

## **RAPPORT DE PROJET DE FIN D'ETUDES**

*Filière*

**Ingénieurs en Télécommunications**

*Option*

**Ingénierie des Réseaux**

# **Etude des mécanismes de différentiation de service niveau IP et MAC**

*Elaboré par :*

**Najet Tibi**

*Encadré par :*

**Mlle. Faïza TABBANA**

**Année universitaire : 2004/2005**

# TABLE DES MATIERES

<b>Introduction générale.....</b>	<b>1</b>
<b>Chapitre 1 : Les réseaux locaux sans fil 802.11.....</b>	<b>3</b>
1.1 Topologie d'un réseau WLAN IEEE 802.11 .....	3
1.1.1 Composants d'un réseau WLAN IEEE 802.11 .....	3
1.1.2 Topologie des réseaux WLAN IEEE 802.11 .....	4
1.2 Architecture de la norme IEEE 802.11 .....	6
1.2.1 La couche physique de la norme IEEE 802.11 .....	7
1.2.2 Couche MAC 802.11 .....	10
1.2.3 Les différentes versions de la norme IEEE 802.11 .....	13
<b>Chapitre 2: Protocoles d'accès au médium sans fil pour la norme IEEE 802.11 .....</b>	<b>16</b>
2.1 Le protocole DCF (Distributed Coordination Function) .....	16
2.1.1 Pourquoi CSMA/CA ?.....	16
2.1.2 Description générale du mécanisme DCF .....	17
2.1.3 Espaces inter-trames .....	17
2.1.4 L'algorithme de Backoff .....	19
2.1.5 Mécanisme de détection virtuelle .....	22
2.2 Le protocole PCF (Point Coordination Function) .....	22
2.3 Limites en terme de qualité de service des mécanismes DCF-PCF et du standard 802.11e .....	24
2.3.1 Limites de DCF (Distributed Coordination Function) .....	24
2.3.2 Limites de PCF (Point Coordination Function) .....	24
2.3.3 La norme 802.11e .....	25
<b>Chapitre 3 : Etude des mécanismes de différenciation de services.....</b>	<b>27</b>
3.1 Le mécanisme de différenciation de service niveau MAC : EDCF (Enhanced Distributed Coordination Function) .....	28
3.2 Les mécanismes de différenciation de service niveau IP : IntServ / DiffServ.....	30
3.2.1 Le protocole à intégration de service IntServ.....	30

---

3.2.2	Le protocole à Différentiation de service DiffServ .....	34
3.3	Architecture de différenciation de service niveau IP et/ou MAC .....	41
3.3.1	Position du problème : .....	41
3.3.2	Solution proposée : association des paramètres de DiffServ et l'EDCF : .....	42
3.3.3	Exemple de mapping: .....	44
<b>Chapitre 4 : Simulation et évaluation des performances de DiffServ et 802.11e sous NS-2.....</b>		<b>47</b>
4.1	Présentation du Network Simulator .....	47
4.2	Modèle de l'EDCF implémenté.....	48
4.2.1	Présentation de L'EDCF.....	48
4.2.2	Paramètres de simulation.....	50
4.3	Modèle de DiffServ .....	52
4.4	Paramètres de Simulation.....	53
4.4.1	Débit utile (throughput) .....	53
4.4.2	Le taux de pertes.....	54
4.4.3	Le délai.....	54
4.5	Simulation : résultats et interprétations .....	54
4.5.1	Introduction .....	54
4.5.2	EDCF (résultats et interprétations) : .....	54
4.5.3	DiffServ (résultats et interprétations) .....	57
4.5.4	Couplage de DiffServ et EDCF (résultats et interprétations) : .....	58
4.5.5	Evaluation des performances des approches simulées : .....	61
<b>Conclusion générale .....</b>		<b>66</b>
<b>Bibliographie.....</b>		<b>68</b>
<b>Annexe : Le simulateur NS-2 .....</b>		<b>70</b>

---

# LISTE DES FIGURES

Figure 1.1 Mode infrastructure .....	5
Figure 1.2 Mode Ad Hoc .....	6
Figure 2. 1 L'accès au médium en mode DCF .....	17
Figure 2. 2 Les relations entre différents IFS .....	19
Figure 2. 3 Procédure de Backoff.....	20
Figure 2. 4 La procédure CSMA/CA et Algorithme de Backoff .....	21
Figure 2. 5 Mécanisme d'écoute virtuelle VSC .....	22
Figure 2. 6 Succession de deux super-trames PCF.....	23
Figure 2. 7 Mécanisme de transmission pour les deux périodes CFP et CP .....	24
Figure 2. 8 Structure d'une super-trame HCF .....	26
Figure 3. 1 Une station implémentant IEEE802.11e .....	29
Figure 3. 2 L'accès en mode EDCF .....	29
Figure 3. 3 Structure temporelle du CFB .....	30
Figure 3. 4 Architecture de IntServ.....	32
Figure 3. 5 Sens des messages RSVP.....	33
Figure 3. 6 Entête d'un datagramme IPV4 .....	34
Figure 3. 7 Traitement d'un paquet par un edge router .....	35
Figure 3. 8 Traitement d'un paquet par un core router.....	36
Figure 3. 9 Différentiation avec AF .....	38
Figure 3. 10 Structure du réseau bout en bout .....	42
Figure 3. 11 Champ Qos de la trame 802.11 .....	43
Figure 3. 12 Qos combinant DiffServ et 802.11e.....	45
Figure 4. 1 Scénario EDCF .....	55
Figure 4. 2 Courbe de débit avec EDCF (Kbits/s).....	55
Figure 4. 3 Courbe de délai avec EDCF.....	56

Figure 4. 4 Courbe de pertes avec EDCF .....	57
Figure 4. 5 Topologie avec DiffServ.....	57
Figure 4. 6 Courbe de débit utile avec DiffServ.....	58
Figure 4. 7 Topologie à simuler : gestion de bout en bout .....	58
Figure 4. 8 Débit utile avec Qos (Kbits/s) .....	59
Figure 4. 9 Délai avec Qos de bout en bout .....	60
Figure 4. 10 Courbe de pertes avec Qos de bout en bout .....	61
Figure 4. 11 Débit util avec/sans Mapping .....	62
Figure 4. 12 Pertes avec/sans Mapping .....	63
Figure 4. 13 Débit utile avec/sans mapping .....	64
Figure 4. 14 Pertes avec/sans mapping.....	65

# LISTE DES TABLEAUX

Tableau 1. 1 Canaux FHSS .....	9
Tableau 1. 2 Les différents nomes 802.11 .....	15
Tableau 2. 1 Valeurs des IFS et du time Slot en fonction de la couche physique .....	19
Tableau 3. 1 Couplage de DiffServ et schéma de Qos pour la norme 802.11 .....	44
Tableau 3. 2 Exemple de Mapping .....	44
Tableau 4. 1 Les paramètres des ACs de l'EDCF implémenté (draft novembre 2003) .....	50
Tableau 4. 2 Les paramètres des trafics simulés .....	51
Tableau 4. 3 Association trafic/AC .....	52

# ACRONYMES

## A

AC Access Categorie

ACK ACKnowledgment

AES Advanced Encryption Standard

AIFS Arbitration Inter-Frame Spacing

AIFSN Arbitration Inter-Frame Spacing Number

AP Access Point

ARQ Automatic Repeat request

## B

BB Black Burst

BS Base Station

BSA Basic Service Area

BSS Basic Service Set

## C

CBR Constant Bit Rate

CF-END Contention Free End

CFP Contention Free Period

CF-POLL Contention Free Poll

CP Contention Period

CRC Cyclic Redundancy Check

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CSMA/CD Carrier Sense Multiple Access with Collison Detection

