante totale à la VoIP, 30% à la vidéo et 10% aux données. Enfin, lorsque le canal est "mauvais", nous devons garantir environ 80% de la bande passante à la VoIP, 15% à la vidéo et 5% aux données. L'importance de modifier la répartition de la bande passante en fonction de la qualité du lien est principalement liée à la qualité de

service souhaitée pour chaque classe et au désir de maintenir les pertes du trafic VoIP le plus faible possible. En effet, et comme nous le verrons plus tard, le WRR est un algorithme qui permet de garantir la fraction de la bande passante allouée à chaque classe de trafic. Ainsi, nous pourrions assigner les poids du WRR de façon à toujours garantir 80% de la bande passante à la VoIP, 15% à la vidéo et 5% aux données. Cependant, avec une telle répartition les trafics de vidéo et de données seront toujours négligés même lorsque la qualité du canal est suffisamment bonne pour acheminer tous les types de trafic. De la même façon, si le trafic VoIP recevait en permanence 50% de la bande passante, sa qualité de service se dégraderait beaucoup (plus de perte et de délai) lorsque le canal est mauvais. La variation de la bande passante est donc un compromis pour assurer en permanence une bonne qualité de service pour la VoIP tout en améliorant les performances des autres trafics lorsque la qualité du canal le permet. Le tableau 5.1 présente le résumé des niveaux de qualité du canal, des états de la chaîne de Markov et des taux de transmission qui leurs sont associés, ainsi que la répartition de la bande passante entre les trois classes de trafic.

Tableau 5.1 Niveaux de qualité et répartition de la bande passante

Qualité	États du canal	Seuils (Mb/s)	EF	AF	BE
Mauvais	{0, 1, 2, 3}	[6, 18]	80%	15%	5%
Bon	{4, 5}	(18, 36]	60%	30%	10%
Très bon	{6}	(36, 54]	50%	35%	15%

Par ailleurs, nous avons expliqué dans le chapitre 4 la relation entre la fréquence Doppler F_d du canal, son temps de cohérence T_c et l'intervalle de variation du canal. Tous les paquets transmis pendant une période de temps égale à $T_c/10$ sont envoyés avec un même taux de transmission car l'état du média physique ne change qu'à intervalle de $T_c/10$. Aussitôt le canal réévalué, les poids du WRR devront être mis à jour en fonction de la qualité observée. Ainsi, les poids du WRR ne sont pas fixes mais peuvent varier au cours du temps en fonction de la bande passante disponible à cet instant.

Dans la suite, nous utiliserons une fréquence de variation de 5 Hz pour le canal, ce qui veut dire que l'état des canaux est mesuré toutes les 3,581 ms et immédiatement après, les poids du WRR seront mis à jour. La fréquence F_w de mise à jour des poids du WRR est donc donnée par

$$F_w = \frac{1}{T_c/10} = \frac{1}{3,581 \text{ ms}} = 279,253 \text{ Hz} \quad (5.1)$$

5.2 Répartition de la bande passante

Dans le chapitre 3, nous avons expliqué comment choisir les poids du WRR de façon à obtenir une différenciation de services où le trafic VoIP est mieux protégé que le trafic vidéo, lui même mieux protégé que le trafic données. Nous expliquons dans cette section comment choisir les poids du WRR de façon à garantir de la bande passante pour chaque classe. Soit θ_i et C_i respectivement le pourcentage de bande passante et la capacité alloués à la classe i et soit C la capacité totale d'un lien (logique ou physique)

$$\theta_i = \frac{C_i}{C}$$

En remplaçant C_i par l'expression de l'équation (2.1), on trouve que

$$\theta_i = \frac{\alpha_i L_i}{\sum_{k=1}^n \alpha_k L_k} \quad (5.2)$$

où α_i est le poids assigné à la classe i et L_i la longueur moyenne (entêtes comprises) des paquets de cette classe. En prenant le rapport de ces pourcentages deux-à-deux, il vient que

$$\begin{aligned} \frac{\theta_{EF}}{\theta_{BE}} &= \frac{\alpha_{EF} L_{EF}}{\alpha_{BE} L_{BE}} \\ \frac{\theta_{AF}}{\theta_{BE}} &= \frac{\alpha_{AF} L_{AF}}{\alpha_{BE} L_{BE}} \end{aligned} \quad (5.3)$$

En supposant que le poids de la classe BE est égal à 1, les poids des deux autres classes sont alors donnés par

$$\begin{aligned} \alpha_{EF} &= \frac{\theta_{EF} \alpha_{BE} L_{BE}}{\theta_{BE} L_{EF}} \\ \alpha_{AF} &= \frac{\theta_{AF} \alpha_{BE} L_{BE}}{\theta_{BE} L_{AF}} \\ \alpha_{BE} &= 1 \end{aligned} \quad (5.4)$$

Cette expression dépend uniquement de la longueur des paquets, de la portion de bande passante par classe et du poids de la classe BE. Ainsi, l'équation (5.4) permet de déterminer les poids à assigner aux trois classes de service en fonction de la qualité du lien, qu'il soit logique ou physique. Étant donné que les quantités α_i doivent être entières, nous allons d'abord implémenter cette équation dans le simulateur puis vérifier la qualité des valeurs obtenues. À cet effet, nous mesurons expérimentalement la fraction de bande passante occupée par chaque classe afin de la comparer au pourcentage réel désiré. Soit N_i le nombre total de paquets de la classe i reçus par la destination et soit T_{sim} la durée d'une simulation. Les capacités C_i et

C sont données par

$$C_i = \frac{N_i L_i}{T_{sim}} \quad (5.5)$$

$$C = \sum_{i=1}^3 C_i$$

Pour valider la sélection de poids donnée par l'équation (5.4) dans le simulateur NS-2, nous évaluons d'abord expérimentalement cette équation puis nous utilisons les poids obtenus pour servir les files d'attentes des trois classes. À la fin de la simulation, nous mesurons la bande passante par classe. Trois scénarios sont étudiés :

1. Le canal est de très bonne qualité,
2. Le canal est de bonne qualité,
3. Le canal est de mauvaise qualité.

Toutes les simulations sont répétées 30 fois afin de calculer les intervalles de confiance à 95%. La charge totale par classe varie entre 8,19 Mb/s (256 sources/classe) et 12,22 Mb/s (382 sources/classe). Les simulations sont réalisées sur un lien logique (voir Figure 5.1) et nous utilisons le modèle DiffServ*.

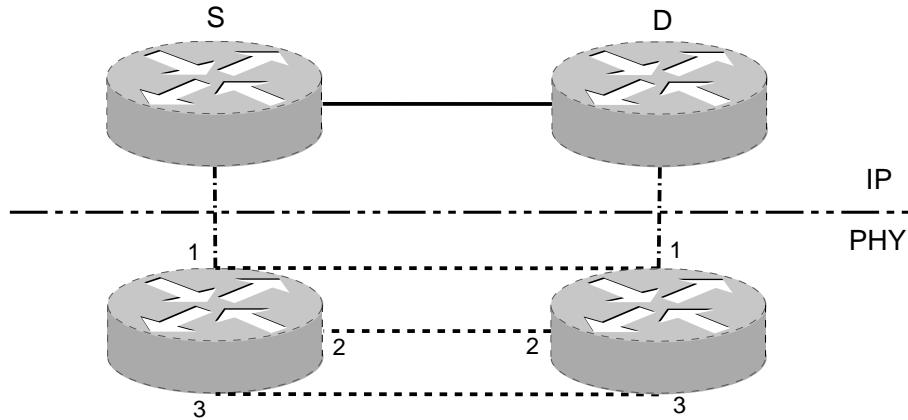


Figure 5.1 Lien logique entre deux noeuds

La qualité du lien physique est dite “très bonne” lorsque son taux de transmission moyen est égal à 54 Mb/s. Dans ce cas, nous garantissons, par hypothèse, environ 50% de la bande passante totale au trafic voix. D'après le tableau 5.1, la répartition souhaitée par classe est

$$\begin{aligned} \theta_{EF} &= 50\% \\ \theta_{AF} &= 35\% \\ \theta_{BE} &= 15\% \end{aligned} \quad (5.6)$$

et les poids du WRR sont

$$\begin{aligned}\alpha_{EF} &= 34 \\ \alpha_{AF} &= 5 \\ \alpha_{BE} &= 1\end{aligned}\tag{5.7}$$

La figure 5.2 montre la bande passante occupée par chaque trafic lorsque tous les liens physiques transmettent à 54 Mb/s. La répartition mesurée est présentée dans le tableau 5.2.

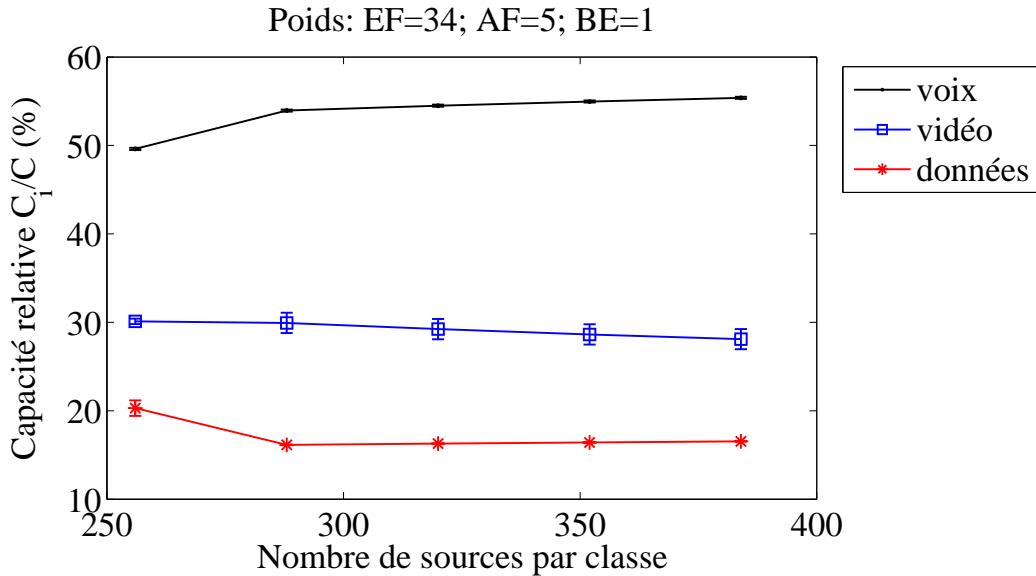


Figure 5.2 Répartition de la bande passante sur un lien “très bon”

Tableau 5.2 Sommaire de la répartition de la bande passante sur un lien “très bon”

Classe	Proportion mesurée (%)
EF	53,68 ± 0,12
AF	29,20 ± 0,98
BE	17,13 ± 0,21

Bien que le trafic vidéo ne reçoive pas le pourcentage (35%) qui devrait lui être dédié, les résultats obtenus sont très satisfaisants car la pondération trouvée ($\alpha_{EF} = 34$, $\alpha_{AF} = 5$, $\alpha_{BE} = 1$) est celle qui permet d’obtenir un écart très faible entre les proportions désirées pour les classes EF et BE et les proportions obtenues par simulation.

Lorsque le taux de transmission moyen du lien physique est supérieur à 18 Mb/s et au plus égal à 36 Mb/s, le lien est dit “bon”. Dans ce cas, d’après le tableau 5.1, la répartition désirée

par classe est

$$\begin{aligned}\theta_{EF} &= 60\% \\ \theta_{AF} &= 30\% \\ \theta_{BE} &= 10\%\end{aligned}\tag{5.8}$$

et les poids du WRR sont

$$\begin{aligned}\alpha_{EF} &= 61 \\ \alpha_{AF} &= 6 \\ \alpha_{BE} &= 1\end{aligned}\tag{5.9}$$

La figure 5.3 montre la bande passante occupée par chaque trafic lorsque tous les liens transmettent à 24 Mb/s. La répartition mesurée est résumée dans le tableau 5.3.

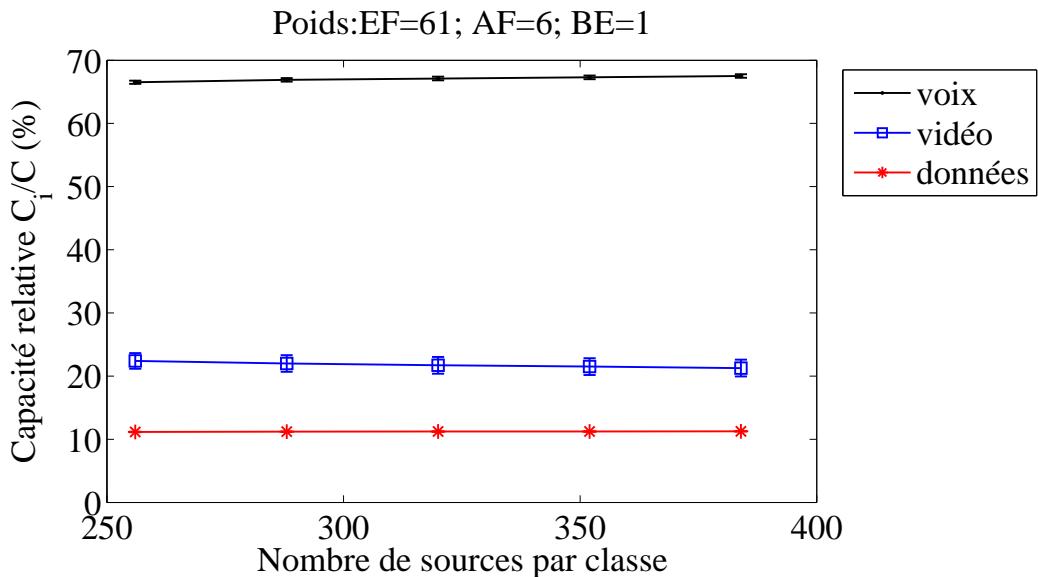


Figure 5.3 Répartition de la bande passante sur un lien “bon”

Bien que la répartition trouvée expérimentalement soit encore un peu défavorable pour la vidéo, elle garantit, toutefois, au trafic VoIP de conserver sa priorité de service face aux deux autres classes.

Tableau 5.3 Sommaire de la répartition de la bande passante sur un “bon” lien

Classe	Proportion mesurée (%)
EF	67 ± 0,28
AF	21,78 ± 1,31
BE	11,22 ± 0,05

Lorsque le taux de transmission moyen du lien est inférieur ou égal à 18 Mb/s, le lien est “mauvais”. Dans ce cas, d'après le tableau 5.1, la répartition souhaitée par classe est

$$\begin{aligned}\theta_{EF} &= 80\% \\ \theta_{AF} &= 15\% \\ \theta_{BE} &= 5\%\end{aligned}\tag{5.10}$$

et les poids du WRR sont

$$\begin{aligned}\alpha_{EF} &= 163 \\ \alpha_{AF} &= 6 \\ \alpha_{BE} &= 1\end{aligned}\tag{5.11}$$

La figure 5.4 montre l'occupation de bande passante pour chaque trafic lorsque tous les liens transmettent à 6 Mb/s. Les proportions mesurées par classe sont résumées dans le tableau 5.4.

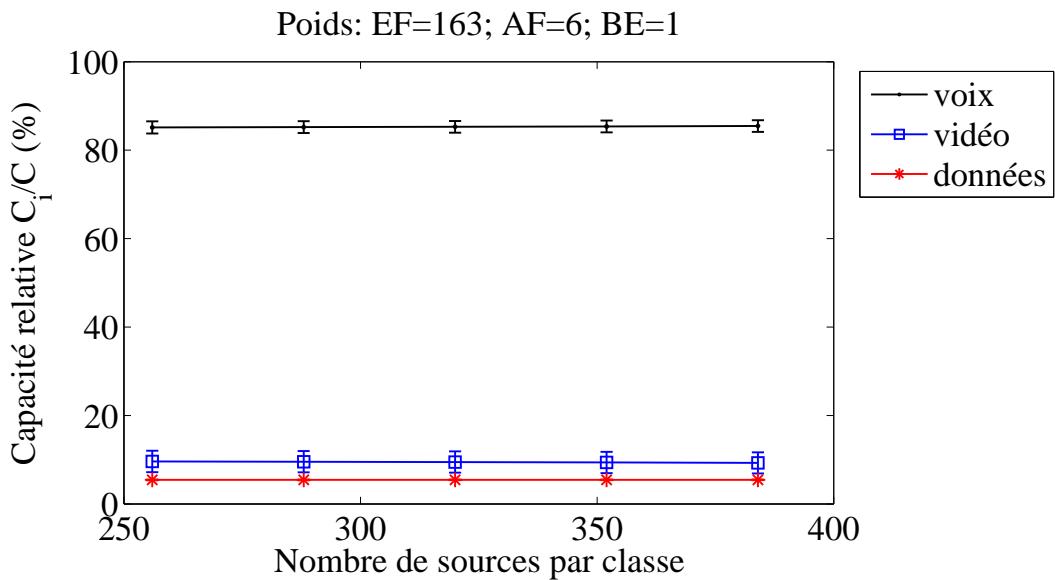


Figure 5.4 Répartition de la bande passante sur un “mauvais” lien

Tableau 5.4 Mesure de la répartition de la bande passante sur un “mauvais” lien

Classe	Proportion mesurée (%)
EF	85,1 ± 1,34
AF	9,46 ± 2,40
BE	5,44 ± 0,08

Une fois de plus, le trafic vidéo ne reçoit pas la part qui lui est due au détriment du trafic VoIP. Cette situation est due aux approximations faites lors de l'implémentation de l'équation (5.4) dans le simulateur. En effet, puisque les poids du WRR doivent être des valeurs entières, il est nécessaire d'arrondir les valeurs décimales à l'entier supérieur ou inférieur. Ce sont donc ces approximations qui affectent les proportions trouvées expérimentalement. Néanmoins ces valeurs expérimentales sont celles qui rapprochent le plus des proportions désirées pour chaque niveau de qualité.

5.3 Performance sur un lien

Dans la section précédente, nous avons défini trois niveaux de qualité et proposé une répartition de la bande passante entre les classes de trafic puis, nous avons vérifié par simulation que l'équation (5.4) permet de trouver la pondération qui donne approximativement les répartitions désirées. Pour cela, pour chaque niveau de qualité, tous les canaux ont été maintenus constants (c'est-à-dire pas d'AMC) afin de mesurer les proportions de bande passante reçues par chaque classe. À présent, l'AMC est de nouveau activée. La fréquence de variation des canaux est 5 Hz, le SNR moyen égal à 15 dB et le PER égal à 10^{-4} .

Le chapitre 4 de ce document a présenté les résultats de performance pour les poids statiques $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$ et nous avions vu que les performances des modèles DiffServ et *DiffServ** sont similaires pour les pertes et le délai alors que la gigue est nettement améliorée avec *DiffServ**. Dans la première partie de cette section, nous comparerons les performances du modèle *DiffServ** avec poids variables et celles de DiffServ également avec poids variables puis dans la deuxième partie nous comparerons la qualité de service offerte par les poids variables et les poids statiques en utilisant uniquement *DiffServ**.

5.3.1 Comparaison DiffServ vs *DiffServ**

Nous comparons les performances de DiffServ et *DiffServ** lorsque les deux modèles utilisent des poids variables. Dans le modèle DiffServ classique, il y a un ordonnanceur par interface donc trois ordonnanceurs par noeud, et chaque ordonnanceur utilise des poids variables en fonction de l'état de leur canal. Les poids de chaque ordonnanceur sont déterminés en fonction de la qualité du canal que l'interface voit. Dans le modèle *Wireless-DiffServ**, il y a un seul ordonnanceur par noeud tandis que trois liens physiques constituent un lien logique. Il nous faut donc déterminer la qualité du lien logique avant de pouvoir calculer les poids du WRR. Quelques possibilités se présentent pour trouver la qualité du lien logique. La plus simple de toutes est de prendre la moyenne des taux de transmission de tous les liens physiques et de la comparer aux seuils du tableau 5.1. Cependant, notre objectif étant

de proposer un mécanisme DiffServ* toujours plus efficace que DiffServ, après analyse de plusieurs tests, nous avons vu que la qualité du lien logique doit être déterminée par celle du lien physique ayant le plus mauvais canal. Ainsi, la qualité du lien logique est celle du lien physique ayant le plus faible taux de transmission.

5.3.1.1 Classe EF – Trafic VoIP

Nous comparons les performances du trafic VoIP avec les modèles DiffServ et DiffServ*. La figure 5.5 montre que le trafic VoIP est très bien protégé par les deux méthodes et n'enregistre presque aucune perte. Les poids de l'ordonnanceur DiffServ* sont alors équivalents aux poids des trois ordonnanceurs DiffServ.

Même si les pertes ne sont pas différencierées, la figure 5.6 montre que les délais le sont tant que le lien n'est pas saturé. En effet, comme nous l'avons expliqué dans le chapitre 4, DiffServ* réduit le délai des trafics lorsque le lien n'est pas saturé car, lorsque toutes les interfaces peuvent recevoir un paquet à transmettre, l'ordonnanceur DiffServ* envoie le paquet vers le canal de meilleure qualité.