CTS Clear To Send

CW Contention Window

CTS Clear To Send

CW Contention Window

## **D**

DCF Distributed Coordination Function

DFWMAC Distributed Foundation Wireless Medium Access Control

DIFS DCF IFS

DS Distribution System

DSSS Direct Sequence Spread Spectrum

## **E**

EDCF Enhanced Distributed Coordination Function

ESS Extended Service System

ETSI European Telecommunications Standards Institute

## **F**

FHSS Frequency Hopping Spread Spectrum

## **H**

HC Hybrid Coordinator

HCF Hybrid Coordination Function

HiperLAN High Performance European Radio LAN

## **I**

IBSS Independent BSS

ID IDentification

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IFS Inter Frame Spacing

IP Internet Protocol

ISO International Organization for Standardization

## **L**

LAN Local Area Network

## **M**

MAC Medium Access Control

MSDU Mac Service Data Unit

## **N**

NACK Negative ACKnowledgement

NAV Network Allocation Vector

NS Network Simulator

## **O**

OFDM Orthogonal Frequency Division Multiplexing

OSI Open System Interconnection

OTCL Object Tool Command Language

## **P**

PCF Point Coordination Function

PDA Personal Digital Assistant

PHY PHYSical layer

PIFS Polling IFS

PLCP Physical Layer Convergence Procedure

PMD Physical Media Dependant<sup>2</sup>

## **Q**

QoS Quality of Service

## **R**

RTS Request To Send

RSVP Resource Reservation Protocol

## **S**

S-Aloha Slotted Aloha

SIFS Short IFS

STA Station

## **T**

TC Traffic Category

TCID Traffic Category Identifier

TDMA Time Division Multiple Access

TXOP Transmission Opportunity

## **U**

UDP User Datagram Protocol

## **V**

VCS Virtual Carrier Sense

## **W**

WLAN Wireless Local Area Network

WT Wireless Terminal

# Introduction générale

---

Les radiocommunications utilisent les ondes électromagnétiques afin de permettre une communication à distance entre plusieurs systèmes. Il peut s'agir de téléphones, de satellites, d'ordinateurs,... De nos jours, on assiste à un développement spectaculaire des réseaux sans fils tant au niveau débit qu'au niveau architecture dans le but de permettre plus de mobilité et de fiabilité à faible coût.

Grâce aux réseaux locaux sans fil, il est possible de créer des réseaux locaux sans fils à haut débit pour peu que la station à connecter ne soit pas trop distante par rapport au point d'accès. De nos jours, les réseaux locaux sans fil connaissent une évolution considérable grâce à des normalisations rapides, un développement continu et une commercialisation réussie.

En effet, différents organismes de normalisation ont défini des standards pour les réseaux locaux sans fil :

- IEEE 802.11: Wireless Local Area Networks,
- IEEE 802.15: Working Group for Wireless Personal Area Networks, Bluetooth,
- IEEE 802.16: Working Group on Broadband Wireless Access Standards,
- ETSI: HIPERLAN (High Performance Radio Local Area Network) type 1, 2, 3 et 4.

La norme IEEE 802.11 a été publiée en 1997. Il s'agit aujourd'hui de la norme des réseaux locaux sans fil la plus déployée. Cette norme offre deux modes d'interconnexion : le mode infrastructure et le mode Ad Hoc.

Ainsi, ces réseaux offrent aujourd'hui une qualité de service insuffisante : pertes dues aux mobilité de nœuds combinées à la faible portée radio, perturbations des transmissions, incapacité à gérer un grand nombre d'utilisateurs simultanés,... Cette problématique a engendré la naissance de nouvelles versions de la norme 802.11 à savoir la norme 802.11e. Ce dernier standard a été défini dans le but d'améliorer la qualité de service (QoS) au niveau MAC pour les réseaux locaux sans fil IEEE 802.11, c'est-à-dire de garantir la bande passante, le délai pour la transmission des données tel que la voix, la vidéo, la voix sur IP,...

Dans ce contexte, des travaux sont effectués pour offrir une meilleure qualité de service aux applications multimédia reposent essentiellement sur certains aspects liés aux réseaux ad hoc,

à savoir : le routage avec qualité de service, la réservation de ressource et l'introduction de la différenciation de service au niveau de la couche MAC.

Ces mécanismes de qualité de service permettent de gérer au mieux les ressources du réseau, dans le but de satisfaire les différents besoins de qualité de service pour les applications multimédia.

Dans notre projet nous nous intéressons plus particulièrement au niveau MAC et IP. Dans un premier temps, on se propose de simuler l'EDCF (Enhanced Distribution Coordination Function) et en second phase de faire un couplage entre l'EDCF et l'architecture DiffServ proposée par l'IETF.

Dans le premier, nous présenterons les spécificités générales de la norme IEEE 802.11. Nous commencerons par décrire les différents modes topologiques des réseaux sans fil 802.11, puis nous nous intéresserons à l'architecture logique de la norme.

Dans le deuxième chapitre, nous détaillerons le mécanisme de contrôle d'accès au support sans fil, en mode distribué (protocole DCF) et en mode centralisé (protocole PCF). Dans la dernière partie de ce chapitre, nous évoquons les limites des deux protocoles DCF et PCF en terme de qualité de service, et nous présenterons la version 802.11e de la norme qui vise à remédier à ces insuffisances en terme de qualité de service.

Dans le troisième chapitre, nous détaillerons le mécanisme de différenciation de service au niveau IP. Nous commencerons par présenter la problématique liée au réseau Internet, en montrant par la suite les solutions proposées, à savoir les services intégrés et différenciés, en présentant l'architecture de chaque mécanisme ainsi que le mode de fonctionnement de chacun, nous passerons par la suite à traiter les limites de l'architecture InServ expliquant ainsi le choix de l'architecture DiffServ. Dans la dernière partie de ce chapitre, nous présentons l'architecture proposée qui consiste à effectuer un couplage des deux mécanismes de différenciation de service niveau MAC et IP.

Le dernier chapitre comportera trois parties. Les deux premières parties seront consacrées à l'évaluation des résultats des simulations relatifs à chacune des deux mécanismes de différenciation de service ainsi que les limites de l'EDCF.

Finalement, nous présenterons les simulations relatives à l'architecture proposée.

# Chapitre 1 : Les réseaux locaux sans fil

## 802.11

---

Avec la récente adoption de nouveaux standards pour les réseaux locaux (LAN) sans fil haut débit, les utilisateurs nomades disposent désormais de performances, de débits et de disponibilités comparables à ceux des réseaux Ethernet filaires classiques. Les LAN sans fil (Wireless LAN), dont fait partie le Wi-Fi, sont devenus une solution de choix pour la mise en place d'un réseau.

Le standard IEEE 802.11 est un système de transmission de données assurant la liaison entre les périphériques par les ondes radio plutôt que par un réseau filaire. L'IEEE (Institute of Electrical and Electronics Engineers) a ratifié la spécification 802.11, norme régissant les réseaux locaux sans fil, en 1997.

Le standard 802.11 couvre sur les deux premières couches inférieures du modèle OSI, la couche physique et la couche de liaison de données. Toutes les applications réseau, tous les protocoles réseaux fonctionnent aussi simplement sur un réseau 802.11 que sur Ethernet [1]

Dans ce chapitre, nous commencerons dans une première partie par décrire les topologies suivant lesquels les WLAN 802.11 fonctionnent.