Par ailleurs, nous avons aussi montré dans le chapitre 4 que le principal avantage du DiffServ* apparaît sur la mesure de la gigue des paquets. Ainsi, la figure 5.7 confirme nos attentes et prouve que même avec une pondération variable, *DiffServ** permet de maintenir la gigue faible.

5.3.1.2 Classe AF – Trafic Vidéo

Nous comparons les performances du trafic vidéo avec les modèles DiffServ et DiffServ*. Les deux mécanismes utilisent des poids variables. La figure 5.8 montre que les deux modèles produisent encore le même taux de perte. Ce qui n'est pas étonnant étant donné les résultats obtenus dans le chapitre 4. Cependant, les pertes sont plus importantes que celles du trafic VoIP et atteignent près de 80% lorsque 270 sources sont actives.

À faible charge, avec moins de 200 sources vidéo par exemple, la figure 5.9 présente une faible différence de délai en faveur de DiffServ* due à l'utilisation préférentielle du meilleur canal. Cependant, en augmentant le nombre de sources de sorte que le lien commence à saturer, les deux méthodes occasionnent le même délai. Ce comportement a également été observé dans le chapitre 4 et est lié au fait que lorsque la charge devient trop importante, toutes les ressources disponibles sont utilisées quel que soit l'état du canal.

Comme espéré, la figure 5.10 montre que la gigue occasionnée par DiffServ* est bien moins importante que celle de DiffServ.

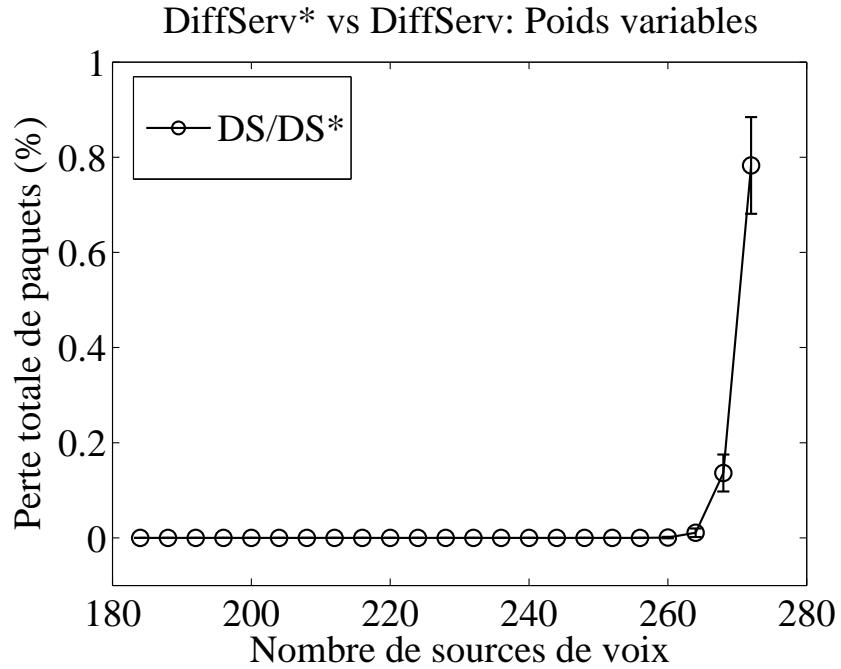


Figure 5.5 Perte du trafic VoIP sur un lien logique avec poids variables : DiffServ vs DiffServ*

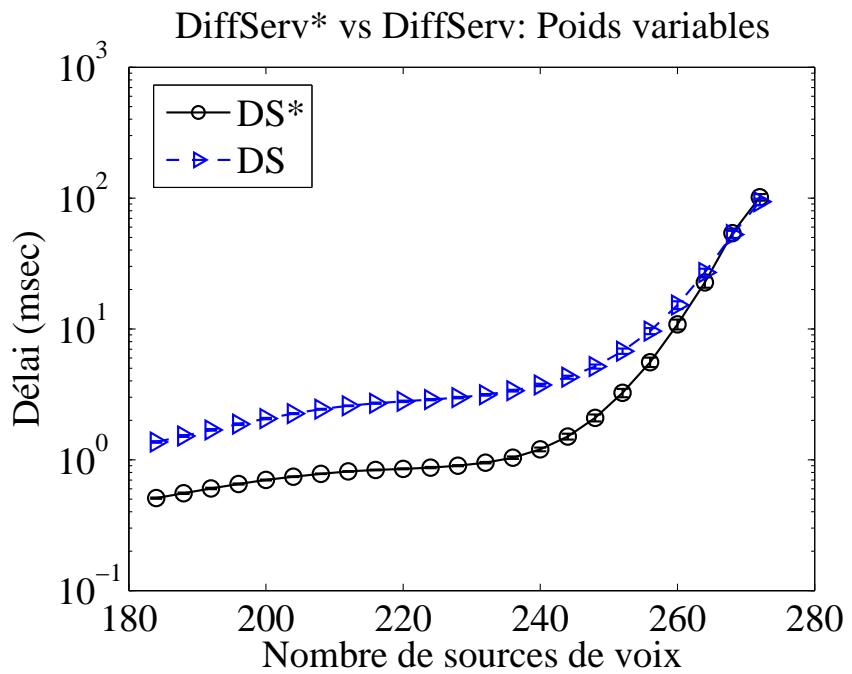


Figure 5.6 Délai moyen du trafic VoIP sur un lien logique avec poids variables : DiffServ vs DiffServ*

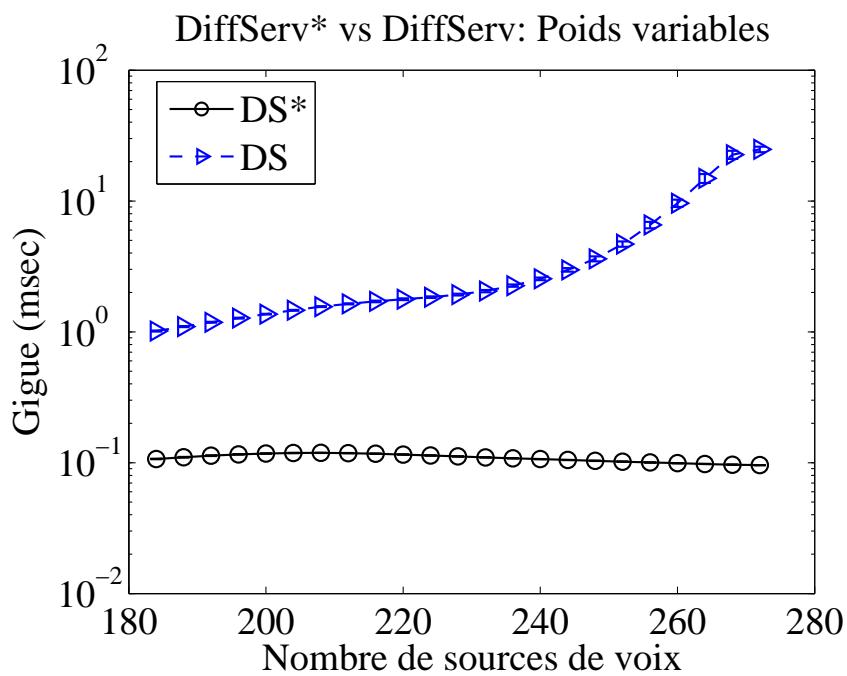


Figure 5.7 Gigue du trafic VoIP sur un lien logique avec poids variables : DiffServ vs DiffServ*

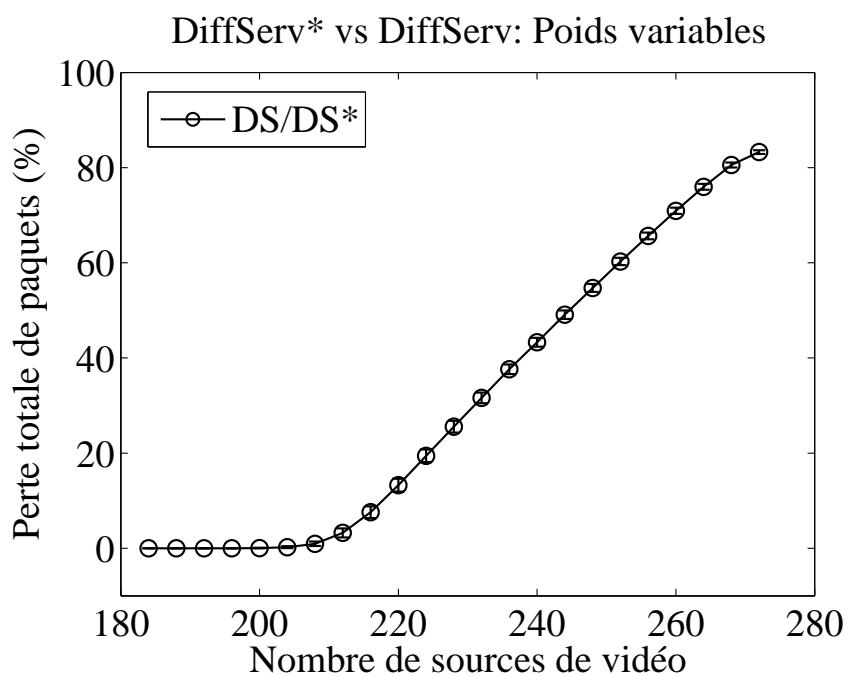


Figure 5.8 Perte du trafic vidéo sur un lien logique avec poids variables : DiffServ vs DiffServ*

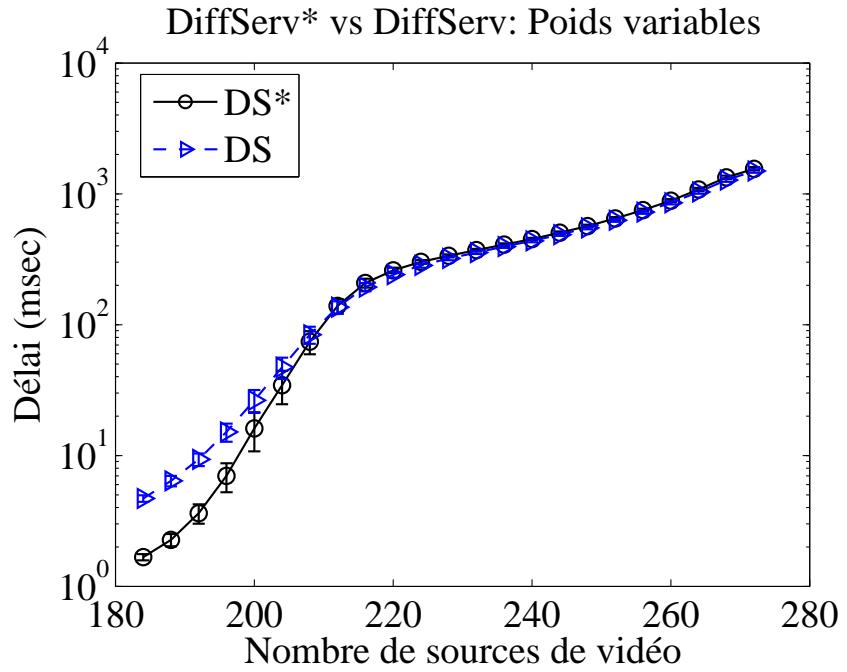


Figure 5.9 Délai moyen du trafic vidéo sur un lien logique avec poids variables : DiffServ vs DiffServ*

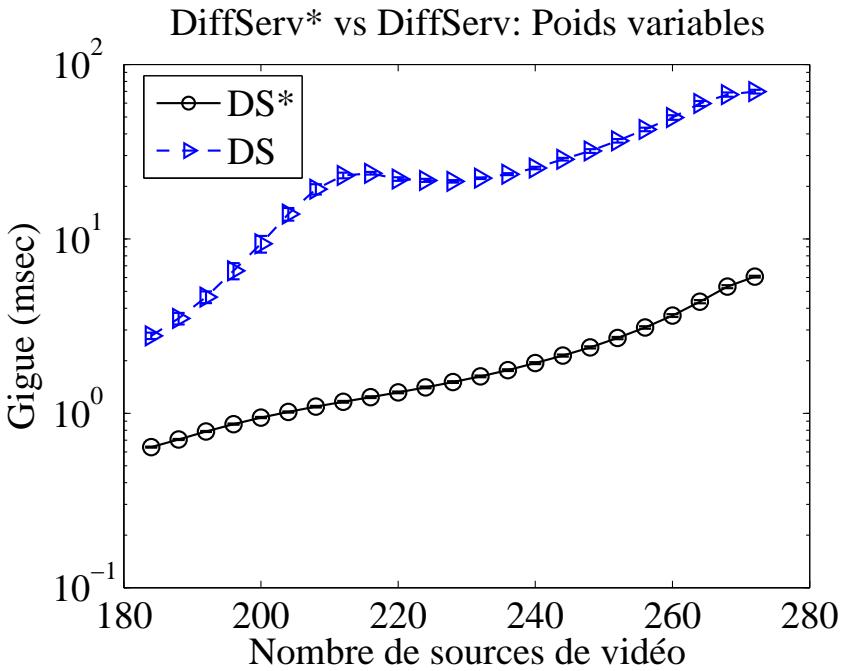


Figure 5.10 Gigue du trafic vidéo sur un lien logique avec poids variables : DiffServ vs DiffServ*

5.3.1.3 Classe BE – Trafic Données

Nous comparons enfin les performances du trafic données avec les modèles DiffServ et DiffServ*. Les conclusions sont les mêmes que celles des deux autres classes. La figure 5.11 montre d'une part, que les deux modèles produisent le même taux de perte et d'autre part, que ce trafic enregistre près de 90% de perte au moment où l'on mesurait (avec 270 sources) à peine 1% pour la VoIP et 80% pour la vidéo. Il est donc clair que les trafics de vidéo et de données sont fortement défavorisés par rapport à la VoIP. C'est donc le prix à payer pour améliorer les performances du trafic VoIP.

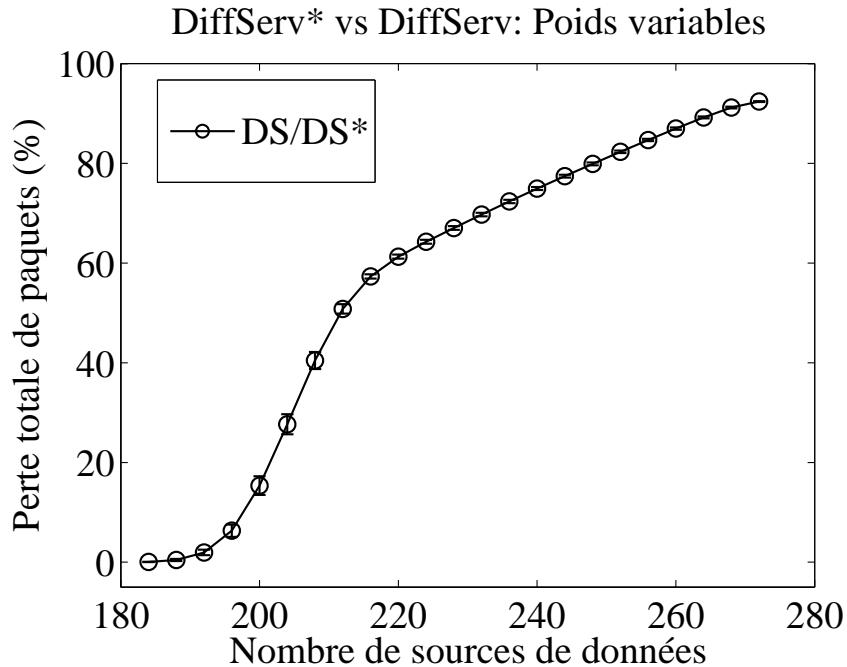


Figure 5.11 Perte du trafic données sur un lien logique avec poids variables : DiffServ vs DiffServ*

L'évolution du délai du trafic données en fonction du nombre de sources est semblable à l'évolution du délai du trafic vidéo. Une très faible différence de délai apparaît sur la figure 5.12 en faveur de DiffServ* lorsque moins de 200 sources sont actives. Cependant, les deux modèles produisent un même délai lorsque le lien est saturé.

En ce qui concerne la gigue, DiffServ* reste sans équivoque un bon régulateur de trafic. En effet, la figure 5.13 montre que la gigue occasionnée par DiffServ* est comme toujours moins importante que celle de DiffServ.

La comparaison des modèles DiffServ et DiffServ* utilisant des poids variables sur un lien logique montre des résultats très semblables aux résultats du chapitre 4. En effet, les deux

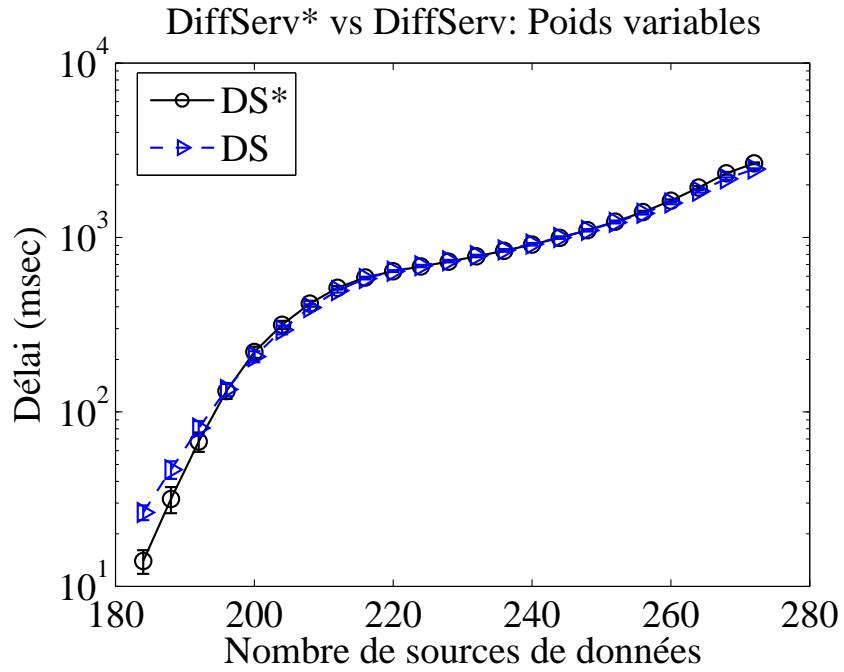


Figure 5.12 Délai moyen du trafic données sur un lien logique avec poids variables : DiffServ vs DiffServ*

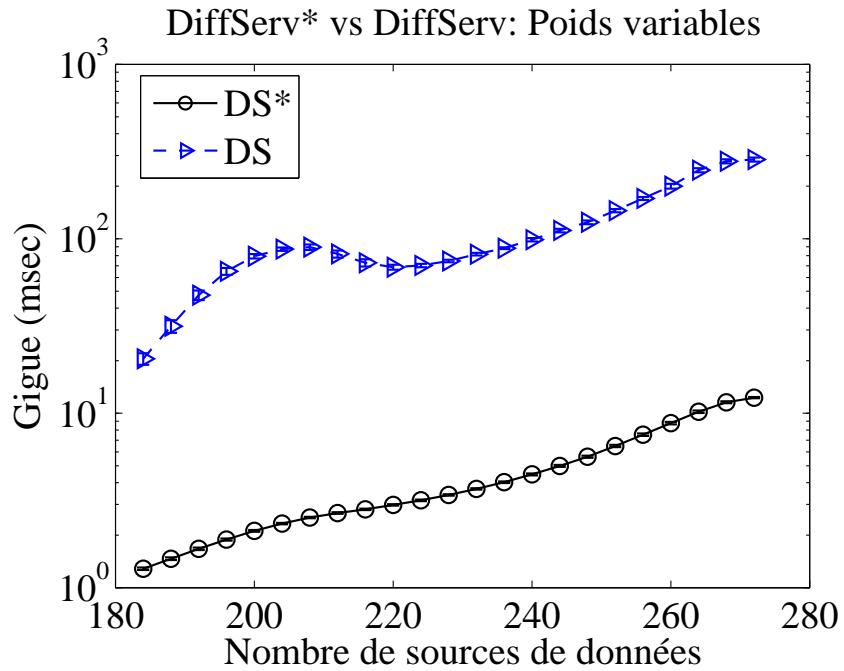


Figure 5.13 Gigue du trafic données sur un lien logique avec poids variables : DiffServ vs DiffServ*

modèles ont des performances similaires en ce qui concerne les pertes et le délai moyen sur un lien fortement chargé, tandis que DiffServ* a un effet bénéfique sur le délai moyen à faible charge. De plus, quelle que soit la charge de trafic, l'utilisation d'un ordonnanceur vaut mieux que trois, car DiffServ* minimise la gigue.

5.3.2 Comparaison poids statiques et poids variables

Nous évaluons, ici, les performances des trois classes de trafic dans un contexte où les poids sont variables et nous les comparons aux performances en régime statique en utilisant le modèle *DiffServ**. La pondération statique utilisée est $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$.

5.3.2.1 Classe EF – Trafic VoIP

Nous comparons l'effet des poids sur les performances du trafic VoIP. La figure 5.14 montre l'évolution des pertes en fonction du volume de trafic écoulé. Jusqu'à 212 sources, il n'y a aucune perte avec les poids variables tandis qu'avec la pondération statique, on mesure approximativement 5% de perte pour une charge de 212 sources. Cela s'explique par le fait que les poids utilisés dans la pondération variable donnent au trafic VoIP un accès plus fréquent au canal.

Sur la figure 5.15, avec les poids variables, le lien n'est pas encore saturé avec 212 sources tandis qu'il l'est avec la pondération statique. Cela suppose que si le lien est capable d'écouler plus de trafic VoIP, il devrait en écouter moins pour les autres classes.

Quant à la figure 5.16, elle montre l'évolution de la gigue en fonction du volume de trafic. Par rapport à cette métrique, les poids variables procurent, encore une fois, une meilleure protection pour ce trafic que les poids fixes.

L'amélioration des performances du trafic VoIP par rapport au cas statique s'explique tout simplement par le fait que les poids variables donnent plus de priorité au trafic VoIP, lui garantissant ainsi un accès plus fréquent au canal. En effet, le triplet le plus petit de la pondération variable est $\alpha_{EF} = 34$, $\alpha_{AF} = 5$ et $\alpha_{BE} = 1$ et puisque $34 > 20$, il va de soi que la voix accède plus souvent au canal avec les poids variables.

5.3.2.2 Classe AF – Trafic Vidéo

La figure 5.17 montre l'évolution des pertes en fonction du volume de trafic vidéo. La vidéo est également mieux protégée avec une pondération variable et on observe une diminution de près de 4% des pertes par rapport aux poids statiques.

De même que les pertes, la figure 5.18 montre que le délai de ce trafic diminue lorsqu'on utilise les poids variables. De plus, comme pour le trafic VoIP, le lien n'est pas encore saturé

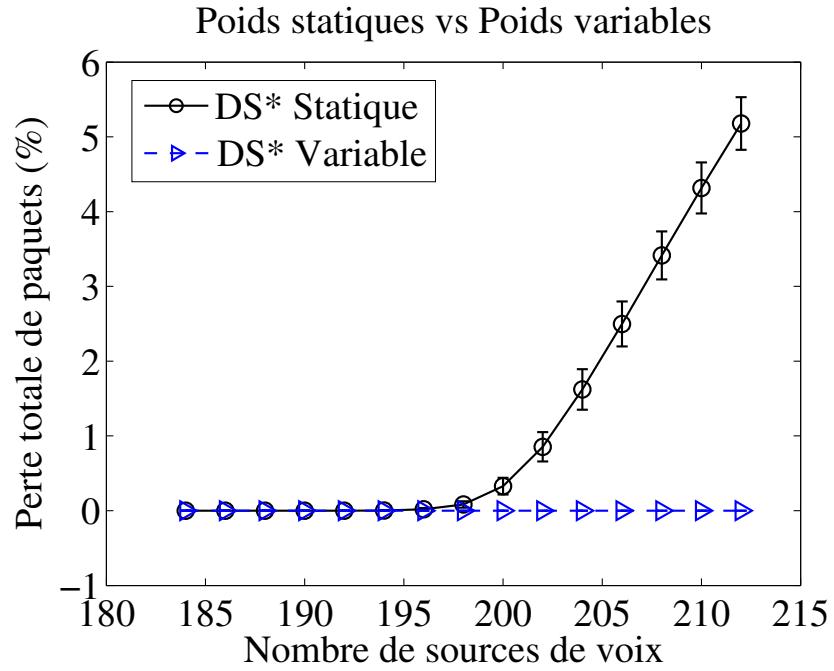


Figure 5.14 Perte du trafic VoIP sur un lien logique : Poids fixes 20-3-1 vs Poids variables

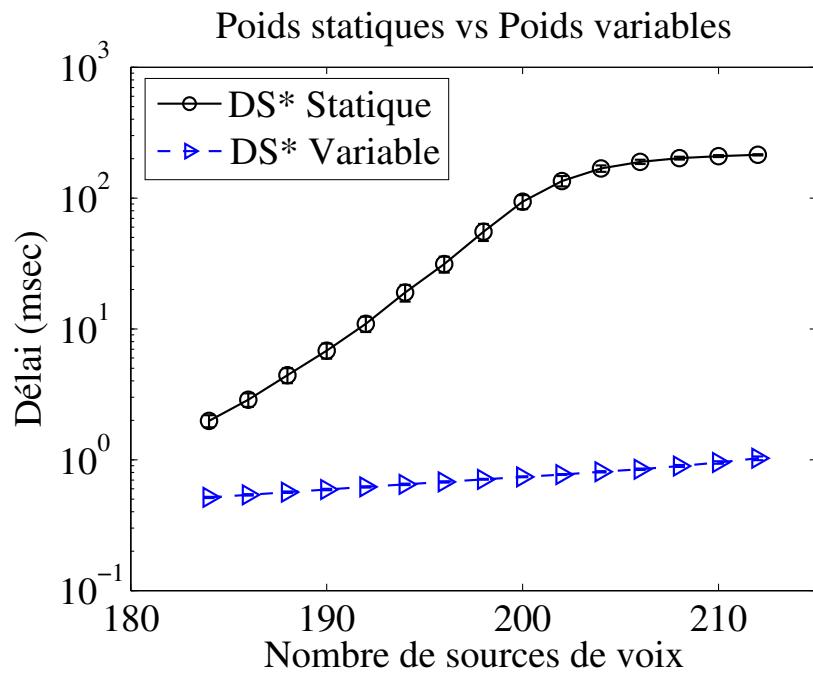


Figure 5.15 Délai moyen du trafic VoIP sur un lien logique : Poids fixes 20-3-1 vs Poids variables

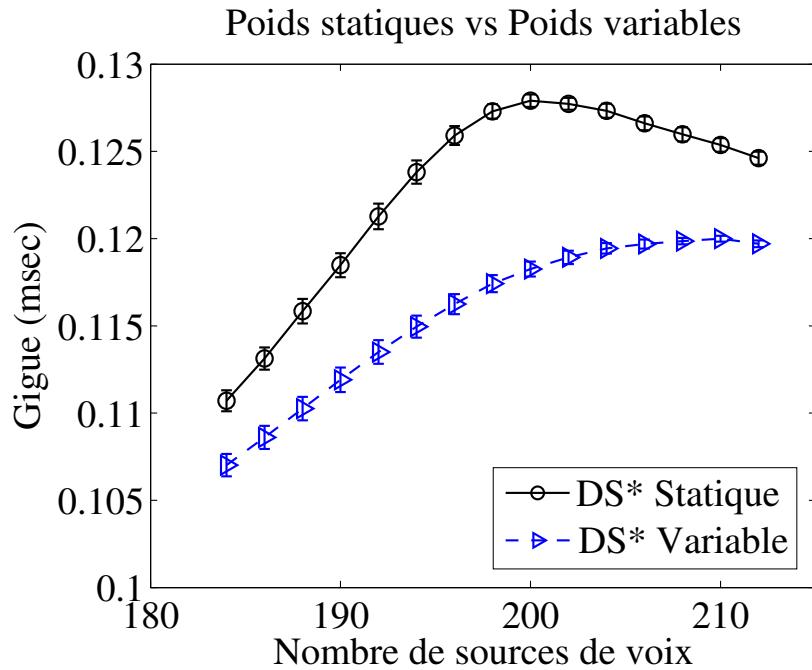


Figure 5.16 Gigue du trafic VoIP sur un lien logique : Poids fixes 20-3-1 vs Poids variables

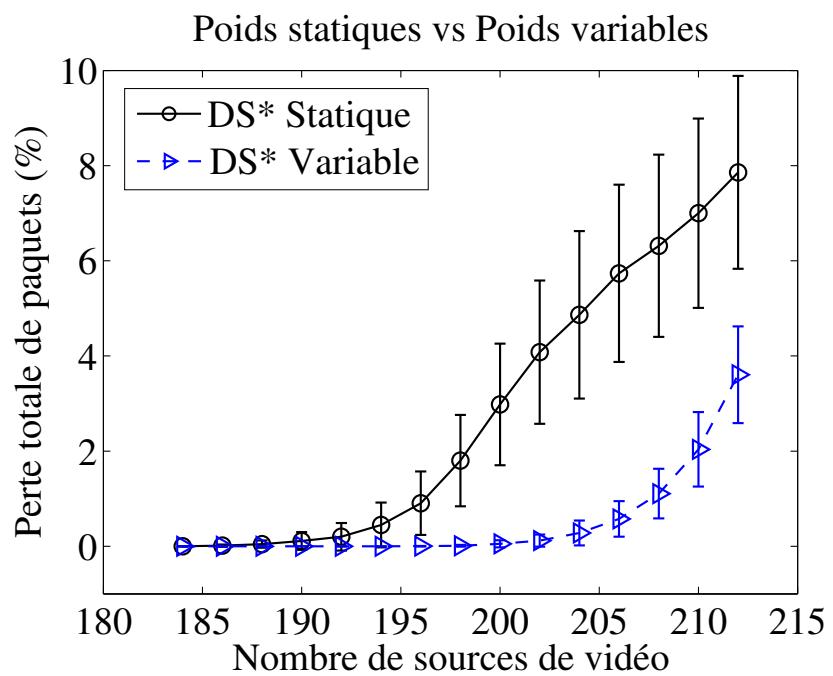


Figure 5.17 Perte du trafic vidéo sur un lien logique : Poids fixes 20-3-1 vs Poids variables

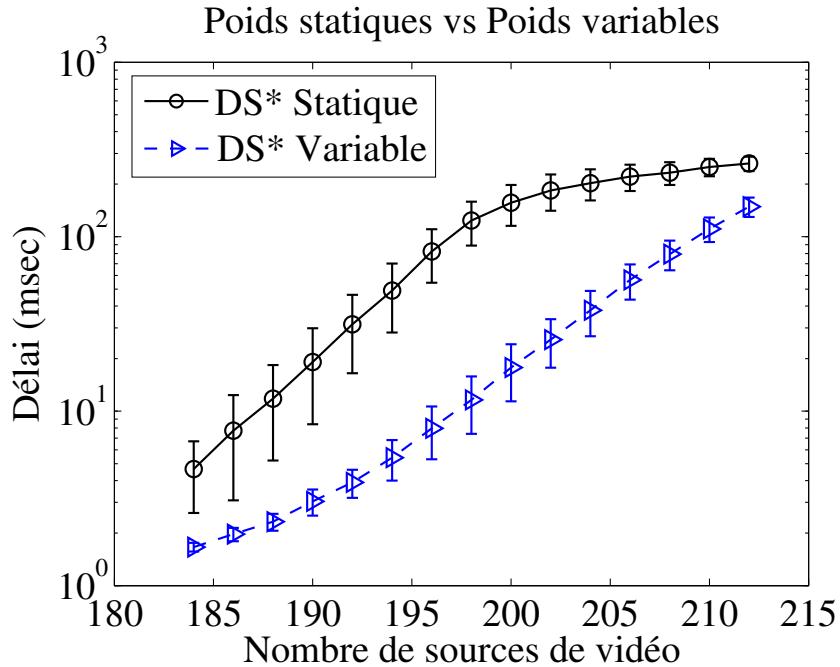


Figure 5.18 Délai moyen du trafic vidéo sur un lien logique : Poids fixes 20-3-1 vs Poids variables

avec 212 sources tandis qu'il l'est avec la pondération fixe $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. La diminution des pertes et du délai du trafic vidéo sur cette plage de mesure est liée à la non saturation du lien logique. En effet, puisque la file du trafic vidéo est servie 5 à 6 fois au cours d'un cycle de WRR soit deux fois plus qu'avec la pondération fixe $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$, il est normal que tant que le lien n'est pas saturé, on mesure moins de perte et moins de délai avec la pondération variable. Cependant, le comportement serait différent une fois le réseau saturé car avec la pondération variable, la vidéo peut recevoir au maximum 35% de la capacité totale alors qu'avec la pondération fixe $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$, elle peut recevoir jusqu'à 37%. Par ailleurs, si le lien est capable d'écouler plus de trafic VoIP et plus de trafic vidéo, cela suppose qu'il devrait écouler moins de trafic données. L'étude des performances de la classe BE permettra de vérifier cette hypothèse.

Contrairement au délai et aux pertes, la figure 5.19 montre que la gigue du trafic vidéo augmente faiblement par rapport à la pondération fixe. Cela est lié au fait que le trafic VoIP soit servi beaucoup plus longtemps avec les poids variables. En effet, en supposant qu'il n'y a pas de retransmission, les paquets du trafic vidéo sont servis après un intervalle de temps T_1 lorsqu'on utilise la pondération statique et après un intervalle de temps T_2 lorsqu'on utilise

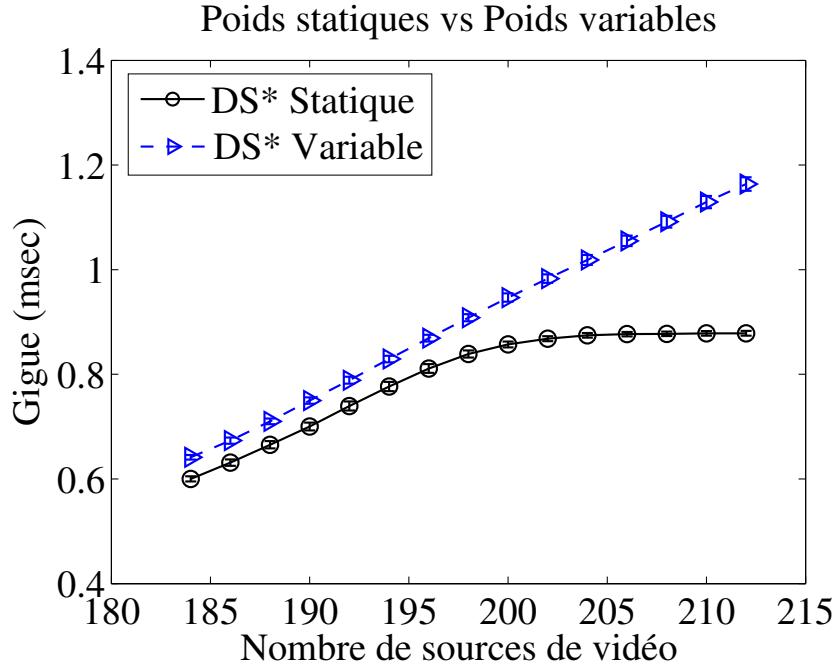


Figure 5.19 Gigue du trafic vidéo sur un lien logique : Poids fixes 20-3-1 vs Poids variables

le plus petit triplet de la pondération variable. Les intervalles T_1 et T_2 sont donnés par

$$\begin{aligned} T_1 &= T_{BE} + 20T_{EF} \\ T_2 &= T_{BE} + 34T_{EF} \end{aligned} \quad (5.12)$$

où T_{BE} et T_{EF} sont les temps de transmission respectifs d'un paquet des classes BE et EF. Comme $T_1 < T_2$, il en résulte une augmentation de la gigue puisque par définition, c'est la variation du délai de transmission de deux paquets consécutifs. L'intervalle T_2 est plus petit avec le triplet $\alpha_{EF} = 34$, $\alpha_{AF} = 5$ et $\alpha_{BE} = 1$.

Par rapport à la gigue, les poids variables procurent une moins bonne protection à ce trafic. Cependant, de manière générale, il ressort de toutes les observations que les poids variables protègent également mieux le trafic vidéo.

5.3.2.3 Classe BE – Trafic Données

Nous comparons l'effet des poids sur les performances du trafic données. La figure 5.20 montre l'augmentation des pertes en fonction du volume de trafic. Comme on pouvait s'y attendre, les pertes pour ce trafic sont beaucoup plus importantes avec les poids variables. En effet, le prix à payer pour améliorer les performances des classes EF et AF consiste à

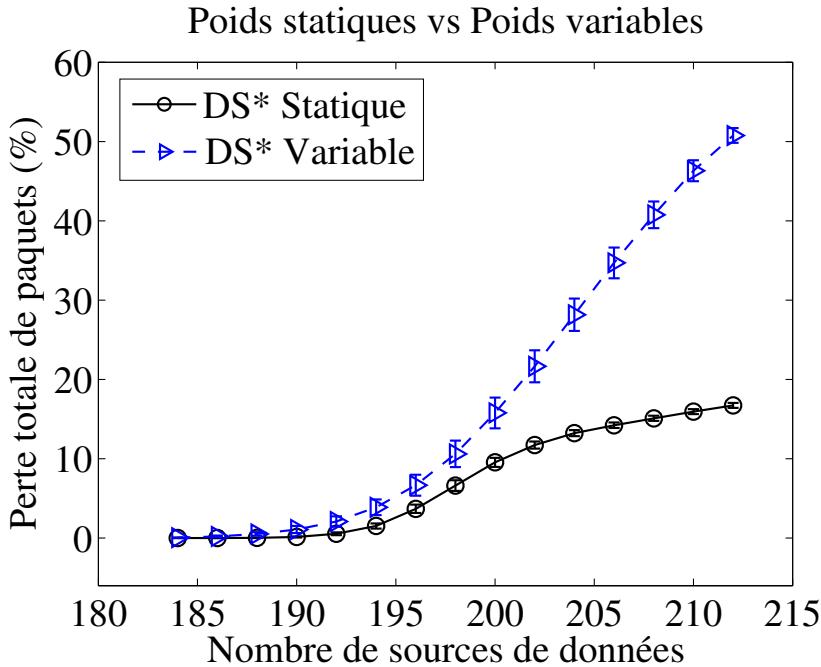


Figure 5.20 Perte du trafic données sur un lien logique : Poids fixes 20-3-1 vs Poids variables

pénaliser la classe BE en réduisant son accès au canal. C'est donc ce que nous faisons avec la pondération variable.

De même que les pertes, le délai du trafic données augmente lorsqu'on utilise les poids variables à cause de la pénalisation de la classe BE. En effet, les ressources étant limitées, l'amélioration des performances des classes EF et AF ne peut qu'entraîner la dégradation de celles de la classe BE. Ainsi, la figure 5.21 montre que le délai de saturation est de l'ordre de 500 ms pour la pondération variable contre 300 ms pour les poids fixes.

La figure 5.22, quant à elle, montre l'évolution de la gigue de ce trafic en fonction de la charge écoulée. Évidemment, cette métrique aussi augmente par rapport aux mesures à poids fixes étant donné que le trafic données accède moins fréquemment au canal. Si T_3 et T_4 désignent les intervalles de temps après lesquels on sert un paquet de données en utilisant respectivement la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$ puis $\alpha_{EF} = 34$, $\alpha_{AF} = 5$ et $\alpha_{BE} = 1$, on a

$$\begin{aligned} T_3 &= 20T_{EF} + 3T_{AF} \\ T_4 &= 34T_{EF} + 5T_{AF} \end{aligned} \tag{5.13}$$

où T_{EF} et T_{AF} sont les temps de transmission respectifs d'un paquet des classes EF et AF. Puisque $T_3 < T_4$, il en résulte une augmentation de la gigue.