Ensuite, nous présenterons les caractéristiques liées à l'architecture logique de la norme (couche physique et couche MAC) et les différentes versions du standard.

### 1.1 Topologie d'un réseau WLAN IEEE 802.11

Les réseaux locaux sans fil mettent en œuvre plusieurs composants qui interagissent suivant deux modes topologiques : mode infrastructure et mode Ad Hoc [2]

#### 1.1.1 Composants d'un réseau WLAN IEEE 802.11

Il existe trois composants majeurs pour la mise en place des réseaux locaux sans fil : les points d'accès (AP), les terminaux mobiles (MT) et les ponts sans fil (WB).

### 1.1.1.1 Les points d'accès (Access Point : AP)

Les points d'accès sont l'équivalent des *hubs* dans les réseaux locaux ordinaires. Ils contrôlent les stations appartenant à une même cellule.

Ils permettent, en outre, le déplacement de stations d'une cellule à une autre, sans interrompre la communication (*roaming*).

Rattachés au *backbone* du réseau, les points d'accès servent aussi de relais entre le réseau sans-fil et d'autres réseaux, qui peuvent être filaires ou non.

### 1.1.1.2 Les terminaux mobiles (Mobile Terminal : MT)

Ce sont de simples stations (PC portables ou autres) équipées de cartes intégrant un émetteur et un récepteur. Ces stations peuvent utiliser n'importe quel système d'exploitation pour supporter les cartes 802.11.

Les antennes généralement utilisées par les MTs et les APs sont des antennes internes omnidirectionnelles de portée ne dépassant pas les 160 mètres.

Des antennes directives et à gain plus élevé peuvent être utilisées pour couvrir des stations à quelques kilomètres de l'émetteur.

### 1.1.1.3 Les ponts sans fil (Wireless Bridge : WB)

Les ponts sans-fil sont utilisés essentiellement pour relier deux réseaux locaux sans fils WLAN éloignés ou isolés l'un par rapport à l'autre, par l'intermédiaire d'antennes directives à gain plus ou moins élevé.

## 1.1.2 Topologie des réseaux WLAN IEEE 802.11

L'entité la plus élémentaire d'une architecture 802.11 est le *BSS (Basic Service Set)*. Un BSS est défini comme étant un ensemble de plusieurs stations qui communiquent entre elles, soit par l'intermédiaire d'un *point d'accès* (mode infrastructure), soit directement (mode Ad Hoc) [1].

Le standard 802.11 définit deux modes opératoires :

- Le mode infrastructure dans lequel les dients sans fils sont connectés à un point d'accès. Il s'agit généralement du mode par défaut des cartes 802.11b.

- Le mode ad hoc dans lequel les clients sont connectés les uns aux autres sans aucun point d'accès.

### 1.1.2.1 Mode infrastructure

En mode infrastructure chaque station (notée STA) se connecte à un point d'accès via une liaison sans fil. L'ensemble formé par le point d'accès et les stations situés dans sa zone de couverture est appelé *ensemble de services de base* (en anglais *basic service set*, noté BSS) et constitue une cellule. Chaque BSS est identifié par un BSSID, un identifiant de 6 octets (48 bits). Dans le mode infrastructure, le BSSID correspond à l'adresse MAC du point d'accès.

Il est possible de relier plusieurs points d'accès entre eux (ou plus exactement plusieurs BSS) par une liaison appelée système de distribution (notée DS pour *Distribution System*) afin de constituer un *ensemble de services étendu* (*extended service set* ou ESS). Le système de distribution (DS) peut être aussi bien un réseau filaire, qu'un câble entre deux points d'accès ou bien même un réseau sans fil

Un ESS est repéré par un ESSID (*Extended Service Set Identifier*), c'est-à-dire un identifiant de 32 caractères de long (au format ASCII) servant de nom pour le réseau. L'ESSID, souvent abrégé en SSID, représente le nom du réseau et représente en quelque sorte un premier niveau de sécurité dans la mesure où la connaissance du SSID est nécessaire pour qu'une station se connecte au réseau étendu.

Lorsqu'un utilisateur nomade passe d'un BSS à un autre lors de son déplacement au sein de l'ESS, l'adaptateur réseau sans fil de sa machine est capable de changer de point d'accès selon la qualité de réception des signaux provenant des différents points d'accès. Les points d'accès communiquent entre eux grâce au système de distribution afin d'échanger des informations sur les stations et permettre le cas échéant de transmettre les données des stations mobiles. Cette caractéristique permettant aux stations de "passer de façon transparente" d'un point d'accès à un autre est appelé *itinérance* (roaming).

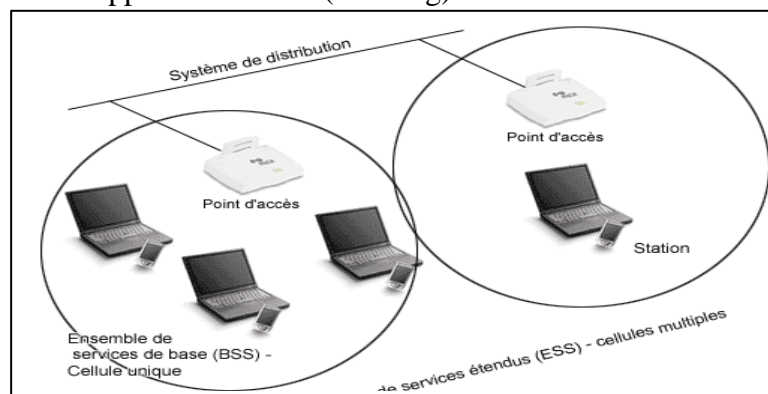


Figure 1.1 Mode infrastructure

### 1.1.2.2 Mode Ad Hoc

En mode ad hoc les machines sans fils clientes se connectent les unes aux autres afin de constituer un réseau point à point (peer to peer ), c'est-à-dire un réseau dans lequel chaque machine joue en même temps le rôle de client et le rôle de point d'accès.

L'ensemble formé par les différentes stations est appelé ensemble de services de base indépendants (Independent basic service set, abrégé en IBSS).

Un IBSS est ainsi un réseau sans fil constitué au minimum de deux stations et n'utilisant pas de point d'accès. L'IBSS constitue donc un réseau permettant à des utilisateurs situés dans une même salle d'échanger des données. Il est identifié par un SSID, comme l'est un ESS en mode infrastructure.

Dans un réseau ad hoc, la portée du *BSS indépendant* est déterminée par la portée de chaque station. Cela signifie que si deux des stations du réseaux sont hors de portée l'une de l'autre, elles ne pourront pas communiquer, même si elles "voient" d'autres stations. En effet, contrairement au mode infrastructure, le mode *ad hoc* ne propose pas de système de distribution capable de transmettre les trames d'une station à une autre. Ainsi un IBSS est par définition un réseau sans fil restreint [3]