En général, les performances de la classe BE se dégradent au profit des deux autres

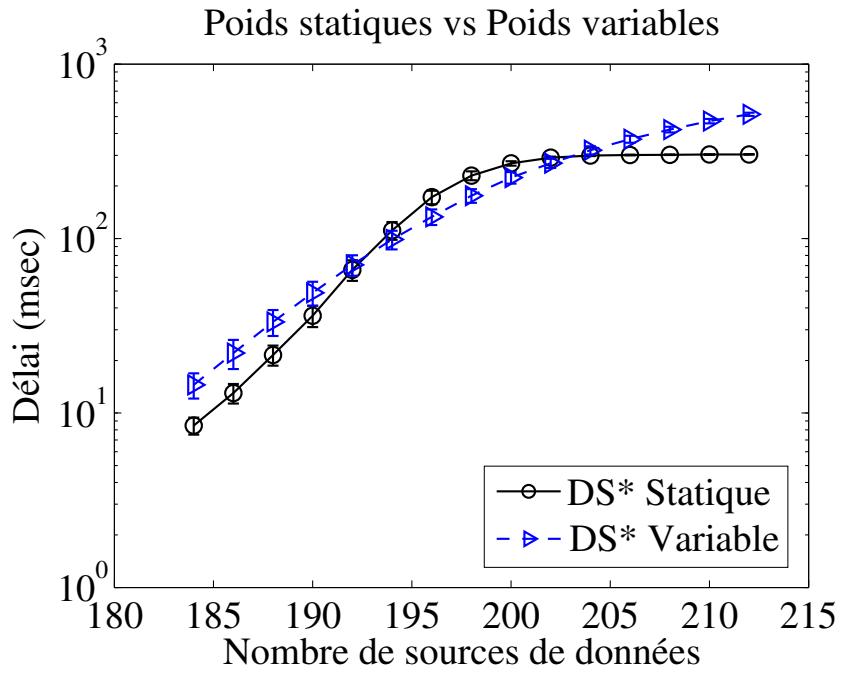


Figure 5.21 Délai moyen du trafic données sur un lien logique : Poids fixes 20-3-1 vs Poids variables

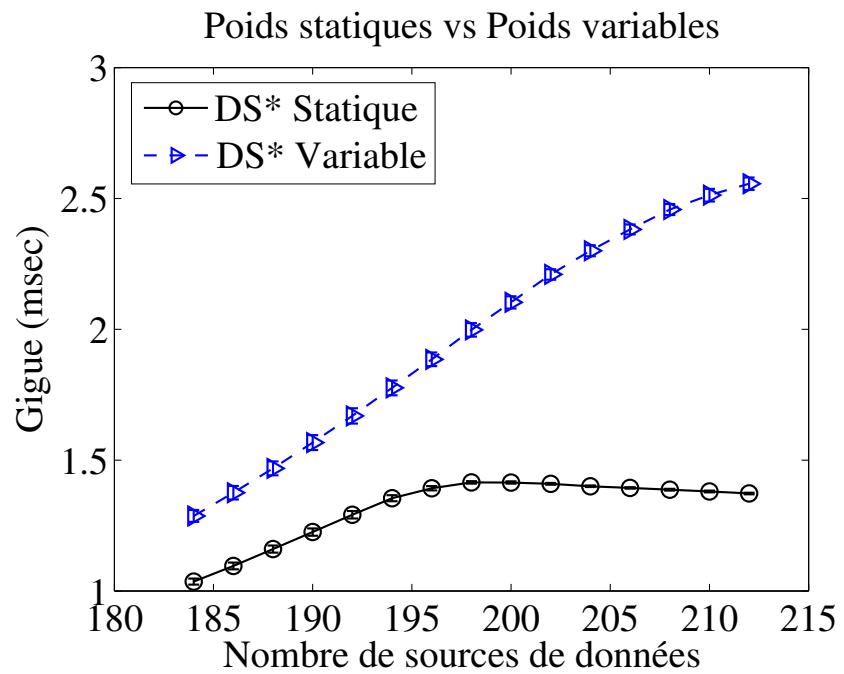


Figure 5.22 Gigue du trafic données sur un lien logique : Poids fixes 20-3-1 vs Poids variables

classes. Il ressort de tout ce qui précède que les poids variables permettent une très bonne différenciation de service par rapport aux poids fixes $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. En effet, puisque la VoIP et la vidéo accèdent plus souvent au canal, leurs performances sont améliorées tandis que celles des données se dégradent.

5.3.3 Utilisation dynamique de la bande passante

Dans la section 5.2, nous avions désactivé le processus d'AMC qui permet de mettre automatiquement à jour le taux de transmission par interface en fonction de la qualité instantanée du lien et nous avions assigné le même état à toutes les interfaces. L'état assigné aux interfaces correspondait au niveau de qualité désirée afin de pouvoir mesurer la bande passante par classe. Cependant, dans la réalité les liens peuvent ne pas être simultanément dans le même état d'évanouissement, bien qu'ils aient le même SNR moyen. Pour cela, nous avons réactivé le procédé d'AMC pour que l'état instantané des canaux varie indépendamment l'un de l'autre mais leur état moyen demeure le même.

L'objectif de cette section est de montrer que les poids variables permettent de garantir de la bande passante non seulement en fonction de l'état instantané des canaux mais aussi en fonction de leur état moyen, c'est-à-dire lorsque le SNR moyen varie. Le SNR moyen est une quantité qui permet d'évaluer la qualité moyenne d'un lien c'est-à-dire qu'elle permet de connaître les états d'évanouissement qui sont prédominants. En effet, nous avons montré dans le chapitre 3 que les probabilités d'état π_k dépendent uniquement du SNR moyen et du PER désiré. De plus, c'est la valeur du SNR moyen qui détermine le débit moyen de chaque lien physique et par conséquent sa qualité. Ainsi, pour vérifier la méthode d'attribution dynamique de la bande passante en fonction de la qualité du canal, on fixe le nombre de sources et on fait varier le SNR moyen de tous les liens de 5 dB à 25 dB puis on mesure la capacité moyenne allouée à chaque classe de trafic. Pour un SNR moyen donné, les canaux sont variables et réestimés à la fréquence F_w soit toutes les 3,581 ms. Le PER moyen est fixé à 10^{-4} . Le lien logique est fortement saturé par 350 sources de trafic par classe soit une charge moyenne de 11,2 Mb/s par classe. Pour chaque valeur de SNR, les simulations sont répétées 30 fois afin de calculer les intervalles de confiance à 95%. Les résultats de cette expérience, réalisée avec DiffServ*, sont présentés sur la figure 5.23. On constate que la bande passante allouée à la voix décroît progressivement lorsque le SNR augmente et inversement pour la vidéo et les données. En effet, à faible SNR par exemple 5 dB, tous les canaux sont en moyenne "mauvais" car il est très difficile d'avoir des taux de transmission de 54 Mb/s avec ce SNR moyen. Ainsi, l'ordonnanceur utilise la plupart du temps les poids $\alpha_{EF} = 163$, $\alpha_{AF} = 6$, $\alpha_{BE} = 1$ qui permettent à la voix de recevoir près de 85% de la capacité disponible, 10% pour la vidéo et 5% pour les données. De même, lorsque la qualité du canal augmente,

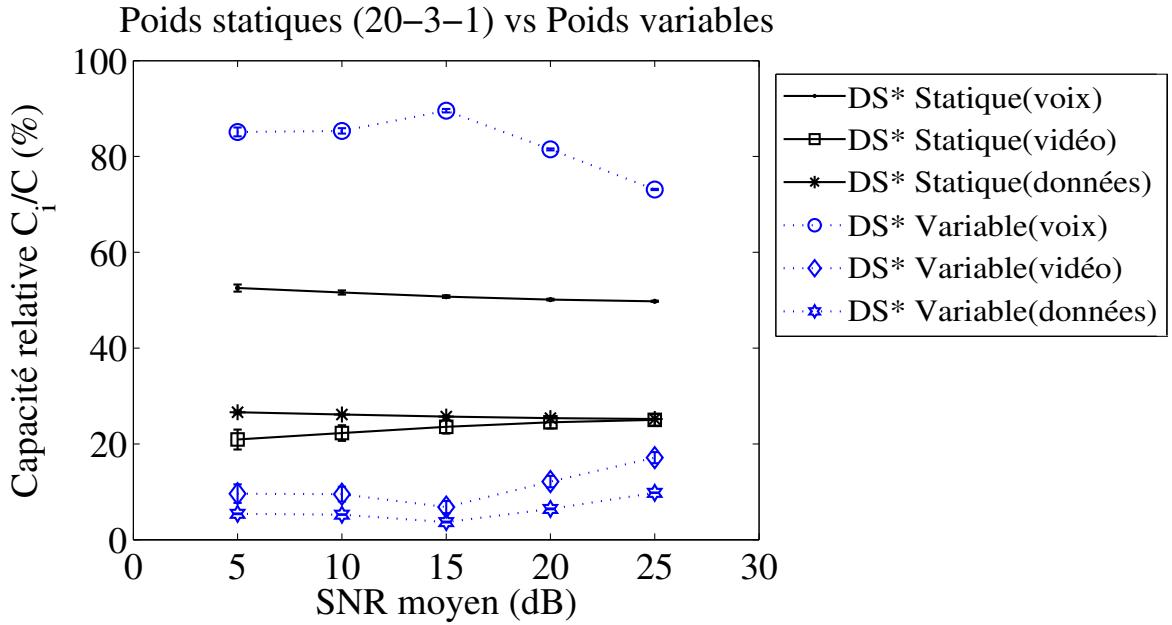


Figure 5.23 Pondération variable vs statique 20-3-1 : capacité relative sur un lien logique

par exemple 25 dB, il est beaucoup plus probable d'avoir des taux de transmission supérieurs à 18 Mb/s donc la qualité du canal varie au cours du temps ce qui permet à l'ordonnanceur d'utiliser différents poids. En particulier à 25 dB, on remarque que la voix reçoit environ 73% de la bande passante, la vidéo 17% et les données 10%. Cette répartition tend vers celle d'un canal de "bonne qualité".

La figure 5.23 montre également que l'utilisation de la pondération minimale $\alpha_{EF} = 20$, $\alpha_{AF} = 3$, $\alpha_{BE} = 1$ ne permet pas de garantir une allocation de bande passante entre les données et la vidéo lorsque le réseau est fortement saturé. En effet, la répartition de la bande passante reste constante, quelle que soit la qualité du canal, et on mesure que le trafic VoIP occupe environ 52%, la vidéo 22% et les données 26% de la capacité disponible. Pourtant, d'après l'équation (5.2) la pondération $\alpha_{EF} = 20$, $\alpha_{AF} = 3$, $\alpha_{BE} = 1$ devrait aboutir à la répartition suivante :

$$\begin{aligned}\theta_{EF} &= 42\% \\ \theta_{AF} &= 37\% \\ \theta_{BE} &= 21\%\end{aligned}\tag{5.14}$$

Bien que la voix reste prioritaire, cette pondération n'assure plus une bonne différenciation entre les autres classes de trafic. Cela est dû au fait que les paquets vidéo sont de taille variable et que cette pondération est la plus faible permettant d'avoir une différenciation

dans le sens désiré, notamment lorsque le lien est faiblement saturé. Les performances du réseau sont donc très sensibles au choix des poids.

5.4 Performance dans un réseau

Nous reprenons l'étude d'un réseau maillé de 5 nœuds (voir Figure 5.24) avec cette fois-ci des poids variables. Tous les liens physiques ont les mêmes propriétés : la fréquence de variation des canaux est 5 Hz, le SNR moyen égal à 15 dB et le PER égal à 10^{-4} . L'objectif de cette section est de montrer les performances de DiffServ et DiffServ* suite au déploiement généralisé du contrôle de qualité sur tous les liens d'un réseau. Dans la première partie de cette section, nous comparons les performances du modèle *DiffServ** avec poids variables et celles de DiffServ également avec poids variables puis dans la deuxième partie nous comparons la qualité de service offerte par les poids variables et les poids statiques en utilisant *DiffServ**.

5.4.1 Comparaison DiffServ vs DiffServ*

Nous comparons les performances de DiffServ et DiffServ* lorsque les deux modèles utilisent des poids variables dans un réseau maillé de 5 nœuds (Figure 5.24). Rappelons qu'en partant du noeud source S , trois (3) routes logiques disjointes peuvent être utilisées pour acheminer les paquets. Il s'agit des routes (S, A, D) , (S, B, D) et (S, C, D) . Il y a donc trois liens logiques qui émanent du noeud S vers les nœuds intermédiaires A , B et C respectivement. Chaque lien logique est composé de 3 liens physiques. Dans le modèle DiffServ*, nous avons défini la qualité d'un lien logique comme étant celle de son "plus mauvais" lien physique. Ainsi, pour trouver les poids de l'ordonnanceur DiffServ* nous évaluons la moyenne de la qualité de tous les liens logiques qui émanent d'un noeud et nous la comparons aux seuils définissant les niveaux de qualité du canal (voir tableau 5.1). Dans le modèle DiffServ classique, chaque ordonnanceur appartient à une interface. Les poids sont donc définis selon l'état de l'interface. Pour connaître l'état d'une interface, on fait la moyenne des taux de transmission de ses canaux physiques. Cette moyenne est ensuite comparée aux seuils définissant les niveaux de qualité du canal.

Illustrons nos propos par un exemple. Supposons qu'à l'instant t_0 , les états des 9 liaisons physiques qui proviennent du noeud S sont présentés dans le tableau 5.5. En examinant le lien logique qui mène vers le noeud A , on remarque que les canaux des interfaces 1 et 2 sont dans l'état 2 donc leur taux de transmission est 12 Mb/s ; le canal de l'interface 3 est dans l'état 3, donc son taux de transmission est 18 Mb/s. De même, on note que les liens qui constituent le lien logique qui mène vers le noeud B sont tous dans l'état 6 et offrent un taux de transmission de 54 Mb/s tandis que les liens qui mènent vers le noeud C sont respectivement dans les états

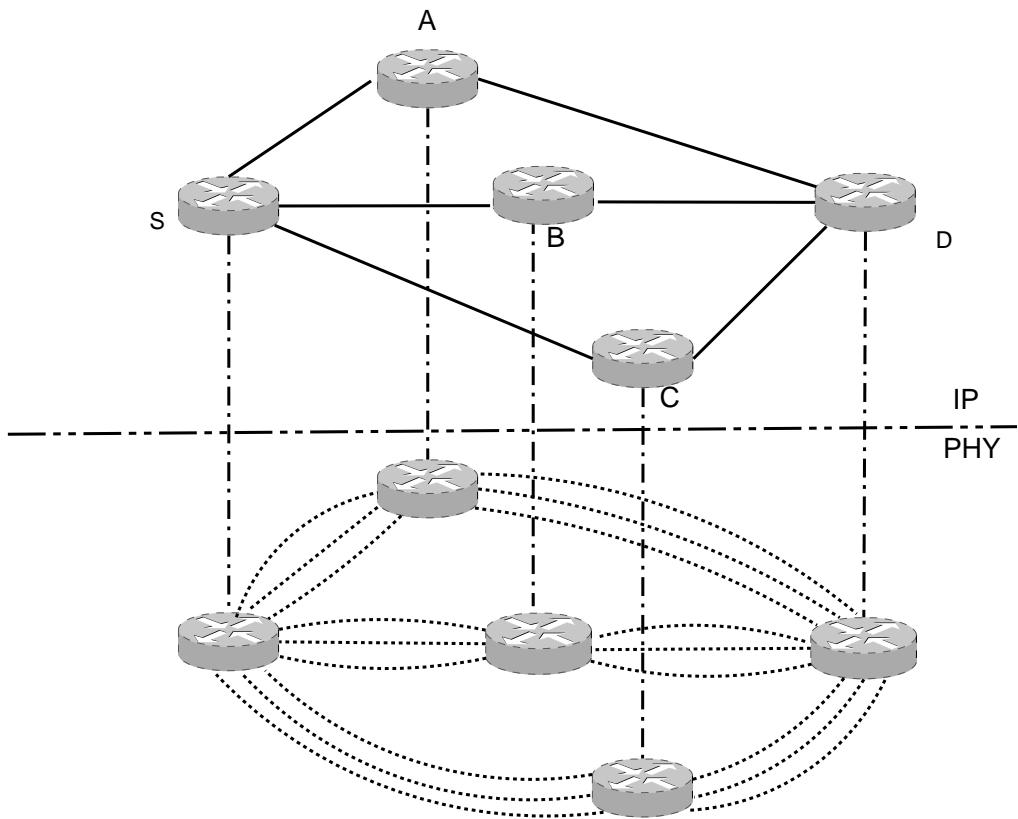


Figure 5.24 Réseau maillé de 5 nœuds

Tableau 5.5 États des canaux du nœud *S* par destination et par interface

		Destinations		
		A	B	C
N° d'interface	1	2	6	4
	2	2	6	5
	3	3	6	4

4, 5 et 4. Soit $I_j(k)$ le taux de transmission de l'interface j qui communique avec le nœud k . On a d'après le tableau 5.5 :

$$\begin{aligned} I_1(A) &= 12 \text{ Mb/s}; & I_1(B) &= 54 \text{ Mb/s}; & I_1(C) &= 24 \text{ Mb/s}; \\ I_2(A) &= 12 \text{ Mb/s}; & I_2(B) &= 54 \text{ Mb/s}; & I_2(C) &= 36 \text{ Mb/s}; \\ I_3(A) &= 18 \text{ Mb/s}; & I_3(B) &= 54 \text{ Mb/s}; & I_3(C) &= 24 \text{ Mb/s}; \end{aligned}$$

Avec le modèle DiffServ*, nous devons déterminer l'état de chaque lien logique afin de déterminer l'état du nœud ainsi que les poids à assigner à l'unique ordonnanceur WRR. Soit $LL(k)$ l'état du lien logique qui mène vers le nœud k .

$$LL(k) = \min \{I_1(k), I_2(k), I_3(k)\} \quad (5.15)$$

L'état du nœud est donné par

$$\bar{LL} = \frac{\sum_k LL(k)}{k} \quad (5.16)$$

Dans cet exemple, on trouve que

$$\begin{aligned} LL(A) &= \min\{12, 12, 18\} = 12 \text{ Mb/s} \\ LL(B) &= \min\{54, 54, 54\} = 54 \text{ Mb/s} \\ LL(C) &= \min\{24, 36, 24\} = 24 \text{ Mb/s} \end{aligned}$$

et

$$\bar{LL} = \frac{12 + 54 + 24}{3} = 30 \text{ Mb/s} \quad (5.17)$$

En se référant au tableau 5.1, un taux de transmission moyen égal à 30 Mb/s correspond à un canal de bonne qualité donc les poids de l'ordonnanceur du modèle DiffServ* sont $\alpha_{EF} = 61$, $\alpha_{AF} = 6$, $\alpha_{BE} = 1$ ce qui permet au trafic VoIP de recevoir approximativement 60% de la capacité, le trafic vidéo 30% et le trafic de données 10%. Cette pondération est utilisée pendant l'intervalle de temps $t' = t_0 + T_c/10$. Au bout du temps t' , l'état de tous les canaux est réévalué et le processus de détermination des poids recommence.

Avec le modèle DiffServ, nous devons calculer le taux de transmission moyen de chaque interface afin de déterminer son état ainsi que les poids à assigner à son ordonnanceur WRR. Soit \bar{I}_j l'état de l'interface j . Cet état est donné par

$$\bar{I}_j = \frac{I_j(A) + I_j(B) + I_j(C)}{3}, \quad j = 1, 2, 3 \quad (5.18)$$

Dans cet exemple, on trouve que

$$\begin{aligned}\bar{I}_1 &= \frac{12 + 54 + 24}{3} = 30 \text{ Mb/s} \\ \bar{I}_2 &= \frac{12 + 54 + 36}{3} = 34 \text{ Mb/s} \\ \bar{I}_3 &= \frac{18 + 54 + 24}{3} = 32 \text{ Mb/s}\end{aligned}$$

Un taux de transmission moyen inférieur ou égal à 36 Mb/s correspond à un canal de bonne qualité donc les poids utilisés par chaque ordonnanceur permettront d'attribuer 60% de la capacité à la VoIP, approximativement 30% à la vidéo et 10% aux données. En d'autres termes, pendant l'intervalle de temps $t' = t_0 + T_c/10$, les trois ordonnanceurs utilisent la pondération $\alpha_{EF} = 61$, $\alpha_{AF} = 6$, $\alpha_{BE} = 1$.

La synthèse des résultats de cet exemple est présenté dans le tableau 5.6. Il en ressort que pendant l'intervalle t' , l'ordonnanceur DiffServ* et les trois ordonnanceurs DiffServ utilisent tous la même pondération $\alpha_{EF} = 61$, $\alpha_{AF} = 6$, $\alpha_{BE} = 1$.

Tableau 5.6 Implémentation des poids variables dans DiffServ et DiffServ*

	Destinations			Taux moyen par interface \bar{I}_j	
	A	B	C		
N° Interfaces	1	12	54	24	30 Mb/s
	2	12	54	36	34 Mb/s
	3	18	54	24	32 Mb/s
Qualité par lien logique $LL(.)$		12 Mb/s	54 Mb/s	24 Mb/s	

5.4.1.1 Classe EF – Trafic VoIP

Nous comparons ici les performances du trafic VoIP avec les modèles DiffServ et DiffServ*, tous deux utilisant des poids variables. La figure 5.25 montre que le trafic VoIP est beaucoup mieux protégé par DiffServ* et enregistre près de la moitié des pertes occasionnées par DiffServ, soit environ une diminution de 7%. Contrairement à l'étude d'un lien, l'avantage de DiffServ* est lié à la procédure de détermination des poids au nœud source, en particulier lorsqu'il y a plusieurs liens logiques incidents.

De même que les pertes, la figure 5.26 montre une très bonne différenciation entre les délais tant que le réseau n'est pas saturé. Cette différenciation va en faveur de DiffServ* car

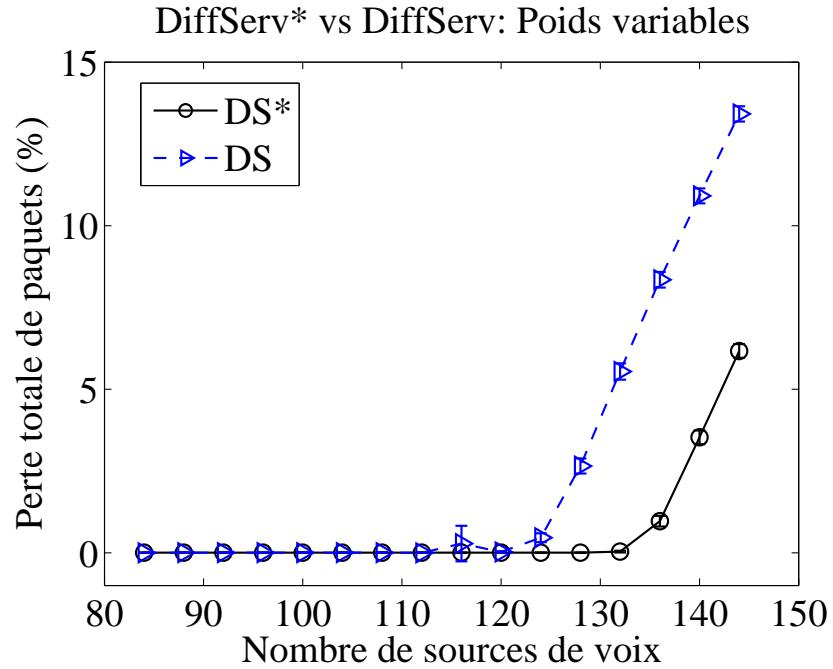


Figure 5.25 Perte du trafic VoIP dans un réseau avec poids variables : DiffServ vs DiffServ*

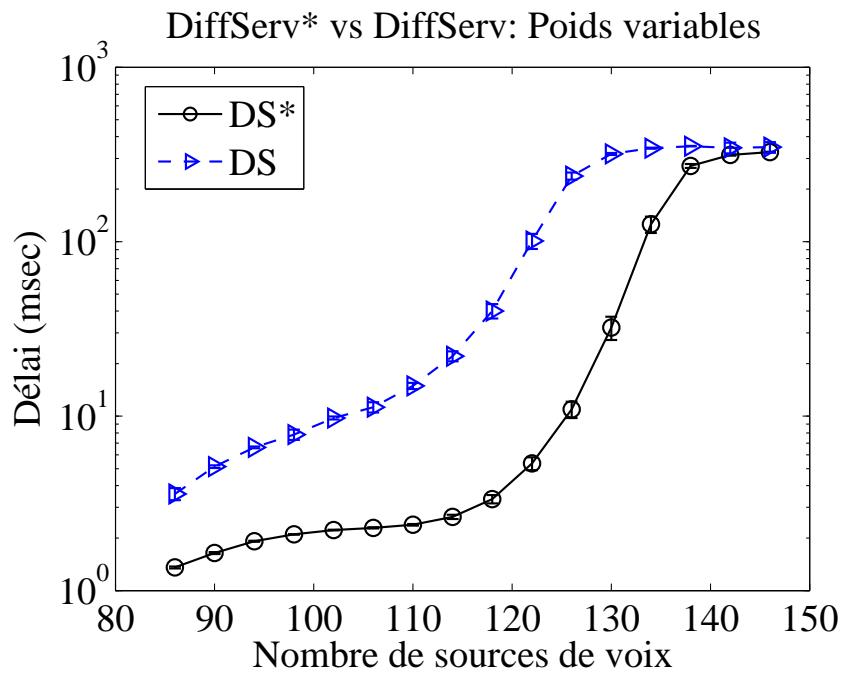


Figure 5.26 Délai moyen du trafic VoIP dans un réseau avec poids variables : DiffServ vs DiffServ*

ce modèle choisit l'interface de meilleure qualité pour transférer les paquets, ce qui réduit le délai des trafics lorsque le réseau est non saturé.

La mesure de la gigue sur la figure 5.27 montre comme toujours que DiffServ* engendre moins de gigue que DiffServ du fait de l'utilisation d'un seul ordonnanceur au lieu de trois.

5.4.1.2 Classe AF – Trafic Vidéo

Nous comparons à présent les performances du trafic vidéo avec les modèles DiffServ et DiffServ*, tous deux utilisant des poids variables. La figure 5.28 montre les pertes occasionnées par les deux modèles. Comme pour le cas d'un lien, les pertes évoluent presque de la même manière tant que le réseau n'est pas saturé et on note une diminution des pertes de 1% grâce à DiffServ*. Cependant, la méthode de détermination des poids variables modifie le comportement du réseau lorsque ce dernier est saturé. En effet, le gain en perte observé pour le trafic VoIP grâce à DiffServ* se traduit ici par une dégradation plus importante des pertes du trafic vidéo. Avec DiffServ*, la vidéo enregistre alors 20% plus de perte que DiffServ. Cette différence de comportement est liée à l'implémentation des poids variables dans les deux modèles.

À faible charge, par exemple moins de 100 sources, la figure 5.29 présente une faible différence de délai en faveur de DiffServ*. Cela est dû au fait que DiffServ* envoie toujours de préférence les paquets vers le meilleur canal. En augmentant le trafic, les deux méthodes occasionnent le même délai lorsque le réseau commence à saturer. Ce comportement a déjà été observé quand on comparait les performances de ce trafic sur un lien. Cependant, une fois saturé, le comportement des deux mécanismes diffèrent et comme on peut le voir, DiffServ* protège moins bien la vidéo puisqu'il privilégie la voix.

Quelle que soit l'évolution des pertes et du délai, l'utilisation d'un seul ordonnanceur, DiffServ*, diminue toujours la gigue par rapport à trois ordonnanceurs, DiffServ. C'est ce qui ressort encore de la figure 5.30.

5.4.1.3 Classe BE – Trafic Données

Nous comparons enfin les performances du trafic données avec les modèles DiffServ et DiffServ*. Les deux mécanismes utilisent des poids variables. La figure 5.31 montre l'évolution des pertes du trafic données en fonction du volume de trafic. On note d'une part une rapide augmentation des pertes et d'autre part un comportement semblable au trafic voix. En effet, DiffServ* produit 1% moins de perte que DiffServ tant que le réseau n'est pas saturé par ce trafic. Toutefois lorsque le réseau est saturé, DiffServ produit environ 6% moins de perte que DiffServ*. Cette dégradation des performances de DiffServ* est aussi due au gain de 7%

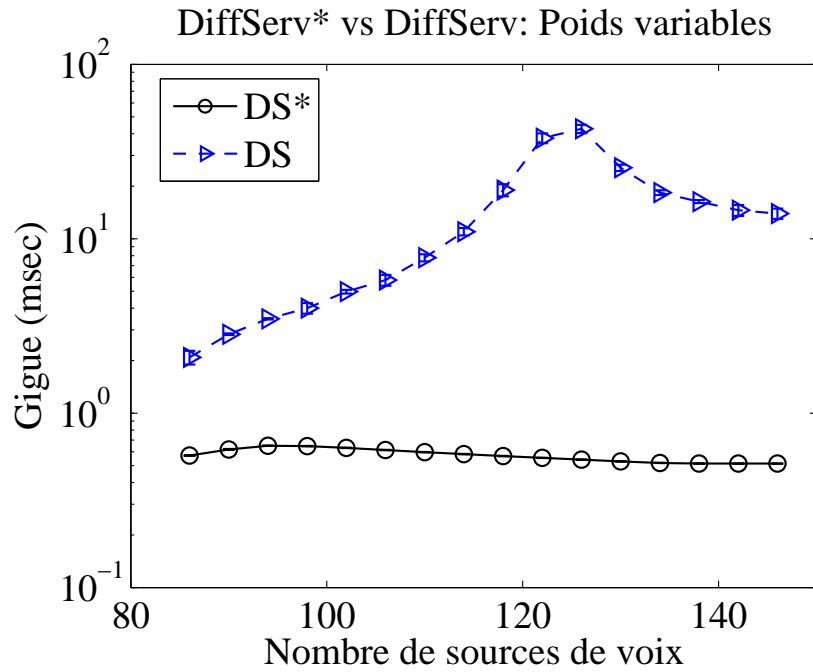


Figure 5.27 Gigue du trafic VoIP dans un réseau avec poids variables : DiffServ vs DiffServ*

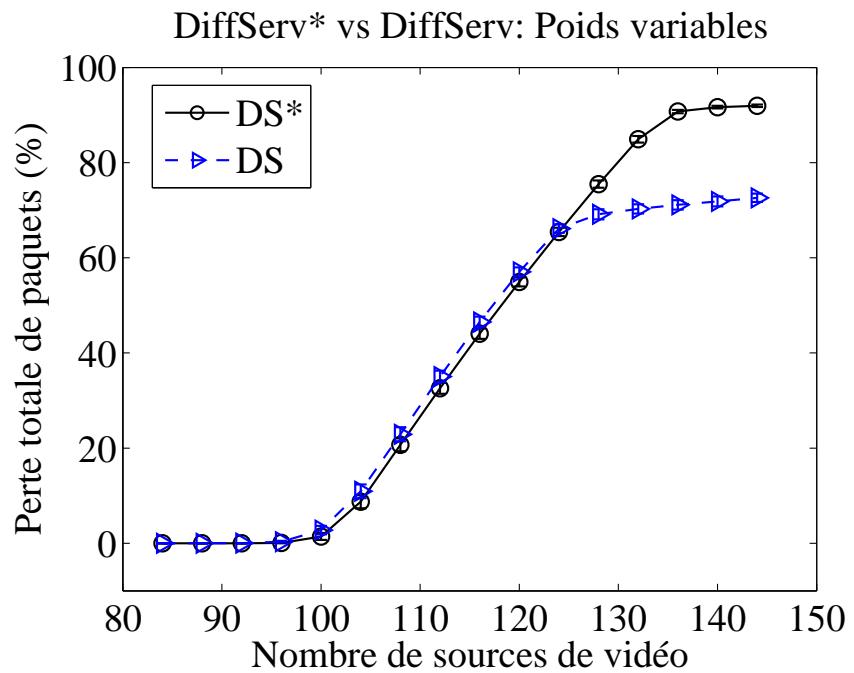


Figure 5.28 Perte du trafic vidéo dans un réseau avec poids variables : DiffServ vs DiffServ*

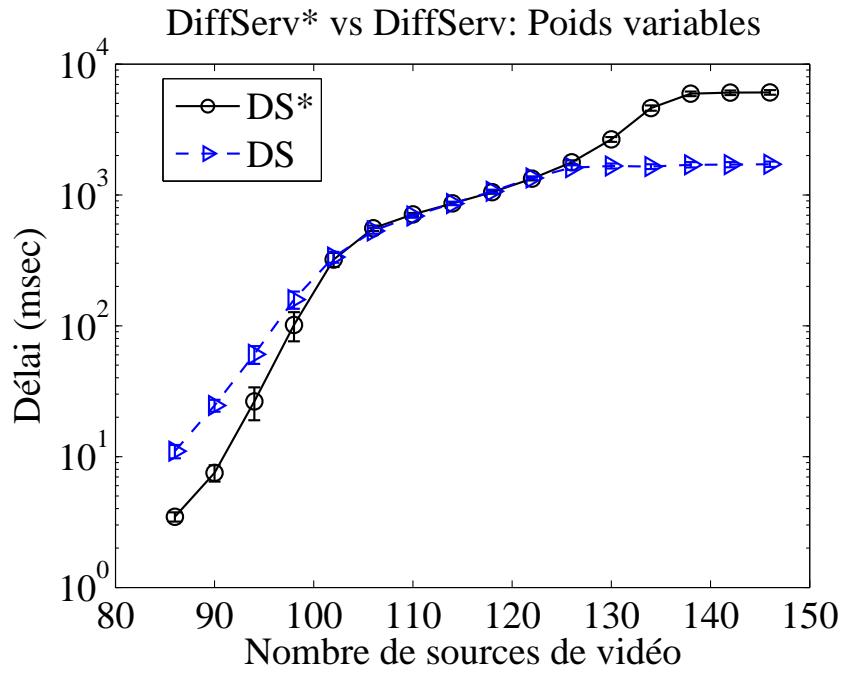


Figure 5.29 Délai moyen du trafic vidéo dans un réseau avec poids variables : DiffServ vs DiffServ*

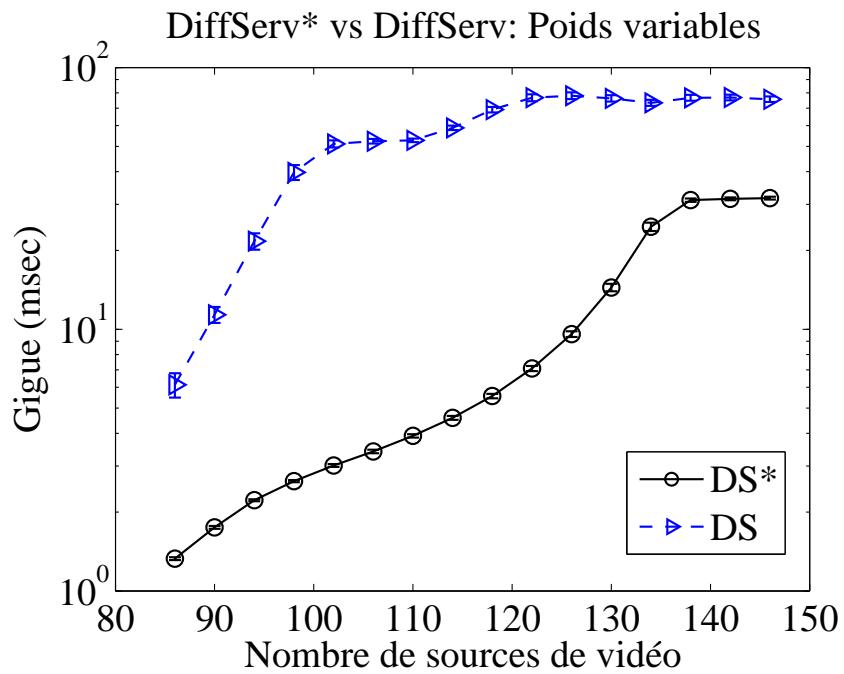


Figure 5.30 Gigue du trafic vidéo dans un réseau avec poids variables : DiffServ vs DiffServ*

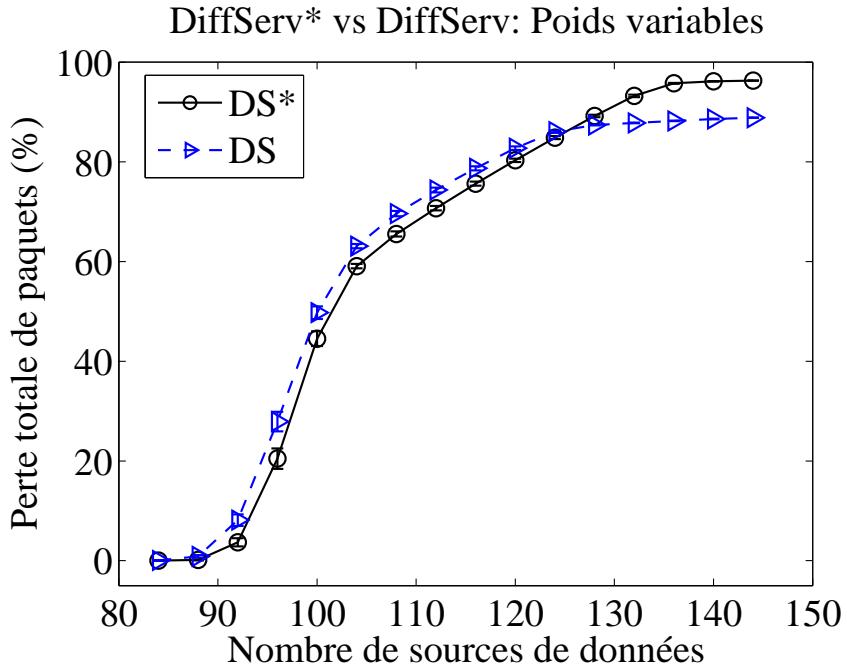


Figure 5.31 Perte du trafic données dans un réseau avec poids variables : DiffServ vs DiffServ*

observé pour le trafic VoIP. Ainsi, le prix à payer pour une amélioration des performances de la VoIP se résume à une dégradation plus importante des pertes des trafics vidéo et données.

L'évolution du délai du trafic données en fonction du nombre de sources est semblable à l'évolution du délai du trafic vidéo. Une faible différence de délai apparaît aussi sur la figure 5.32 en faveur de DiffServ* lorsque moins de 100 sources sont actives. Cette différence, comme nous l'avons expliqué, est liée au choix qu'opère DiffServ* pour sélectionner l'interface de sortie. Cependant, les deux modèles produisent un même délai lorsque le réseau commence à saturer. Une fois saturé, le comportement des deux mécanismes diffèrent et comme on peut le voir, DiffServ* protège d'autant moins les données qu'il privilégie la voix. Il s'agit donc d'un compromis à faire entre les performances désirées pour la voix et les performances des autres classes.

Par rapport à la gigue, la figure 5.33 montre que DiffServ* reste sans équivoque un meilleur régulateur de trafic que DiffServ.

L'étude de performance des ordonnanceurs DiffServ et DiffServ* tous deux utilisant des poids variables dans un réseau maillé a révélé les compromis qu'il y a lieu de faire afin de garantir la qualité de service du trafic VoIP dans un réseau bout à bout. En effet, les performances des trafics vidéo et données doivent être dégradées afin de rendre cela possible. De plus, ces deux trafics ont des performances quasi-similaires en ce qui à trait aux pertes

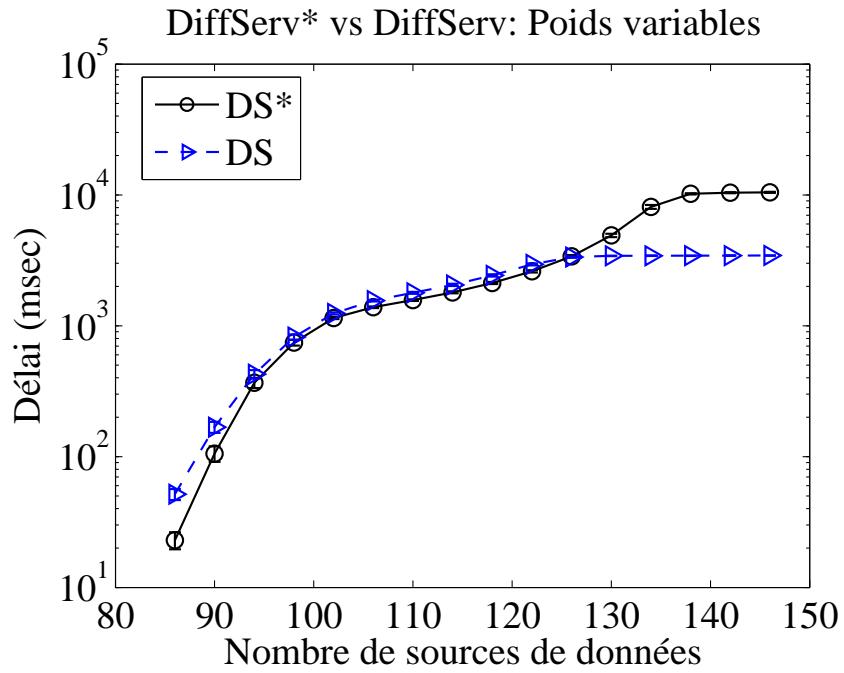


Figure 5.32 Délai moyen du trafic données dans un réseau avec poids variables : DiffServ vs DiffServ*

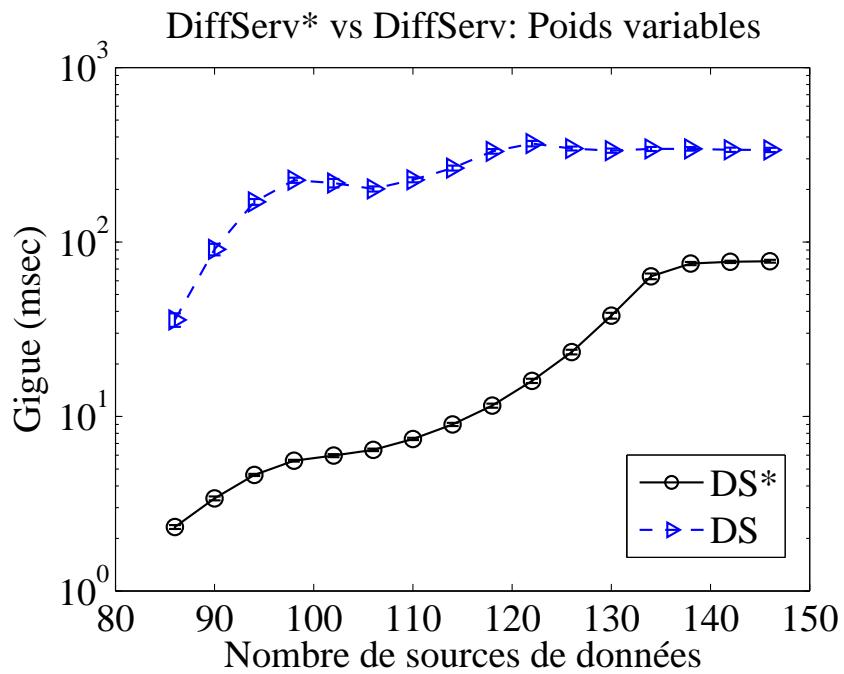


Figure 5.33 Gigue du trafic données dans un réseau avec poids variables : DiffServ vs DiffServ*

et au délai tant que le réseau n'est pas trop saturé. Le comportement en saturation diffère suivant l'utilisation de DiffServ ou DiffServ* compte tenu de la méthode d'implémentation du procédé de sélection des poids variables. Pour ce qui est de la gigue, DiffServ* reste et demeure la solution qui minimise le mieux cette métrique, quel que soit le type ou le volume de trafic. Par ailleurs, les résultats montrent aussi que les différences d'implémentation font du modèle DiffServ* un meilleur protecteur du trafic VoIP que le modèle DiffServ.