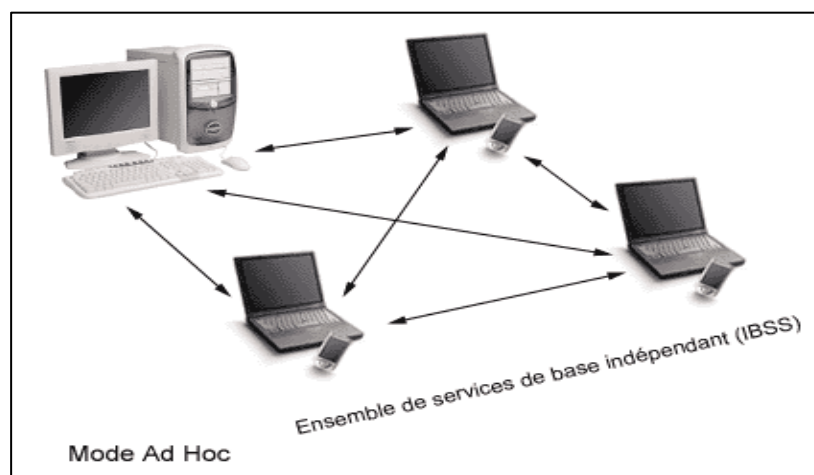
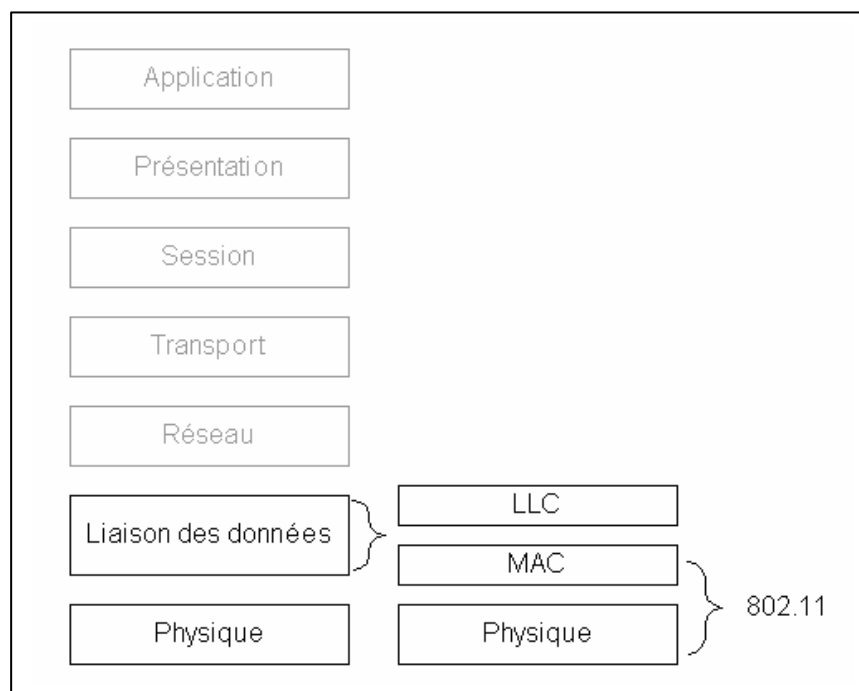


Figure 1.2 Mode Ad Hoc

## 1.2 Architecture de la norme IEEE 802.11

La norme 802.11 s'adresse essentiellement aux niveaux lien et physique du modèle OSI. En fait, elle introduit des modifications sur la couche basse du niveau lien (donc niveau MAC) et sur le niveau physique avec le support de plusieurs méthodes d'accès radio (donc la définition de plusieurs couches physiques). Il est à noter que la nouvelle couche MAC est commune à toutes les couches physiques [4]

La figure 1.3 illustre l'architecture en couches de la norme IEEE 802.11 comparée à celle du modèle OSI.



**Figure 1.3 Architecture en couche de la norme 802.11**

### 1.2.1 La couche physique de la norme IEEE 802.11

Selon la définition du modèle OSI, «la couche physique fournit les moyens mécaniques, électriques, fonctionnels et procéduraux nécessaires à l'activation, au maintien et à la désactivation des connexions physiques destinées à la transmission des éléments binaires entre entités de liaison» [4]. S'agissant des réseaux locaux sans fils WLAN, la couche physique définit la modulation des ondes radioélectriques et les caractéristiques de signalisation pour la transmission de données.

La norme IEEE 802.11 définit deux sous-couches physiques : la sous couche PMD (Physical Media Dependant) et la sous couche PLCP (Physical Layer Convergence Procedure) [1]. La sous-couche PMD gère l'encodage des données et la modulation, tandis que la sous-couche PLCP s'occupe de l'écoute du support et est directement reliée à la couche MAC pour lui signifier que le support de transmission est libre.

D'après la norme IEEE 802.11, la sous-couche PMD peut se baser sur l'une des techniques de transmission suivantes :

- FHSS (Frequency Hopping Spread Spectrum): étalement de spectre à saut de fréquence.
- DSSS (Direct Sequence Spread Spectrum) : étalement de spectre à séquence directe.

- IR (InfraRed) : s'appuie sur la lumière infrarouge.

A ces trois techniques de base, viennent s'ajouter trois versions du protocole IEEE 802.11 pour la couche physique, 802.11b (Wi-Fi), 802.11a (Wi-Fi5) et 802.11g.

### 1.2.1.1 Étalement de spectre à saut de fréquence (FHSS)

La technique **FHSS** (Frequency Hopping Spread Spectrum[2], en français étalement de spectre par saut de fréquence ou étalement de spectre par évaison de fréquence) consiste à découper la large bande de fréquence en un minimum de 75 canaux (hops ou sauts d'une largeur de 1MHz), puis de transmettre en utilisant une combinaison de canaux connue de toutes les stations de la cellule. Dans la norme 802.11, la bande de fréquence 2.4 - 2.4835 GHz permet de créer 79 canaux de 1 MHz. La transmission se fait ainsi en émettant successivement sur un canal puis sur un autre pendant une courte période de temps (d'environ 400 ms), ce qui permet à un instant donné de transmettre un signal plus facilement reconnaissable sur une fréquence donnée[2]

L'étalement de spectre par saut de fréquence a originalement été conçu dans un but militaire afin d'empêcher l'écoute des transmissions radio. En effet, une station ne connaissant pas la combinaison de fréquence à utiliser ne pouvait pas écouter la communication car il lui était impossible dans le temps imparti de localiser la fréquence sur laquelle le signal était émis puis de chercher la nouvelle fréquence [2]

Aujourd'hui les réseaux locaux utilisant cette technologie sont standards ce qui signifie que la séquence de fréquences utilisées est connue de tous, l'*étalement de spectre par saut de fréquence* n'assure donc plus cette fonction de sécurisation des échanges. En contrepartie, le *FHSS* est désormais utilisé dans le standard 802.11 de telle manière à réduire les interférences entre les transmissions des diverses stations d'une cellule.

### 1.2.1.2 Étalement de spectre à séquence directe (DSSS)

La technique **DSSS** (*Direct Sequence Spread Spectrum*, étalement de spectre à séquence directe) consiste à transmettre pour chaque bit une séquence Barker (parfois appelée bruit pseudo-aléatoire ou en anglais pseudo-random noise, noté PN) de bits. Ainsi chaque bit valant 1 est remplacé par une séquence de bits et chaque bit valant 0 par son complément.

La couche physique de la norme 802.11 définit une séquence de 11 bits (*10110111000*) pour représenter un 1 et son complément (*01001000111*) pour coder un 0. On appelle *chip* ou

chipping *code* (en français *puce*) chaque bit encodé à l'aide de la séquence. Cette technique (appelée chipping) revient donc à moduler chaque bit avec la séquence *barker* [2]

Grâce au chipping, de l'information redondante est transmise, ce qui permet d'effectuer des contrôles d'erreurs sur les transmissions [2]

Dans le standard 802.11b, la bande de fréquence 2.400-2.4835 GHz (d'une largeur de 83.5 MHz) a été découpée en 14 canaux séparés de 5MHz, dont seuls les 11 premiers sont utilisables aux Etats-Unis. Seuls les canaux 10 à 13 sont utilisables en France. Voici les fréquences associées aux 14 canaux :

Canal	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Fréquence (GHz)	2.412	2.417	2.422	2.427	2.432	2.437	2.442	2.447	2.452	2.457	2.462	2.467	2.472	2.477

**Tableau 1. 1 Canaux FHSS**

Toutefois, pour une transmission de 11 Mbps correcte il est nécessaire de transmettre sur une bande de 22 MHz car, d'après le théorème de Shannon, la fréquence d'échantillonnage doit être au minimum égale au double du signal à numériser. Ainsi certains canaux recouvrent partiellement les canaux adjacents, c'est la raison pour laquelle des canaux isolés (les canaux 1, 6 et 11) distants les uns des autres de 25MHz sont généralement utilisés [2]

Ainsi, si deux points d'accès utilisant les mêmes canaux ont des zones d'émission qui se recoupent, des distorsions du signal risquent de perturber la transmission. Ainsi pour éviter toute interférence il est recommandé d'organiser la répartition des points d'accès et l'utilisation des canaux de telle manière à ne pas avoir deux points d'accès utilisant les mêmes canaux proches l'un de l'autre.

Le standard 802.11a utilise la bande de fréquence 5.15GHz à 5.35GHz et la bande 5.725 GHz à 5.825 GHz, ce qui permet de définir 8 canaux distincts d'une largeur de 20Mhz chacun, c'est-à-dire une bande suffisamment large pour ne pas avoir de parasitage entre canaux [6]

### 1.2.1.3 L'Infrarouge (IR)

Le standard IEEE 802.11 prévoit également une alternative à l'utilisation des ondes radio : la lumière infrarouge. La technologie infrarouge a pour caractéristique principale d'utiliser une onde lumineuse pour la transmission de données. Ainsi les transmissions se font de façon uni-

directionnelle, soit en "vue directe" soit par réflexion. Le caractère non dissipatif des ondes lumineuses offre un niveau de sécurité plus élevé.

Il est possible grâce à la technologie infrarouge d'obtenir des débits allant de 1 à 2 Mbit/s en utilisant une modulation appelé **PPM** (*Pulse Position Modulation*).

La modulation *PPM* consiste à transmettre des impulsions à amplitude constante, et à coder l'information suivant la position de l'impulsion. Le débit de 1 Mbps est obtenu avec une modulation de *16-PPM*, tandis que le débit de 2 Mbps est obtenu avec une modulation *4-PPM* permettant de coder deux bits de données avec 4 positions possibles [2]

La méthode IR se base sur la diffusion d'une lumière infrarouge de longueur d'onde comprise entre 850 et 950 nm (nanomètres). Grâce aux caractéristiques réfléchives de l'infrarouge, les stations appartenant au réseau ne doivent pas nécessairement être dirigées les unes vers les autres. Cependant, vu la portée très faible de l'infrarouge, les stations ne peuvent être éloignées les unes des autres de plus d'une dizaine de mètres. Un réseau 802.11 IR ne peut donc être déployé que dans un espace ayant la dimension d'une pièce.

### 1.2.2 Couche MAC 802.11

La couche Liaison de données de la norme 802.11 est composé de deux sous-couches : la couche de contrôle de la liaison logique (Logical Link Control, notée LLC) et la couche de contrôle d'accès au support (Media Access Control, ou MAC) [4]

En plus des fonctions habituellement rendues par la couche MAC, la couche MAC 802.11 offre d'autres fonctions qui sont normalement confiées aux protocoles supérieurs, comme :

- la fragmentation et le réassemblage des trames
- le contrôle d'accès au support
- l'adressage et le formatage des trames.
- le contrôle d'erreur sur la trame, à partir d'un CRC (Cyclic Redundancy Check)
- la qualité de service
- la gestion de l'énergie
- la gestion de la mobilité
- la sécurité

Le contrôle d'accès au support est une fonctionnalité qui nous intéresse ici particulièrement, se fait suivant deux méthodes (DCF et PCF) qui seront étudiées ultérieurement.



- Adresse 3 (6 octets) : Adresse perdue, par exemple si FromDS = 1, Adresse 2 contient l'adresse du point d'accès et Adresse 3 celle de la station source d'origine
- *Sequence Control* (2 octets) : représente l'ordre des différents fragments d'une même trame. Ce champ permet aussi de reconnaître la duplication des paquets. Il est constitué de deux sous champs : *Fragment Number* et *Sequence Number* pour respectivement la trame et l'indice du fragment dans cette trame.
- Adresse 4 (6 octets) : Utilisée dans des cas spéciaux tels que la transmission entre deux points d'accès, quand les ToDS et FromDS sont à 1.

### b. Les trames de contrôle MAC 802.11

La norme a prévu d'autres formats pour les trames de contrôle, en particulier les trames RTS, CTS et ACK.

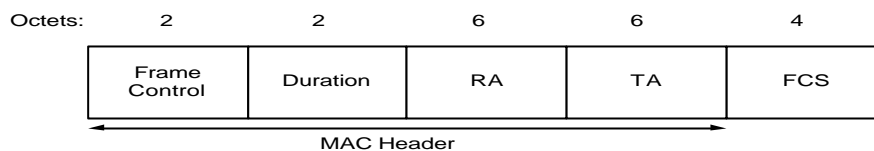
- Les trames RTS et CTS sont utilisées pour la réservation virtuelle des ressources dans le cadre de la procédure d'accès au support physique.
- La trame ACK est utilisée pour acquitter les transmissions réussies. Elle est envoyée par une station réceptrice, ayant correctement reçu une trame de données, à la station source.

Les trames RTS, CTS et ACK sont constituées chacune par un FCS (*Frame Check Sequence*) et un entête MAC.

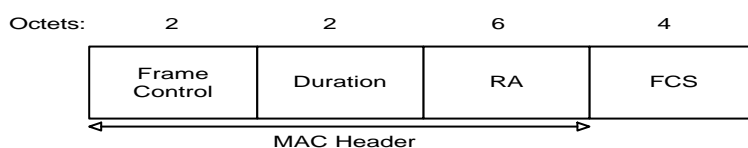
L'entête MAC comporte quelques différences suivant qu'il s'agisse de trames RTS, CTS ou ACK :

- L'entête de la trame RTS comprend les champs suivant :
  - ? Frame Control : analogue au champs de la trame de données MAC.
  - ? Duration : Durée à réserver.
  - ? RA : Adresse de la station réceptrice.
  - ? TA : Adresse de la station émettrice.
- L'entête de la trame CTS comprend les même champs que celui de RTS, hormis le champs TA. Le champ RA étant recopié à partir du champ TA de la trame RTS reçue.

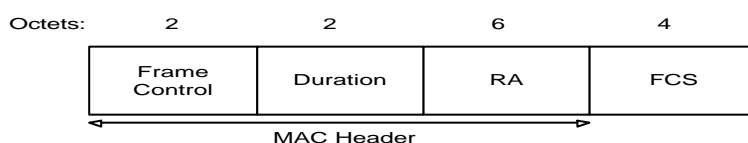
- L'entête de la trame ACK possède un format similaire à celui de CTS. L'adresse RA est recopiée à partir du champ Adresse 2 de la trame MAC à acquitter.