5.4.2 Comparaison poids statiques et poids variables

Nous évaluons, ici, les performances des trois classes dans un contexte où les poids sont variables et nous les comparons aux performances à poids fixes en utilisant le modèle *DiffServ**. En mode statique, les poids utilisés sont encore $\alpha_{EF} = 20$, $\alpha_{AF} = 3$, $\alpha_{BE} = 1$.

5.4.2.1 Classe EF – Trafic VoIP

Nous comparons l'effet des poids sur les performances du trafic VoIP. La figure 5.34 montre l'augmentation des pertes en fonction du volume de trafic envoyé. Les résultats sont semblables aux performances sur un lien puisque tous les liens ont les mêmes propriétés. Les poids variables présentent une diminution des pertes du trafic VoIP de 9% par rapport à la pondération statique.

Comme pour le cas d'un lien, les poids statiques permettent de mesurer le délai de saturation du trafic VoIP tandis qu'avec les poids variables le réseau n'est pas encore saturé avec 106 sources de trafic. C'est ce que montre la figure 5.35. Ainsi, la variation des poids déplace le point de saturation du réseau mais permet de desservir davantage de trafic VoIP au détriment des autres trafics.

Par rapport à la gigue, la figure 5.36 montre que les poids variables procurent, encore une fois, une meilleure protection que les poids fixes. La diminution de la gigue est liée à l'augmentation du poids assigné au trafic VoIP, lui permettant ainsi d'accéder au canal de façon plus fréquente. Comme on pouvait s'y attendre suite aux résultats sur un lien, les poids variables offrent bien une meilleure protection au trafic VoIP que les poids fixes.

5.4.2.2 Classe AF – Trafic Vidéo

La figure 5.37 montre que le trafic vidéo est également mieux protégé avec la pondération variable, pour autant que le réseau n'est pas saturé. Dans un réseau moyennement chargé, moins de 105 sources par exemple, les poids variables permettent une diminution des pertes allant jusqu'à 4% par rapport aux poids statiques. Cependant, les pertes causées par les poids

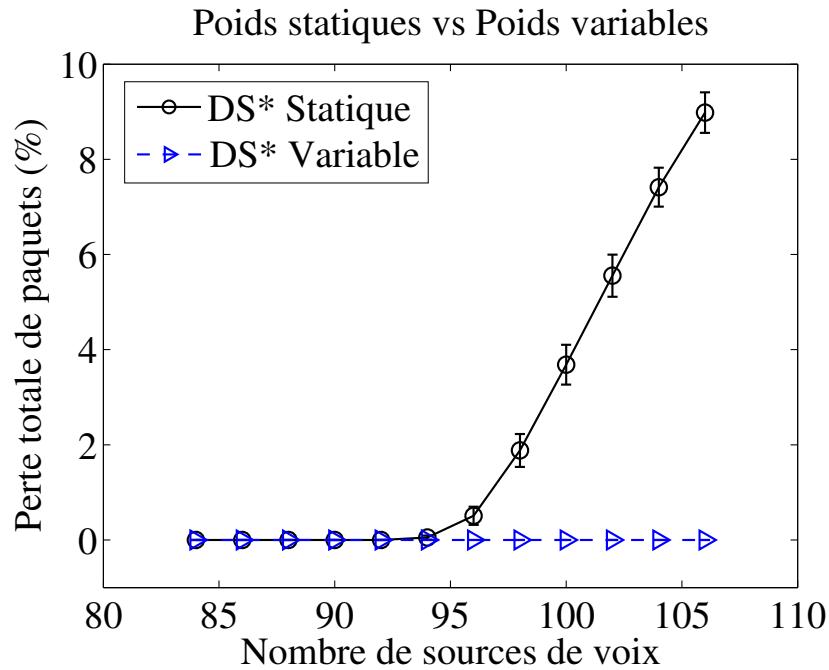


Figure 5.34 Perte du trafic VoIP dans un réseau : Poids fixes 20-3-1 vs Poids variables

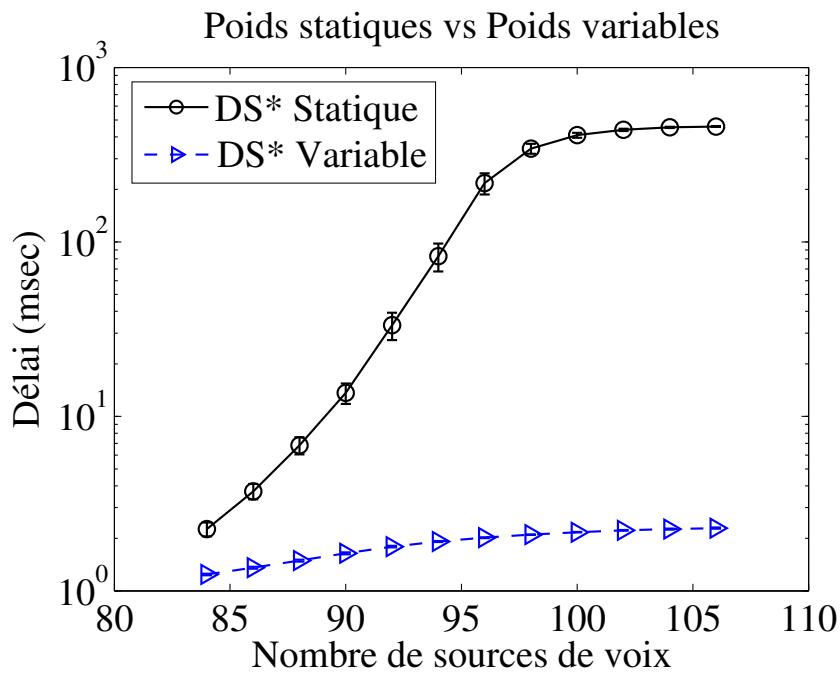


Figure 5.35 Délai moyen du trafic VoIP dans un réseau : Poids fixes 20-3-1 vs Poids variables

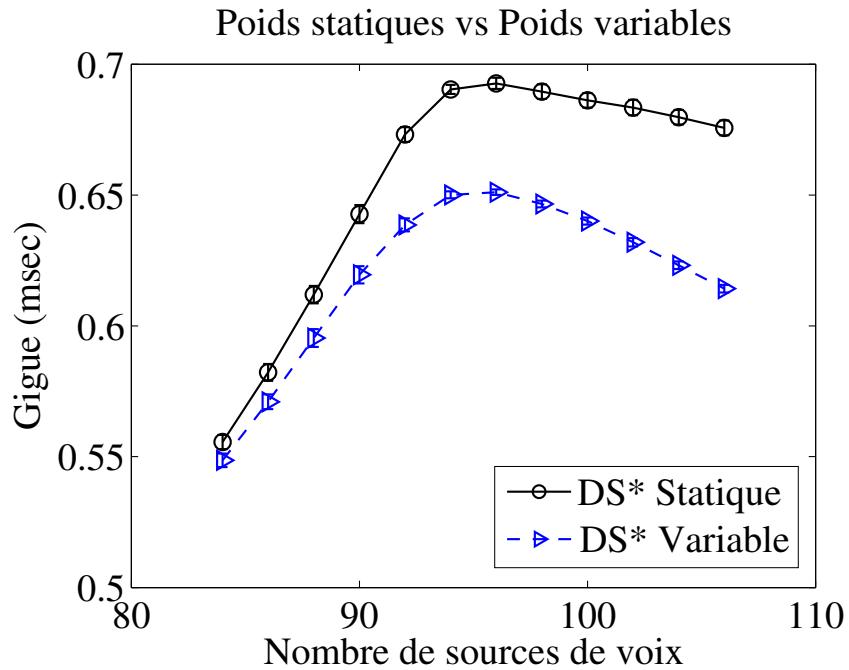


Figure 5.36 Gigue du trafic VoIP dans un réseau : Poids fixes 20-3-1 vs Poids variables

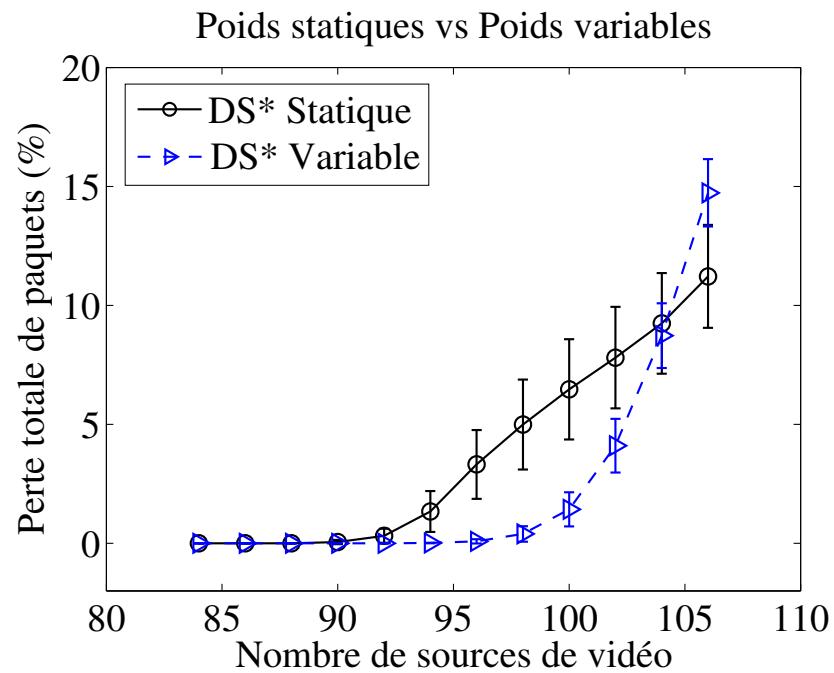


Figure 5.37 Perte du trafic vidéo dans un réseau : Poids fixes 20-3-1 vs Poids variables

variables deviennent plus importantes dès que le réseau commence à saturer. C'est donc une question de compromis entre les pertes du trafic VoIP et celles de la vidéo.

De même que les pertes, le délai de ce trafic reste inférieur au délai mesuré avec les poids fixes tant que le réseau n'est pas saturé (Figure 5.38). Par ailleurs, puisque le délai de saturation n'est pas encore atteint avec les poids variables, il est logique de penser que l'amélioration observée sur les performances de la VoIP et de la vidéo engendre une dégradation de celles des données. Cette hypothèse sera vérifiée dans la section 5.4.2.3.

Contrairement au délai et aux pertes, la figure 5.39 montre que la gigue du trafic vidéo augmente faiblement par rapport à la pondération fixe. Comme nous l'avons expliqué pour un lien, cela est lié à l'augmentation des délais de transfert de ce trafic, augmentation due au fait que le trafic VoIP est servi beaucoup plus longtemps avec les poids variables. Par rapport à cette métrique, les poids variables procurent une moins bonne protection à la vidéo. Cependant, de façon globale, il ressort de toutes ces observations que les poids variables protègent également mieux le trafic vidéo tant que le réseau n'est pas saturé.

5.4.2.3 Classe BE – Trafic Données

Nous comparons l'effet des poids sur les performances du trafic données. La figure 5.40 montre l'augmentation des pertes en fonction du volume de trafic données. Comme on pouvait s'y attendre, les pertes pour ce trafic sont importantes avec les poids variables. En effet, le prix à payer pour améliorer les performances des classes EF et AF consiste à pénaliser la classe BE en réduisant son accès au canal. C'est effectivement ce que nous faisons avec la pondération variable.

De même que les pertes, le délai du trafic données augmente quand on utilise les poids variables et la figure 5.41 montre que le délai de saturation est de l'ordre de 1400 ms contre 650 ms pour les poids fixes. Ce qui n'est pas étonnant étant donné que les performances des trafics VoIP et vidéo ont été améliorées.

La figure 5.42 montre l'évolution de la gigue en fonction du volume de trafic données. Évidemment, cette métrique aussi augmente par rapport aux mesures à poids fixes étant donné que le trafic données accède beaucoup moins souvent au canal. En général, les performances de la classe BE se dégradent au profit des deux autres classes. En conséquence, les poids variables permettent une meilleure différenciation de service par rapport aux poids fixes $\alpha_{EF} = 20$, $\alpha_{AF} = 3$ et $\alpha_{BE} = 1$. Cependant la qualité de service du trafic données se dégrade considérablement.

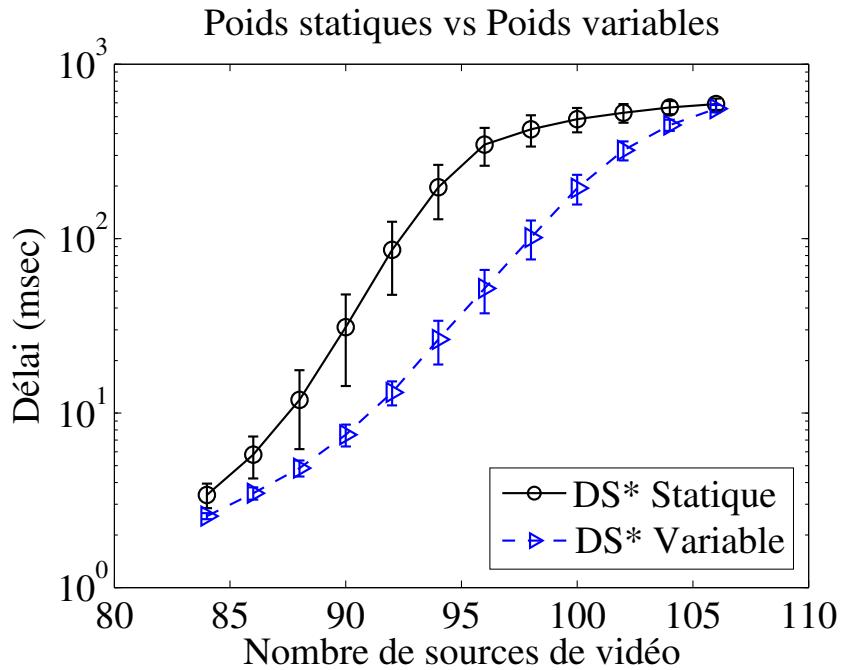


Figure 5.38 Délai moyen du trafic vidéo dans un réseau : Poids fixes 20-3-1 vs Poids variables

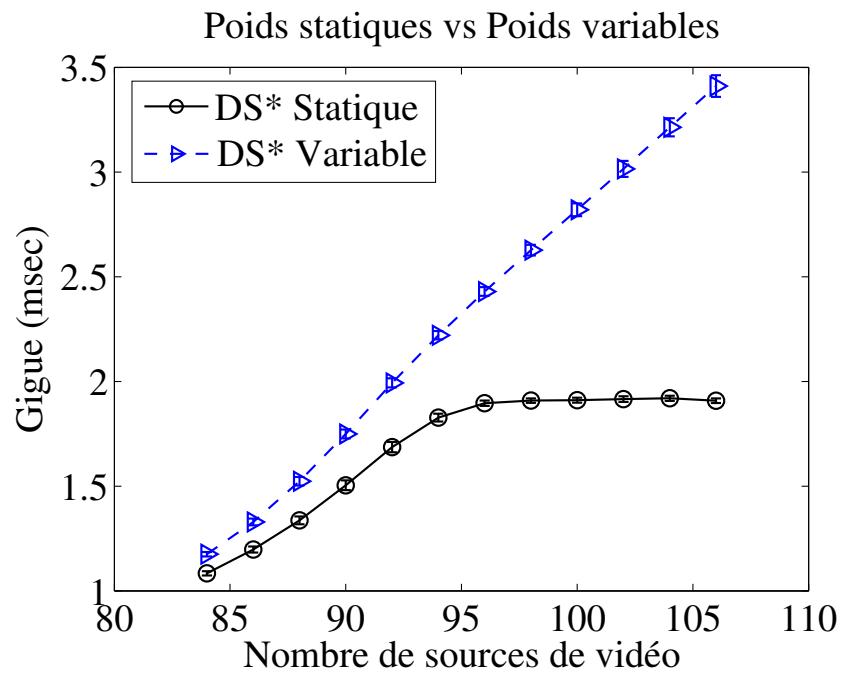


Figure 5.39 Gigue du trafic vidéo dans un réseau : Poids fixes 20-3-1 vs Poids variables

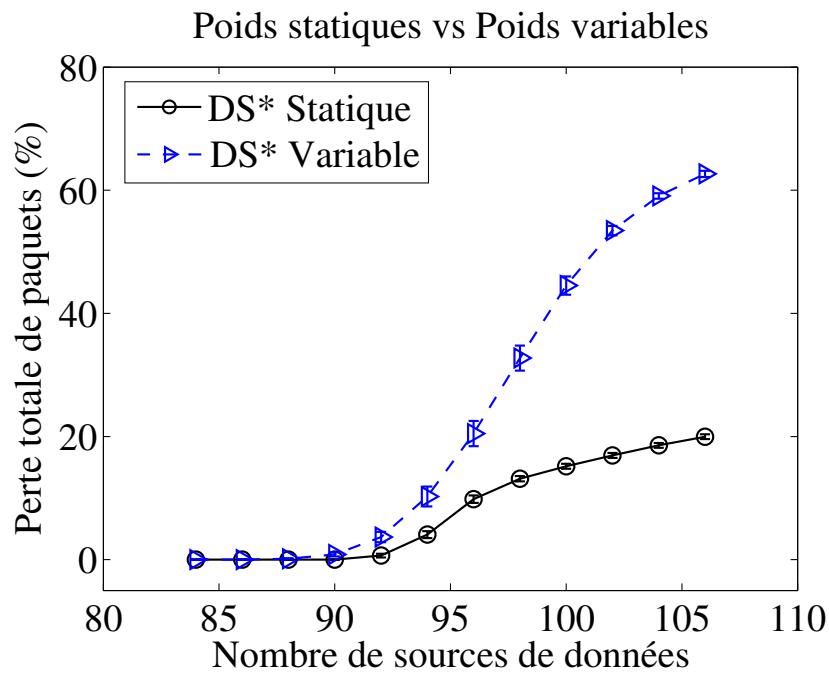


Figure 5.40 Perte du trafic données dans un réseau : Poids fixes 20-3-1 vs Poids variables

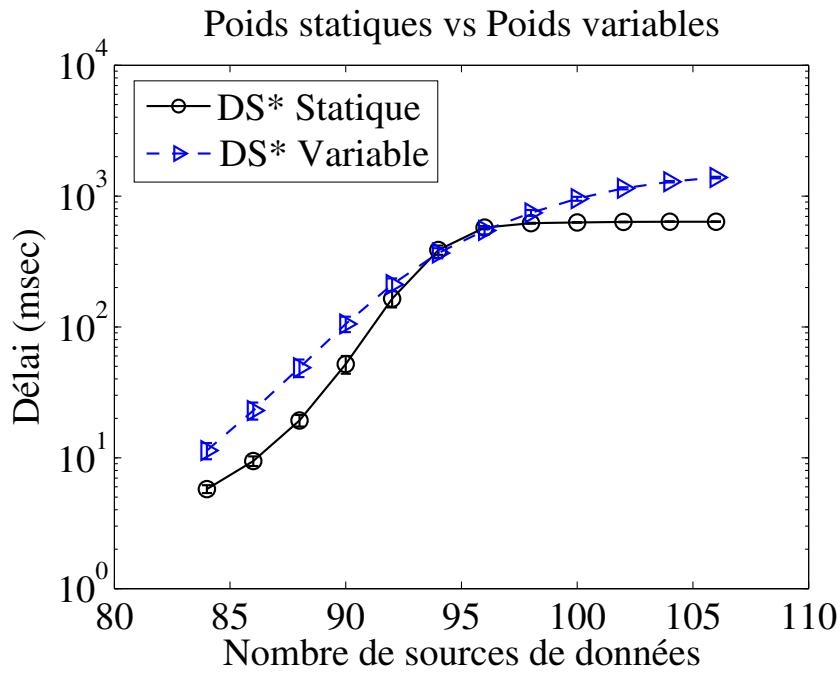


Figure 5.41 Délai moyen du trafic données dans un réseau : Poids fixes 20-3-1 vs Poids variables