**Figure 1.5 Format de la trame RTS**



**Figure 1.6 Format de la trame CTS**



**Figure 1.7 Format de la trame ACK**

### 1.2.3 Les différentes versions de la norme IEEE 802.11

La norme *IEEE 802.11* est en réalité la norme initiale offrant des débits de 1 ou 2 Mbps. Des révisions ont été apportées à la norme originale afin d'optimiser le débit (c'est le cas des normes 802.11a, 802.11b et 802.11g, appelées normes 802.11 physiques) ou bien préciser des éléments afin d'assurer une meilleure sécurité ou une meilleure interopérabilité. Voici un tableau présentant les différentes révisions de la norme 802.11 et leur signification [8]

Norme	Description
<b>802.11a</b>	La norme 802.11a (baptisée <i>WiFi 5</i> ) permet d'obtenir un haut débit (54 Mbps théoriques, 30 Mbps réels). 802.11a spécifie 8 canaux radio dans la bande de fréquence des 5 GHz.

<p><b>802.11b</b></p>	<p>Désignée aussi par Wifi, la norme 802.11b est la norme la plus répandue actuellement. Elle propose un débit théorique de 11 Mbps (6 Mbps réels) avec une portée pouvant aller jusqu'à 300 mètres dans un environnement dégagé. La plage de fréquence utilisée est la bande des 2.4 GHz, avec 3 canaux radio disponibles.</p>
<p><b>802.11c</b></p>	<p>La norme 802.11c n'a pas d'intérêt pour le grand public. Il s'agit uniquement d'une modification de la norme 802.11d afin de pouvoir établir un pont avec les trames 802.11 (niveau <i>liaison de données</i>).</p>
<p><b>802.11d</b></p>	<p>La norme 802.11d est un supplément à la norme 802.11 dont le but est de permettre une utilisation internationale des WLAN 802.11. Son but est de permettre aux différents équipements d'échanger des informations sur les plages de fréquence et les puissances autorisées dans le pays d'origine du matériel.</p>
<p><b>802.11e</b></p>	<p>La norme 802.11e vise à donner des possibilités en matière de qualité de service au niveau de la couche <i>MAC</i>. Cette norme a pour but de définir les besoins des différents paquets en terme de bande passante et de délai de transmission de telle manière à permettre notamment une meilleure transmission de la voix et de la vidéo.</p>
<p><b>802.11f</b></p>	<p>La norme 802.11f est une recommandation à l'intention des vendeurs de point d'accès pour une meilleure interopérabilité des produits. Elle propose le protocole <i>Inter-Access point roaming protocol</i> permettant à un utilisateur itinérant de changer de point d'accès de façon transparente lors d'un déplacement, quelles que soient les marques des points d'accès présentes dans l'infrastructure réseau. Cette possibilité est appelée <i>itinérance</i> (ou <i>roaming</i>)</p>
<p><b>802.11g</b></p>	<p>La norme 802.11g offrira un haut débit (54 Mbps théoriques, 30 Mbps réels) sur la bande de fréquence des 2.4 GHz. cette norme 802.11g a une compatibilité ascendante avec la norme 802.11b, ce qui signifie que des matériels conformes à 802.11g pourront fonctionner en 802.11b</p>

<b>802.11h</b>	La norme <i>802.11h</i> vise à rapprocher 802.11 du standard Européen (HiperLAN 2, d'où le <i>h</i> de 802.11h) et être en conformité avec la réglementation européenne en matière de fréquence et d'économie d'énergie.
<b>802.11i</b>	La norme 802.11i a pour but d'améliorer la sécurité des transmissions (gestion et distribution des clés, chiffrement et authentification). Cette norme s'appuie sur l'AES (Advanced Encryption Standard) et propose un chiffrement des communications pour les transmissions utilisant les technologies 802.11a, 802.11b et 802.11g
<b>802.11j</b>	La norme <i>802.11j</i> est à la réglementation japonaise ce que le 802.11h est à la réglementation européenne.

**Tableau 1. 2 Les différents normes 802.11**

## CONCLUSION

Au de ce chapitre, nous avons présenté la technologie IEEE802.11. En effet les réseaux locaux sans fil IEEE802.11, couvrent les deux premières couches du modèle OSI : couche physique et la couche MAC. Dans la suite de ce rapport, on s'intéresse à l'étude de la couche MAC et plus précisément au niveau au mécanisme de différenciation de service niveau IP et MAC.

# Chapitre 2 : Protocoles d'accès au médium sans fil pour la norme IEEE 802.11

---

L'accès au support est déterminé par une fonction dite fonction de coordination : c'est une fonction logique qui détermine l'instant d'émission/réception d'une station associée à un BSS. Le standard définit deux méthodes d'accès au support : DCF (Distributed Coordination Function) et PCF (Point Coordination Function). La première utilise la technique CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) pour assurer arbitrer au support. Elle est conçue de façon à ce que tous les utilisateurs aient une chance égale d'accéder au médium. La deuxième technique est basée sur l'interrogation régulière, par le point d'accès, de l'ensemble des stations pour leur demander s'ils ont des données à transmettre [1]

Chaque station, pour qu'elle soit interrogée, doit s'enregistrer et réserver préalablement un temps d'émission auprès du point d'accès. Actuellement, la plupart du matériel Wi-Fi disponible sur le marché n'implémente que la méthode d'accès DCF, que ce soit pour les points d'accès ou les stations clientes.

## 2.1 Le protocole DCF (Distributed Coordination Function)

La technique DCF est basée sur le mécanisme CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) ou méthode d'accès multiple à détection de porteuse et évitement de collision. Cet algorithme distribué est exécuté localement sur chaque station afin de déterminer les périodes d'accès au médium.

### 2.1.1 Pourquoi CSMA/CA ?

Les réseaux locaux sans fils adoptent la méthode d'accès CSMA/CA au lieu de la méthode CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) généralement utilisée dans les réseaux LANs classiques [4]

La méthode CSMA/CD consiste, pour une station désirant transmettre des données, à écouter le canal. Si le canal est libre alors la station peut transmettre. Sinon, elle attend que le canal

redevienne libre. La station doit pouvoir détecter d'éventuelles collisions. Elle avortera dans ce cas la transmission et tentera de réémettre ultérieurement.

L'utilisation de cette méthode s'avère très coûteuse pour des réseaux sans fils. En effet, pour pouvoir implémenter la méthode CSMA/CD on doit disposer d'un circuit *full duplex* pour la détection de collision.

Ainsi, la méthode CSMA/CA a été retenue pour les WLANs puisque le canal varie au cours du temps. Cette méthode abandonne la détection de collisions, tout en renforçant les mécanismes pour les éviter. Dans un environnement radio-mobilité, ce n'est pas possible d'appliquer le CSMA/CD.

### 2.1.2 Description générale du mécanisme DCF

Avant chaque émission, la station désirant émettre écoute le support. S'il est libre pendant une certaine durée DIFS (voir paragraphe 2.1.3), la transmission est possible. Si le support est occupé, une procédure de *Backoff* est enclenchée.

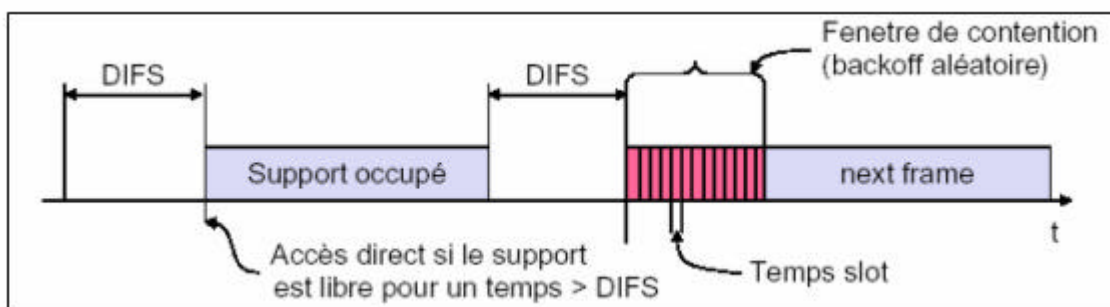


Figure 2. 1 L'accès au médium en mode DCF

Une station ayant correctement reçu un paquet, renvoie un accusé de réception (ACK) à la station émettrice. L'ACK indique à l'émetteur qu'aucune collision n'a eu lieu.

Par contre, si l'émetteur ne reçoit pas d'acquiescement au bout d'un certain temps, le fragment est retransmis jusqu'à réception d'un acquiescement par le récepteur.

Enfin, si après un nombre défini de retransmissions, aucun accusé de réception n'est reçu, l'émission est abandonnée.

### 2.1.3 Espaces inter-trames

Un espace inter-trames IFS (*Inter-Frame Space*) est la durée pendant laquelle une station doit attendre avant de transmettre sur le canal.

Pour définir les différentes sortes d'IFS, la norme a tout d'abord introduit la notion de Time Slot comme étant l'intervalle de temps qui permet à une station de savoir si une autre station a accédé au canal au début du slot précédent. La valeur d'un Time Slot dépend de la couche physique utilisée. Pour la couche PMD à étalement de spectre à séquence directe, cette valeur est 20  $\mu$ s.

A partir de la notion de Time Slot, la norme a ensuite introduit 4 types d'espaces inter trames, définis comme suit :

- SIFS (Short IFS) : utilisé pour séparer les transmissions d'un même «dialogue» : entre une trame émise et son acquittement par exemple, ou encore entre plusieurs fragments d'une même trame. Ainsi, si une station commence à émettre on ne peut pas la "couper", puisque le SIFS est le plus court des espaces inter trames. La valeur de SIFS retenue par la norme est de 10  $\mu$ s.
- PIFS (Point Coordination IFS) : utilisé par le point d'accès pour avoir une priorité d'accès au canal par rapport aux autres stations (PIFS < DIFS), ce qui permet à l'AP de basculer en mode d'accès avec contrôle centralisé. Sa valeur est calculée comme suit :

$$\text{PIFS} = \text{SIFS} + 1 \text{ Time Slot} = 30 \mu\text{s}.$$

- DIFS (Distributed IFS) est l'IFS utilisé par une station pour accéder au support de transmission. La valeur de DIFS est donnée par :

$$\text{DIFS} = \text{PIFS} + 1 \text{ Time Slot} = 50 \mu\text{s}.$$

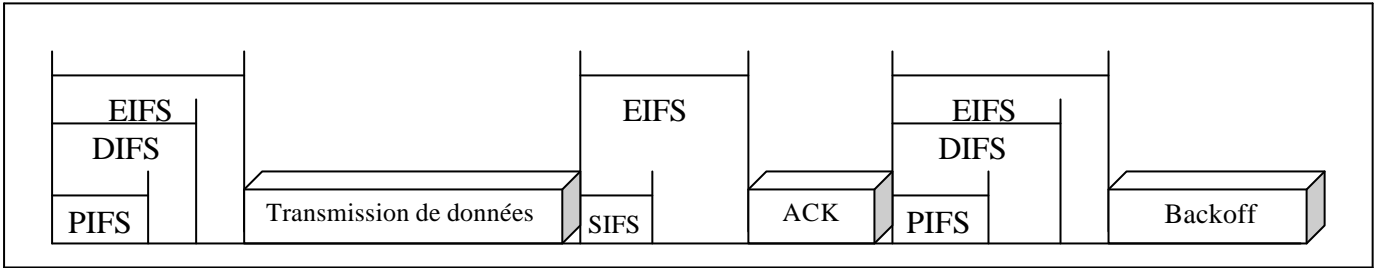
- EIFS (Extended IFS) c'est l'IFS le plus long, utilisé seulement en mode DCF. Il est utilisé dans le cas d'une trame erronée. La station réceptrice doit attendre pendant un EIFS l'acquittement de cette trame. Cela permet d'éviter les collisions. Dès la réception des données correctes pendant l'EIFS, celui-ci se termine et la station peut recommencer à transmettre des données. Sa valeur est donnée par la formule suivante :

$$\text{EIFS} = 8 * ?t_{\text{ACK}} + \text{SIFS} + \text{DIFS} + ?t_{\text{MAC}} + ?t_{\text{PLCP}}$$

Avec :

- ?  $t_{\text{ACK}}$  : Durée de l'émission d'un acquittement
- ?  $t_{\text{MAC}}$  : Durée d'une encapsulation de la couche MAC
- ?  $t_{\text{PLCP}}$  : Durée d'une encapsulation de la couche PLCP

La figure 2.2 illustre les relations entre les différents IFS.



**Figure 2. 2 Les relations entre différents IFS**

Le tableau suivant donne une idée sur les différentes valeurs des IFS en fonction de la valeur du Time Slots introduit par le standard 802.11 et la nature de la couche physique utilisée [1].

	<b>FHSS</b>	<b>DSSS</b>	<b>IR</b>
<b>Time Slot (µs)</b>	50	20	8
<b>SIFS (µs)</b>	28	10	7
<b>DIFS (µs)</b>	128	50	23
<b>PIFS (µs)</b>	78	30	15

**Tableau 2. 1 Valeurs des IFS et du time Slot en fonction de la couche physique**

### **2.1.4 L'algorithme de Backoff**

La procédure de Backoff est un mécanisme simple, basé sur le calcul d'un temporisateur gérant les transmissions et les retransmissions. Il permet de réduire la probabilité de collision sur le canal en essayant de minimiser les chances d'avoir plusieurs stations qui accèdent au support en même temps [2]

#### **2.1.4.1 Déroulement :**

Une station S désirant envoyer des données attend pendant une période DIFS. Si après cette durée le canal est libre, la station accède directement au canal. Dans le cas contraire, la station déclenche le mécanisme de Backoff qui se déroule en 3 étapes :

- La station calcule son temporisateur Backoff\_Timer :

$$\text{Backoff\_Timer} \leftarrow \text{Random} ( ) \times \text{TS}$$

Avec

- Random ( ) : nombre pseudo-aléatoire choisi entre 0 et CW-1 ; où CW est la taille de la fenêtre de contention.
- TS : durée d'un time-slot définie comme étant l'intervalle de temps nécessaire pour une station pour savoir si une autre a accédé au canal au début du time-slot précédant.

- Quand le canal devient libre, et après un DIFS, la station commence à décrémenter son temporisateur time-slot par time-slot.
- Lorsque la valeur de Backoff\_Timer est égale à 0, la station peut alors envoyer. Si par contre au cours de la phase de décrémentation, une autre station S' termine de décrémenter son temporisateur, la station S bloque son temporisateur. Elle pourra continuer de le décrémenter une fois la transmission de la station S' est finie.

### 2.1.4.2 Exemple de fonctionnement :

La figure 2.3 présente un exemple tiré de [1] où plusieurs stations désirent transmettre des données, utilisant la procédure de Backoff : Pendant que la station A transmet sur le support, les stations ayant des données à transmettre (B, C et D) diffèrent leurs transmissions.

Une fois que A eut terminé de transmettre, B, C et D attendent pendant une durée DIFS pour ensuite commencer à décrémenter leurs temporisateurs de Backoff.

Étant la première à terminer de décrémenter son *Backoff\_Timer*, la station C peut transmettre sur le support. Les stations B et D bloquent alors leurs temporisateurs respectifs.