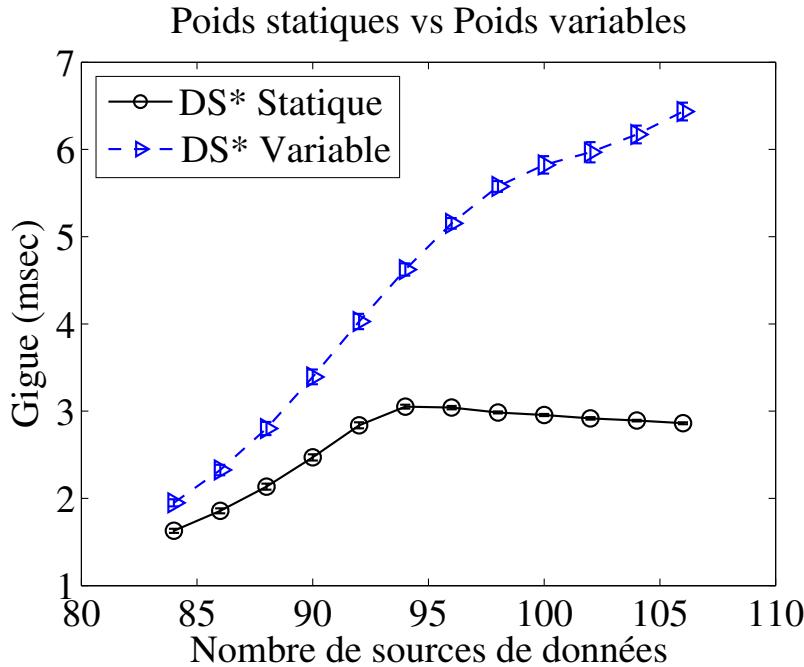


Figure 5.42 Gigue du trafic données dans un réseau : Poids fixes 20-3-1 vs Poids variables

5.4.3 Utilisation dynamique de la bande passante

Afin de s'assurer que la répartition de bande passante a également lieu *lien-par-lien* dans un réseau bout-à-bout, nous mesurons la bande passante utilisée par chaque classe lorsque le SNR moyen de chaque lien physique varie de 5 dB à 25 dB. C'est l'ordonnanceur DiffServ* qui est utilisé dans cette étude. Tous les liens ont les mêmes propriétés. Pour un SNR moyen donné, les canaux sont variables et réestimés à la fréquence F_w soit toutes les 3,581 ms. Le PER moyen est fixé à 10^{-4} et le réseau est saturé par 200 sources de trafic par classe soit une charge moyenne de 6,4 Mb/s par classe. Avec une telle charge, le réseau est fortement saturé pour de faibles SNR car cela permet de s'assurer qu'il reste saturé même lorsque la qualité du canal s'améliore. La figure 5.43 montre l'évolution de la bande passante en fonction du SNR moyen. Les résultats sont similaires à ceux d'un lien logique (voir Figure 5.23). On constate que la bande passante allouée à la voix décroît progressivement lorsque le SNR augmente et inversement pour la vidéo et les données. En effet, à faible SNR, par exemple 5 dB, tous les canaux sont en moyenne "mauvais" car il est très difficile d'avoir des taux de transmission de 54 Mb/s avec ce SNR moyen. Ainsi, l'ordonnanceur utilise la plupart du temps les poids $\alpha_{EF} = 163$, $\alpha_{AF} = 6$, $\alpha_{BE} = 1$, qui permettent à la voix de recevoir près de 85% de la capacité disponible, 10% à la vidéo et 5% pour les données. De même, lorsque la qualité du canal augmente, par exemple 25 dB, il est beaucoup plus probable d'avoir des taux de

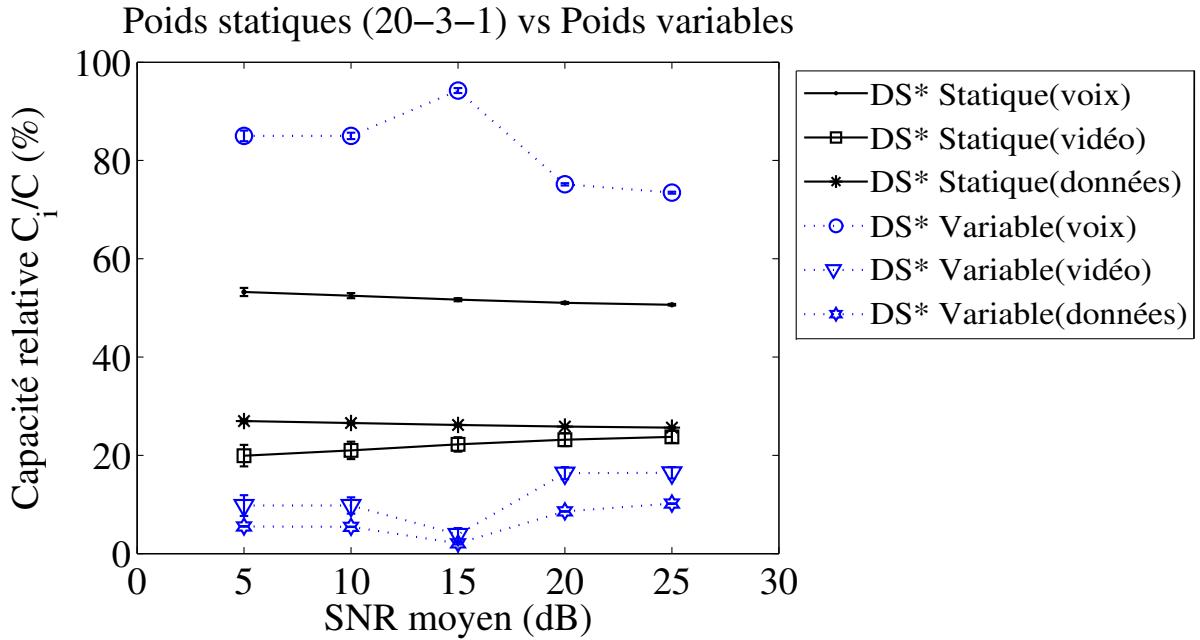


Figure 5.43 Pondération variable vs statique 20-3-1 : capacité relative dans un réseau

transmission supérieurs à 18 Mb/s donc la qualité du canal varie au cours du temps ce qui permet à l'ordonnanceur d'utiliser différents poids. En particulier à 25 dB, on remarque que la voix reçoit environ 73% de la bande passante, la vidéo 17% et les données 10%. Cette répartition tend vers celle d'un canal de "bonne qualité". Cependant, avec la pondération fixe $\alpha_{EF} = 20$, $\alpha_{AF} = 3$, $\alpha_{BE} = 1$, la répartition de la bande passante reste constante quelle que soit la qualité du canal et on mesure que le trafic VoIP occupe environ 52% de bande passante, la vidéo 22% et les données 26%. Quoique la voix reste mieux protégée, la vidéo est par contre défavorisée et au lieu de gagner 37% de la bande passante, elle n'en reçoit que 22% soit 4% de moins que les données. Cela est dû au fait que, dans ce contexte où le réseau est fortement saturé, la pondération minimale protège moins bien la vidéo notamment parce que les paquets de ce trafic sont de taille variable. Il est donc préférable de choisir des poids supérieurs à la pondération minimale pour être à l'abri d'une telle situation. Pour cela, nous recommandons d'intégrer au modèle DiffServ* une méthode de garantie de bande passante lorsqu'on utilise l'ordonnancement WRR. Cela permet d'assurer une meilleure différenciation de service même quand le réseau est fortement saturé.

5.5 Sommaire

Dans ce chapitre, nous avons étudié l'impact de la variation des poids sur les performances des réseaux sans fil. Pour cela, un regroupement des états de l'AMC a permis de définir

différents niveaux de qualité dans un canal et une répartition de la bande passante pour chaque niveau. Cette répartition est ensuite utilisée pour déterminer les poids du WRR. Étant donné que la qualité du canal évolue dans le temps, les poids du WRR varient également en fonction de la qualité du canal. Les résultats obtenus en comparant les performances avec des poids variables et les performances avec des poids constants ont montré qu'en augmentant les poids assignés aux trafics VoIP et vidéo, leurs performances sont améliorées au détriment du trafic données. Par ailleurs, l'utilisation d'un ordonnancement WRR nécessite une allocation de bande passante pour chaque classe de trafic afin de maintenir une bonne différenciation entre les classes lorsque le réseau est fortement saturé. Bien que cette étude soit basée sur des paquets de longueurs différentes, l'expression (5.4) qui permet de déterminer les poids de l'ordonnanceur WRR en fonction des proportions de bande passante reste valide même si les paquets sont tous de même taille. Ainsi les conclusions de cette étude restent aussi valides. D'autre part, l'implémentation des poids variables dans les modèles DiffServ et DiffServ* et l'étude de performances des deux modèles a abouti aux mêmes conclusions que celles du chapitre 4 notamment en ce qui concerne les performances sur un lien logique et la forte diminution de la gigue liée à l'utilisation d'un seul ordonnanceur au lieu de trois. Toutefois, les performances dans un réseau ont révélé que le modèle DiffServ* protège mieux le trafic VoIP que le modèle DiffServ. Enfin, les avantages de la variation des poids s'observent quand le SNR moyen est variable. Ceci est le cas des réseaux maillés sans fil étant donné que les effets des évanouissements combinés aux interférences radio varient le SNR moyen d'un lien à l'autre. Une telle situation nécessite alors qu'à chaque noeud, la politique d'ordonnancement des paquets soit dynamique pour assurer en toute situation la qualité de service du trafic VoIP. Lorsque la qualité du canal est bonne ou très bonne, il y a assez de marge pour écouler tous les trafics. Cependant lorsque la qualité se dégrade, la variation dynamique des poids pénalise l'accès des classes AF et BE au canal afin de garantir l'écoulement du trafic EF.

CONCLUSION

Ce mémoire a été consacré à l'analyse des performances de *Wireless-DiffServ**, modèle de protection différenciée dans les réseaux maillés 802.11. Ces réseaux gagnent en popularité à cause de leur facilité d'installation, de leurs faibles coûts et leur utilisation principale qui vise à fournir l'Internet sans fil à haut débit sur des étendues géographiques très vastes. Ces réseaux sont donc destinés à acheminer différents services IP, notamment la voix, la vidéo et les données, ayant des besoins en qualité de service différents. De plus, les exigences de ces différentes applications doivent être satisfaites en dépit de la nature variable du canal sans fil. Ainsi, le modèle *Wireless-DiffServ** offre la possibilité d'adapter l'ordonnancement des paquets des différentes classes de service en fonction de la qualité du médium sans fil. Cette caractéristique permet de protéger les classes de trafic de haute priorité contre les effets des évanouissements du canal sans fil et de satisfaire en tout temps leurs exigences en qualité de service.

Dans un premier temps, une description des réseaux maillés et du standard IEEE 802.11 a été présentée afin de faire ressortir les problèmes majeurs liés au déploiement de cette architecture dans les réseaux de type 802.11. En effet, d'une part les interférences et la nature variable du canal sans fil dégradent les performances des réseaux sans fil mais d'autre part, le mécanisme d'accès au médium de la couche MAC ne garantit aucune priorité d'accès pour les applications en temps réel et engendre de surcroît des délais d'attente supplémentaires par paquet. Ces délais nuisent gravement aux performances du réseau car ils diminuent considérablement la capacité du réseau en ce qui concerne le nombre de sessions de voix admissibles. Bien que les causes soient multiples, le problème crucial dans les réseaux maillés 802.11 est donc celui de la garantie de la qualité de service.

Basé sur l'utilisation conjointe de DiffServ* et d'un procédé d'adaptation de modulation et de codage, un modèle cross-layer, nommé *Wireless-DiffServ**, a été conçu dans ce projet pour analyser les effets de l'interaction entre les couches physique, MAC et logique sur la qualité de service des applications de voix, vidéo et données. DiffServ* est un mécanisme de protection différenciée capable d'offrir de la qualité de service en cas de congestion ou de panne physique dans les réseaux IP de fibres optiques. Dans ce projet, *Wireless-DiffServ** est utilisé pour donner de la qualité de service en cas d'évanouissement du canal sans fil. À cet effet, un procédé d'adaptation de la modulation et du codage est mis en place dans un canal sans fil de Rayleigh modélisé par une chaîne de Markov discrète à états finis. Chaque

état de la chaîne représente un état d'évanouissement et correspond à un intervalle de valeurs du rapport signal sur bruit (SNR) caractérisé par un taux de transmission particulier. Dans chaque intervalle de SNR, la probabilité d'erreur par paquet est constante et égale à une valeur cible. Les bornes des intervalles de SNR permettent de calculer la distribution de probabilité stationnaire utilisée pour connaître l'état du canal à un instant donné au cours des simulations. La simulation du modèle *Wireless-DiffServ** et du modèle DiffServ standard, montre que l'interaction entre trois couches (IP, MAC et PHY) conduit à un réseau de meilleures performances qu'une interaction entre seulement les couches MAC et PHY. En effet, les résultats de simulation montrent que le modèle *Wireless-DiffServ** diminue considérablement la gigue moyenne du réseau d'une part et à faible et moyenne charge, diminue aussi le délai moyen des paquets. Il est donc nécessaire de limiter la charge maximale du réseau afin d'assurer à *Wireless-DiffServ** un fonctionnement optimal.

Par la suite, un regroupement des états d'évanouissement a permis de définir trois niveaux de qualité : mauvais, bon et très bon. Pour chaque niveau de qualité, la largeur de bande disponible est répartie différemment entre les classes de service de façon à toujours garantir au moins 50% de la capacité au trafic voix. En cas d'évanouissement profond, le trafic de voix occupe à lui seul au moins 80% de la bande passante disponible. Les résultats de simulation montrent que la variation des poids en fonction des conditions du canal permet effectivement de protéger le trafic VoIP des évanouissements du canal sans fil mais cette protection se fait au coût d'une dégradation des performances des autres classes de trafic. Les avantages de la variation des poids se mesurent en particulier dans un environnement où le SNR moyen est variable. Ceci est le cas des réseaux maillés sans fil étant donné que les effets des évanouissements combinés aux interférences radio varient le SNR moyen d'un lien à l'autre. Une telle situation nécessite alors que la politique d'ordonnancement des paquets soit dynamique pour assurer en toute situation la qualité de service du trafic VoIP.

Par ailleurs, l'étude de l'impact du SNR moyen et du taux d'erreur cible montre que la performance des réseaux (taux de perte, délai de transmission, gigue) est meilleure pour des valeurs de SNR élevées et qu'une probabilité d'erreur par paquet de l'ordre de 10^{-2} semble être un compromis idéal entre les performances du réseau et le taux d'erreur désiré.

Dans ce projet, *Wireless-DiffServ** ne fait pas de contrôle de congestion mais uniquement un contrôle de la qualité du canal étant donné que dans les réseaux sans fil les pertes d'information sont principalement dues à des problèmes de transmission du signal. De plus, le modèle étudié ne prend pas en compte les phénomènes d'interférence radio mais uniquement

les évanouissements du canal sans fil. Par ailleurs, l'utilisation des probabilités stationnaires pour déterminer l'état du médium à un instant donné ne permet pas d'étudier, par simulation, l'impact de la fréquence Doppler sur les performances du réseau. Bien qu'un mécanisme de mise à jour périodique des états ait été introduit pour simuler la vitesse de variation du canal, la fréquence de ce mécanisme est indépendante des paramètres physiques du canal (SNR et PER) donc une étude d'impact de la fréquence de variation des canaux ne réflecte pas l'incidence réelle de la fréquence Doppler sur les performances du réseau. D'autre part, en utilisant l'ordonnanceur WRR, cette étude s'appuie sur le fait que la taille moyenne des paquets est préalablement connue. Cependant, dans les réseaux IP la taille moyenne des paquets est une quantité qui pourrait ne pas être connue d'avance. En outre, le modèle utilisé pour les trafics vidéo et données n'est pas très réaliste car la vidéo produit généralement une charge plus importante que celle des deux autres trafics et les données produisent moins de charge.

Améliorations futures

Les possibilités de développement du modèle *Wireless-DiffServ** sont nombreuses. La première amélioration possible serait de définir autant de niveaux de qualité qu'il y a d'états dans la chaîne de Markov afin qu'il y ait plus de niveaux intermédiaires entre un mauvais canal et un très bon canal pour que le passage d'un niveau à l'autre soit moins brusque. Une autre amélioration consisterait à utiliser les probabilités de transition afin d'étudier l'impact de la fréquence Doppler sur l'adaptation des poids.

D'autre part, les performances du réseau pourraient être réévaluées avec d'autres caractéristiques de trafic pour la vidéo et les données. Les codecs MPEG-4 ou H-264 pourraient être utilisés pour la vidéo tandis que les données pourraient être générées à débit constant pendant toute la durée de la simulation (Constant Bit Rate). Par ailleurs, en ce qui concerne les métriques de qualité de service, le taux de perte pourrait prendre également en compte les paquets de voix ayant un délai de transmission supérieur à 70 msec et les paquets de vidéo ayant un délai supérieur à 100 msec. Ces limitations permettraient d'étudier de façon plus précise l'impact du modèle *Wireless-DiffServ** sur la gigue des paquets.

Une ouverture possible de ce projet serait de remplacer l'ordonnanceur WRR par le *Deficit WRR (DWRR)* qui ne nécessite pas de connaître la taille moyenne des paquets mais opère sur la base du nombre de bits servis.

Une autre perspective d'ouverture serait de rendre l'implémentation actuelle du modèle plus flexible en considérant l'utilisation de différents modèles de canal d'une interface à l'autre. Ainsi, le canal de Rayleigh pourrait être remplacé par le modèle de Rice qui prend en considération l'existence de la composante de ligne directe.

Une autre possibilité serait d'intégrer un algorithme d'attribution dynamique de fréquence afin de limiter les interférences radio et maximiser la capacité du réseau. En outre, l'adaptation de la modulation et du codage pourrait être basée sur le rapport signal sur interférence et bruit (SINR) et non pas seulement sur le SNR.

Enfin, une étude analytique du modèle permettrait de voir s'il y a une façon plus optimale d'implémenter les poids variables en fonction des conditions du canal sans fil et comment choisir la répartition de la bande passante par niveau.

RÉFÉRENCES

- [1] (2012). The network simulator–ns–2. <http://www.isi.edu/nsnam/ns/>.
- [2] AKYILDIZ, I. F. et WANG, X. (2009). *Wireless Mesh Networks*. Advanced Texts in Communications and Networking. Wiley.
- [3] AWAD, C., SANSÒ, B. et GIRARD, A. (2009). Differentiated reliability in traffic engineered MPLS and DiffServ-aware next generation networks. In *Proc. 7th International Workshop on Design of Reliable Communication Networks (DRCN)*, IEEE, pages 265–272.
- [4] BARRY, M., CAMPBELL, A. et VERES, A. (2001). Distributed control algorithms for service differentiation in wireless packet networks. In *Proc. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 1, pages 582–590.
- [5] BAUMANN, R., HEIMLICHER, S., LENDERS, V. et MAY, M. (2007). Routing packets into wireless mesh networks. In *Proc. Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 38–45.
- [6] BONONI, L., DI FELICE, M., MOLINARO, A. et PIZZI, S. (2009). A cross-layer architecture for service differentiation in multi-channel multi-radio wireless mesh networks. In *Proc. 6th international Symposium on Wireless Communication Systems (ISWCS)*, IEEE Press, Piscataway, NJ, USA, pages 171–175.
- [7] BROSH, E., BASET, S. A., MISRA, V., RUBENSTEIN, D. et SCHULZRINNE, H. (2010). The delay-friendliness of TCP for real-time traffic. *IEEE/ACM Trans. Netw.*, 18(5), 1478–1491.
- [8] CALVO, R. A. et CAMPO, J. P. (2007). Adding multiple interface support in ns-2. <http://personales.unican.es/aguero/cr/files/ucMultiIfacesSupport.pdf>.
- [9] CHEN, J. et YANG, Y. (2006). The impact of doppler spreading on delay performance over multi-hop wireless communications. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.148.4570>.
- [10] CHENG, H. T., JIANG, H. et ZHUANG, W. (2006). Distributed medium access control for wireless mesh networks. *Wireless Communications and Mobile Computing*, 6(6), 845–864.
- [11] DRAVES, R., PADHYE, J. et ZILL, B. (2004). Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. 10th annual international conference on Mobile computing and networking (MobiCom)*, ACM, pages 114–128.

- [12] EGELAND, G. et ENGELSTAD, P. (2008). The reliability and availability of wireless backhaul mesh networks. In *Proc. IEEE International Symposium on Wireless Communication Systems (ISWCS)*, pages 178–183.
- [13] FALL, K. et VARADHAN, K. (2010). The *ns* Manual (formerly *ns* notes and documentation). Rapport technique, UC Berkeley, LBL, USC/ISI, and Xerox PARC. <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [14] GHAZISAIDI, N., KASSAEI, H. et BOHLOOLI, M. (2009). Integration of WiFi and WiMAX-mesh networks. In *Proc. Second International Conference on Advances in Mesh Networks*, pages 1–6.
- [15] GNASSOU, A. (2008). *Impact de la couche physique sur la performance des applications IP dans les réseaux sans fil locaux IEEE 802.11*. Mémoire de maîtrise, École Polytechnique de Montréal.
- [16] GORE, A. D. et KARANDIKAR, A. (2011). Link scheduling algorithms for wireless mesh networks. *IEEE Communications Surveys and Tutorials*, 13(2), 258–273.
- [17] GUPTA, B., ACHARYA, B. et MISHRA, M. (2009). Optimization of routing algorithm in wireless mesh networks. In *Proc. World Congress on Nature Biologically Inspired Computing (NaBIC)*, pages 1150–1155.
- [18] HOENE, C., GUNTHER, A. et WOLISZ, A. (2003). Measuring the impact of slow user motion on packet loss and delay over IEEE 802.11b wireless links. In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)*, pages 652–662.
- [19] HOU, Y. et LEUNG, K. (2007). A novel distributed scheduling algorithm for wireless mesh networks. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM'07)*, pages 5053–5058.
- [20] IEEE (2003). IEEE standard for information technology—telecommunications and information exchange between systems—Local and Metropolitan Area Networks—specific requirements part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—amendment 4 : Further higher data rate extension in the 2.4 ghz band. *IEEE Std 802.11g-2003*, 1–67.
- [21] IEEE (2011). IEEE standard for information technology—telecommunications and information exchange between systems—Local and Metropolitan Area Networks—specific requirements part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications amendment 10 : Mesh networking. *IEEE Std 802.11s-2011*, 1–372.
- [22] IEEE (2012). IEEE standard for information technology—telecommunications and information exchange between systems Local and Metropolitan Area Networks—specific re-

- quirements part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *IEEE Std 802.11-2012*, 1–2793. Revision of IEEE Std 802.11-2007.
- [23] JIANG, H., ZHUANG, W., SHEN, X., ABDRABOU, A. et WANG, P. (2006). Differentiated services for wireless mesh backbone. *IEEE Communications Magazine*, 44(7), 113–119.
 - [24] JUN, J. et SICHITIU, M. (2003). The nominal capacity of wireless mesh networks. *IEEE Wireless Communications*, 10(5), 8–14.
 - [25] KAZANTZIDIS, M., GERLA, M. et LEE, S.-J. (2001). Permissible throughput network feedback for adaptive multimedia in AODV MANETs. In *Proc. IEEE International Conference on Communications*, volume 5, pages 1352–1356.
 - [26] KRISHNAMURTHY, L., CONNER, S., YARVIS, M., CHHABRA, J., ELLISON, C., BRABENAC, C. et TSUI, E. (2002). Meeting the demands of the digital home with high-speed multi-hop wireless networks. *Intel Technology Journal*, 6(4), 57–68.
 - [27] NASSEREDDINE, B., MAACH, A. et BENNANI, S. (2009). The scalability of the hybrid protocol in wireless mesh network 802.11s. In *Proc. Mediterranean Microwave Symposium (MMS)*, pages 1–7.
 - [28] NGUYEN, L., EYERS, T. et CHICHARO, J. (2000). Differentiated service performance analysis. In *Proceedings Fifth IEEE Symposium on Computers and Communications*, pages 328–333.
 - [29] NGUYEN, T., VERHULST, D. et MONTMORENCY, G. (2006). Marchés du WiFi en France et potentiel des réseaux maillés. Rapport technique, Autorité de Régulation des Communications Électroniques et des Postes (ARCEP).
 - [30] NICULESCU, D., GANGULY, S., KIM, K. et IZMAILOV, R. (2006). Performance of VoIP in a 802.11 Wireless Mesh Network. In *Proceedings of the 25th IEEE INFOCOM*, pages 1–11.
 - [31] ONG, L. et ALIAS, M. (2007). Performance of voice over IP (VoIP) over a wireless LAN (WLAN) for different audio/voice codecs. *Jurnal Teknologi*, 47(D), 39–60.
 - [32] PATHAK, H. et DUTTA, R. (2010). A survey of network design problems and joint design approaches in wireless mesh networks. *IEEE Communications Surveys & Tutorials*, 13(3), 396–428.
 - [33] QIAO, D. et CHOI, S. (2005). Fast-responsive link adaptation for IEEE 802.11 WLANs. In *Proc. IEEE International Conference on Communications*, volume 5, pages 3583–3588.
 - [34] QIAO, D., CHOI, S. et SHIN, K. (2002). Goodput analysis and link adaptation for IEEE 802.11a wireless LANs. *IEEE Transactions on Mobile Computing*, 1(4), 278–292.

- [35] RANIWALA, A. et CKER CHIUEH, T. (2005). Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *Proceedings of the 24th IEEE INFOCOM*, volume 3, pages 2223–2234.
- [36] RANIWALA, A., GOPALAN, K. et CHIUEH, T.-C. (2004). Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8, 50–65.
- [37] RAPPAPORT, T. (2001). *Wireless Communications : Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA.
- [38] RIGGIO, R., MIORANDI, D., DE PELLEGRINI, F., GRANELLI, F. et CHLAMTAC, I. (2008). A traffic aggregation and differentiation scheme for enhanced QoS in IEEE 802.11-based wireless mesh networks. *Comput. Commun.*, 31(7), 1290–1300.
- [39] SANSÒ, B., AWAD, C. et GIRARD, A. (2006). Can DiffServ guarantee IP QoS under failures ? *IEEE Network*, 20(4), 32–40.
- [40] SAYENKO, A., HÄMÄLÄINEN, T., JOUTSENSALO, J. et KANNISTO, L. (2006). Comparison and analysis of the revenue-based adaptive queuing models. *Comput. Netw.*, 50(8), 1040–1058.
- [41] SCHULZRINNE, H., CASNER, S., FREDERICK, R. et JACOBSON, V. (2003). RFC 3550 : RTP : A transport protocol for real-time applications. IETF.
- [42] SEMERIA, C. et STEWART III, J. (2001). Supporting differentiated service classes in large IP networks. White paper, Juniper Networks. <http://tinyurl.com/bsz9ltp>.
- [43] SHERIFF, I. et BELDING-ROYER, E. (2006). Multipath selection in multi-radio mesh networks. In *Proc. 3rd International Conference on Broadband Communications, Networks and Systems (BROADNETS)*, pages 1–11.
- [44] SRIKRISHNA, D. et CHARI, A. (2005). *Selection of routing paths based upon path quality of a wireless mesh network*. Tropos Networks, Sunnyvale, CA, US. United States Patent 6965575.
- [45] TARGON, V., SANSÒ, B. et CAPONE, A. (2010). The joint gateway placement and spatial reuse problem in wireless mesh networks. *Computer Networks*, 54(2), 231–240.
- [46] TSE, D. et VISWANATH, P. (2005). *Fundamentals of Wireless Communications*. Cambridge University Press.
- [47] WANG, W., LIEW, S. et LI, V. (2005). Solutions to performance problems in VoIP over a 802.11 wireless LAN. *IEEE Transactions on Vehicular Technology*, 54(1), 366–384.
- [48] WEI, Y., MAOSHENG, R., ZHAO, T. et XIAOMING, L. (2008). A bandwidth management scheme support for real-time applications in wireless mesh networks. In *Proc. ACM symposium on Applied computing*, ACM, pages 2063–2068.

- [49] ZHU, H., LI, M., CHLAMTAC, I. et PRABHAKARAN, B. (2004). A survey of quality of service in IEEE 802.11 networks. *IEEE Wireless Communications*, 11(4), 6–14.

ANNEXE A

DESCRIPTION DE L'IMPLÉMENTATION

NS (*Network Simulator*) est un logiciel de simulation des réseaux de télécommunications. Il est disponible gratuitement sur Internet [1] et très utilisé par la communauté scientifique à cause de sa modularité. En effet, *ns* a une architecture hiérarchique ce qui facilite son extension ou l'ajout de nouveaux modules. Son noyau est écrit en C++ mais il dispose d'une interface de programmation en OTcl (*Object-oriented Tcl*). Le langage C++, orienté objet, est utilisé pour implémenter les algorithmes et décrire le fonctionnement des différents protocoles car il permet de manipuler rapidement et efficacement les composants créés au cours d'une simulation. Le langage OTcl permet de modifier facilement les paramètres de configuration de certains éléments implementés en C++ sans avoir besoin de recompiler l'exécutable *ns*. Ce langage permet donc de lier des objets OTcl à leurs homologues C++.

Il existe plusieurs versions du simulateur. Nous avons utilisé dans ce projet la version 2.34 qui était la plus récente au début de nos travaux.

Pour simuler un réseau avec *ns*, l'utilisateur doit écrire un script OTcl qui décrit la topologie du réseau, précise la durée de la simulation, initialise l'horloge et les premiers évènements de la simulation. À la fin de la simulation, les statistiques désirées, taux de perte, délai et gigue, peuvent être extraites de la trace produite à l'aide de différents outils.

A.1 Modèle de nœud dans NS

Le simulateur *ns* permet de simuler tous types de réseaux IP, câblés, optiques et sans fil, nécessitant des modules différents. En effet, les paramètres d'initialisation diffèrent selon qu'on simule un réseau Ethernet ou un réseau sans fil. En particulier pour simuler un réseau sans fil 802.11, l'utilisateur doit nécessairement définir dans son script OTcl des paramètres tels que le modèle de propagation, le protocole de routage, le mécanisme d'accès de la couche MAC, le nombre d'interfaces, etc. Par défaut, un nœud sans fil n'a qu'une seule interface dans *ns*. Étant donné que l'utilisation de DiffServ* nécessite qu'un nœud ait trois interfaces, nous avons d'abord modifié l'architecture des nœuds sans fil grâce au tutoriel de [8] afin d'ajouter plusieurs interfaces à un nœud. Par ailleurs, comme nous l'avons montré sur la figure 2.2 du chapitre 2, le fonctionnement de DiffServ* est étroitement lié au protocole de routage. Le routage est l'opération qui permet de déterminer la route vers une destination et le prochain nœud auquel le paquet doit être envoyé. Bien que dans notre étude le routage soit statique,

nous avons intégré le module DiffServ* au protocole de routage MPOLSR qui est un protocole permettant de déterminer puis d'utiliser des routes disjointes entre deux noeuds.

A.2 Modifications apportées au simulateur

Pour intégrer le modèle DiffServ* au simulateur, il a été nécessaire de modifier le module de routage implémenté dans la classe **mpolsr** tandis que pour intégrer le modèle d'évanouissement de Rayleigh, nous avons modifié le module 802.11 implémenté dans la classe **mac-802_11** et créé la classe **CanalSansFil** qui permet de simuler l'évolution de l'état des canaux à partir des paramètres d'entrée tels que le SNR moyen et le PER. Cette classe interagit d'une part avec la classe **mpolsr** pour communiquer à DiffServ* l'état des canaux et d'autre part avec la classe **mac-802_11.cc** car c'est dans cette classe que se fait la spécification du taux de transmission des paquets ainsi que le marquage des paquets corrompus. Enfin, pour simuler le modèle DiffServ dans les réseaux 802.11 nous avons créé la classe **DiffServQueue** qui décrit le fonctionnement du DiffServ standard. Aucune modification n'a été nécessaire pour décrire le fonctionnement des files d'attente FIFO puisque le simulateur disposait déjà d'un module capable de le faire. Cette fonctionnalité est implementée dans la classe **PriQueue**.

Pour obtenir plus de détails sur l'implémentation, le lecteur est invité à lire les codes sources de ces différents modules.

A.3 Cheminement des paquets

Nous expliquons dans cette section le cheminement d'un paquet selon qu'on simule le modèle DiffServ*, DiffServ ou FIFO.

A.3.1 Simulation DiffServ*

Le fonctionnement d'un noeud déployé avec DiffServ* est représenté sur la figure A.1. Lorsque les paquets entrent dans la couche IP d'un noeud, ils sont immédiatement répartis dans leurs files d'attente respectives à l'aide du classificateur. L'ordonnanceur WRR retire ensuite un paquet à la fois et le passe au bloc de routage. À ce stade, il ne s'agit pas de déterminer les routes vers une destination mais on suppose que toutes les routes ont déjà été calculées et qu'on utilise un *round robin* pour distribuer les paquets sur les routes trouvées ou choisir le prochain noeud à visiter. Après la détermination du prochain noeud, l'interface de sortie du paquet doit être sélectionnée. Pour cela, le bloc de sélection demande aux couches MAC l'état de leurs interfaces et sélectionne la couche MAC qui offre le taux de transmission

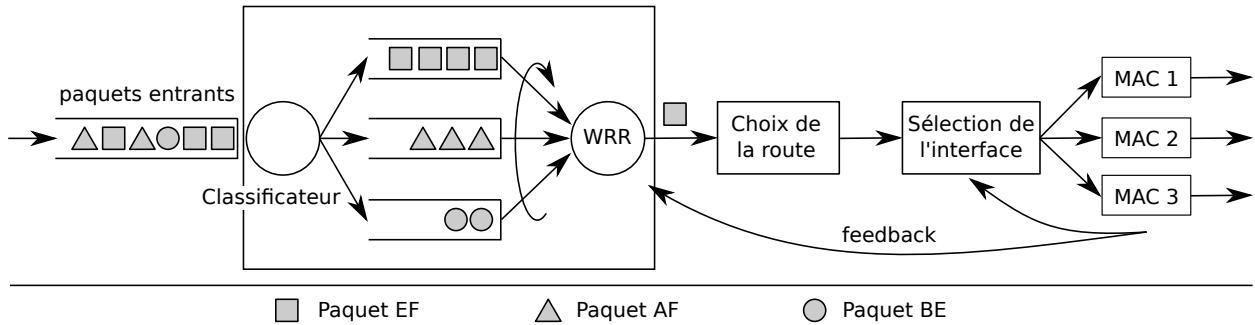


Figure A.1 Cheminement d'un paquet avec DiffServ*

le plus élevé. Après cela, le paquet est transmis à la couche MAC sélectionnée puis est envoyé vers le média physique.

Dans le cas où les poids variables désirent être utilisés, l'ordonnanceur WRR recevra à une fréquence régulière un feedback des couches MAC, indiquant l'état de leurs interfaces. Ainsi l'ordonnanceur modifiera ou non les poids des trois files d'attente et poursuivra son service.

En résumé, voici les actions exécutées par chaque classe.

- classe mpolsr :
 - classificateur,
 - ordonnanceur WRR,
 - choix d'une route,
 - choix d'une interface.
- classe mac-802_11 : mise à jour du taux de transmission.
- classe CanalSansFil : feedback.

A.3.2 Simulation DiffServ et FIFO

Le fonctionnement d'un nœud déployé avec DiffServ est représenté sur la figure A.2. Lorsque les paquets arrivent dans la couche IP d'un nœud, ils sont distribués successivement ou avec un *round robin* sur les routes déjà pré-déterminées. Le routage peut être effectué par n'importe quel protocole de routage multi-chemins. À ce stade, le prochain nœud qui sera visité par un paquet est connu mais on ne connaît pas encore par quelle interface il doit être envoyé. Pour cela, on fait un tirage aléatoire et chaque interface a une probabilité égale à $1/3$ d'être choisie. Le paquet est ensuite envoyé vers l'interface choisie puis mis en attente dans la file qui lui correspond. L'ordonnanceur WRR de l'interface choisie retire ensuite un paquet à la fois et le transmet à la couche MAC de cette interface qui se charge de l'envoyer vers le média physique. Dans le cas où les poids du WRR devraient varier en fonction de l'état

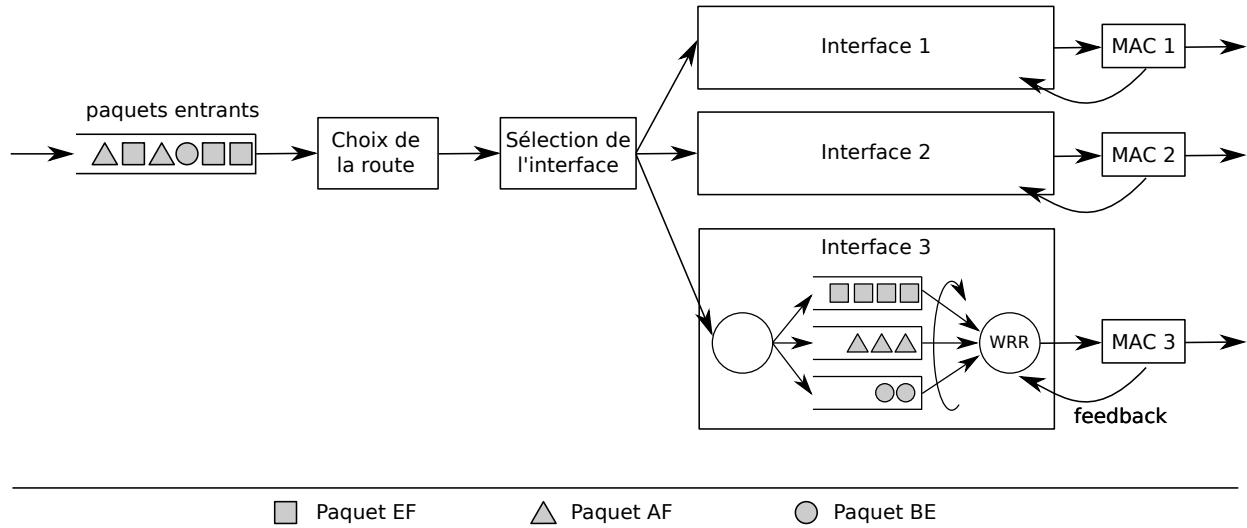


Figure A.2 Cheminement d'un paquet avec DiffServ

du canal, l'ordonnanceur demande périodiquement à la couche MAC de l'interface concernée l'état du canal afin d'actualiser les poids des files.

En résumé, voici les actions exécutées par chaque classe.

- classe mpolsr :
 - choix d'une route,
 - choix d'une interface.
- classe DiffServQueue :
 - classificateur,
 - ordonneur WRR.
- classe mac-802_11 : mise à jour du taux de transmission.
- classe CanalSansFil : feedback.

Le cheminement d'un paquet dans un nœud déployé avec des files FIFO est très similaire au cheminement observé avec DiffServ. Le fonctionnement d'un nœud déployé avec FIFO est représenté sur la figure A.3. Le choix de la route et de l'interface s'opèrent exactement de la même manière que si on simulait DiffServ. Contrairement au parcours avec DiffServ*, on constate qu'en simulant DiffServ ou FIFO, chaque interface possède son propre module de file d'attente.

En résumé, voici les actions exécutées par chaque classe.

- classe mpolsr :
 - choix d'une route,
 - choix d'une interface.

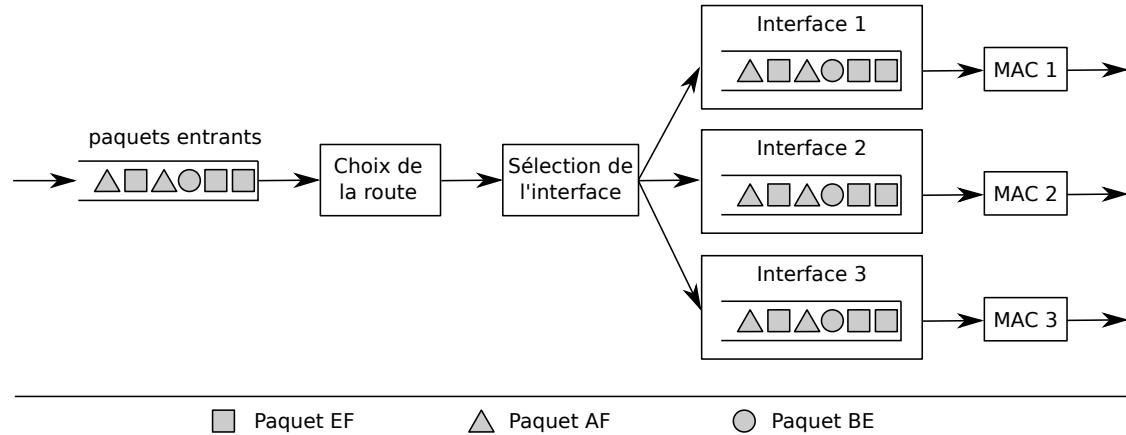


Figure A.3 Cheminement d'un paquet avec FIFO

- classe PriQueue : une seule file pour mettre en attente les paquets
- classe mac-802_11 : mise à jour du taux de transmission.