Lorsque C termine de transmettre et après avoir attendu pendant un DIFS, les stations B et D reprennent la décrémentation de leurs *Backoff\_Timer*, là où ils les ont bloqués. Entre temps, une nouvelle station E désirent transmettre active à son tour une procédure de Backoff.

Les mécanismes se répètent jusqu'à ce que toutes les stations aient accédé une à une au support.

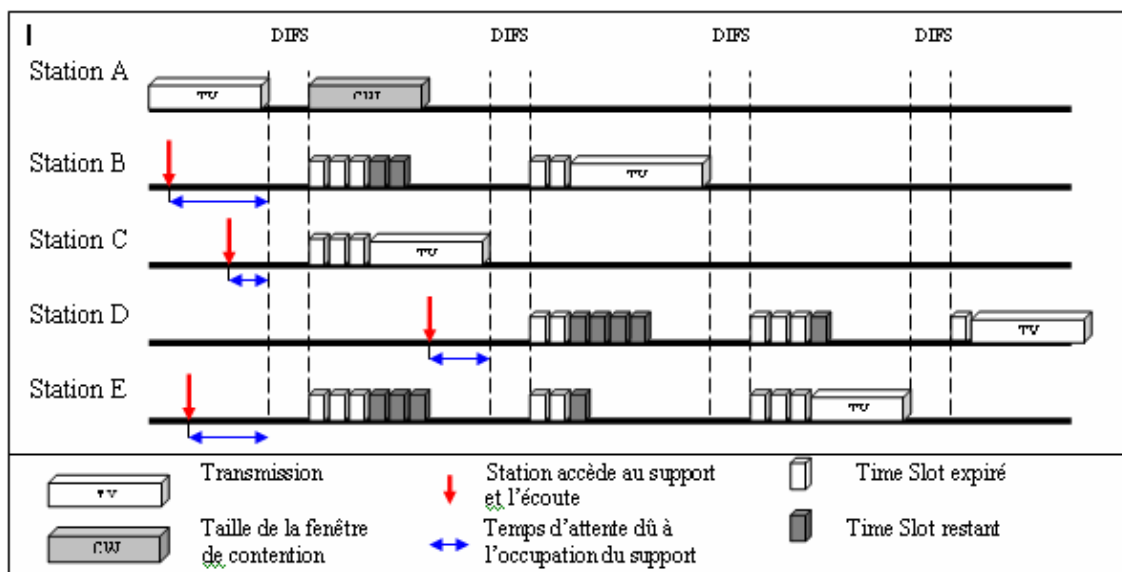


Figure 2. 3 Procédure de Backoff

2.1.4.3 Diagramme de fonctionnement :

La figure ci dessous résume le fonctionnement de la procédure CSMA/CA et de l'algorithme de Backoff [3].

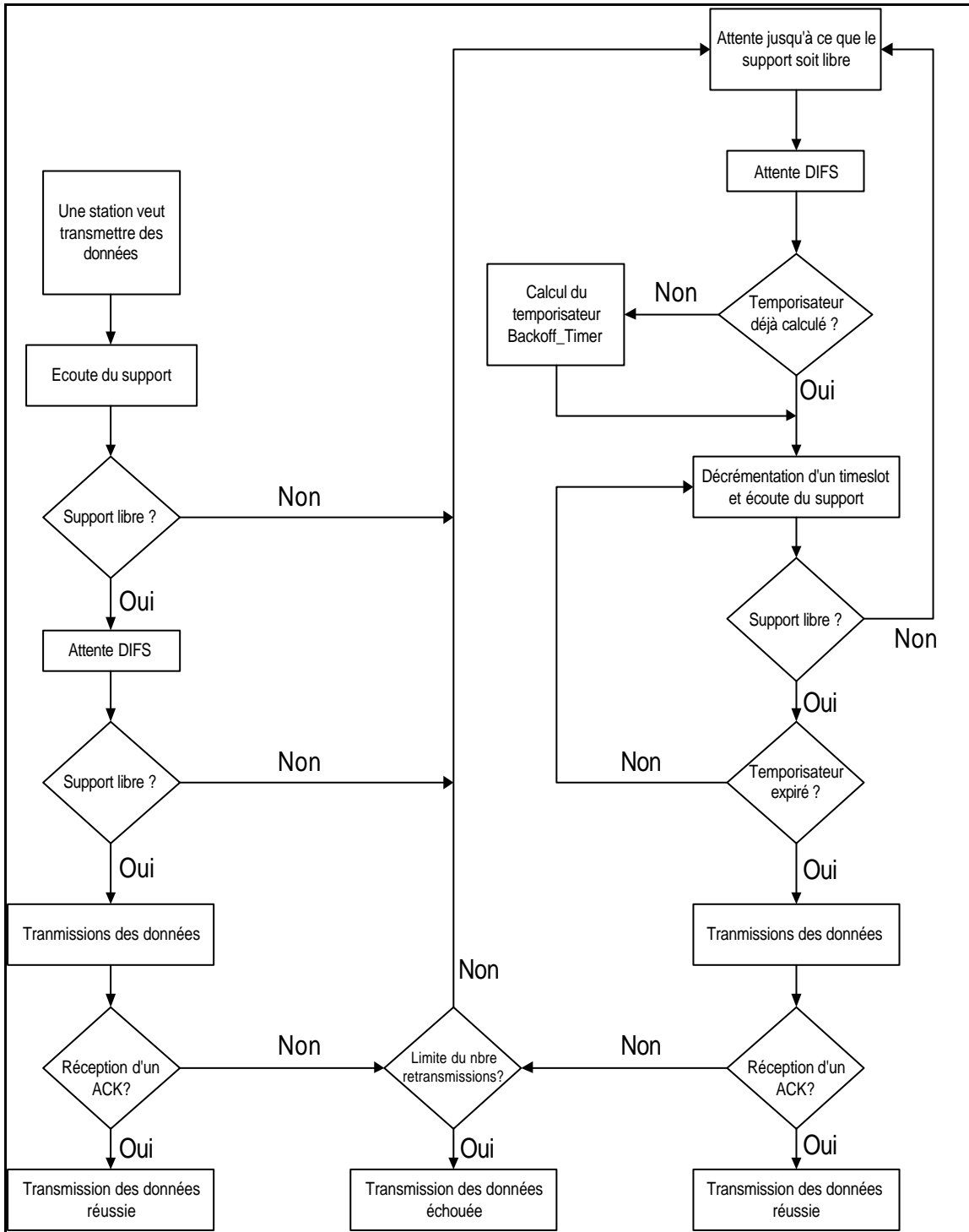


Figure 2. 4 La procédure CSMA/CA et Algorithme de Backoff

### 2.1.5 Mécanisme de détection virtuelle

Le mécanisme de détection virtuelle de la porteuse (*VCS : Virtual Carrier Sense*) est un procédé qui réduit les chances de collisions entre deux stations qui ne sont pas en visibilité. Ce mécanisme peut être introduit uniquement dans le cas de l'accès avec contrôle distribué (DCF)[3].

Chaque station tient à jour un compteur NAV (Network Allocation Vector) contenant la durée estimée des émissions et permettant ainsi de prévoir l'état d'occupation du support physique. Lorsqu'une station souhaite émettre, elle transmet tout d'abord une trame RTS (Request To Send) indiquant la source, la destination et la durée de la communication. La station réceptrice répond avec une trame CTS (Clear To Send) incluant ces mêmes informations. Détectant un RTS ou un CTS, les autres stations mettent à jour leur NAV.

Chaque station décrémente alors le NAV jusqu'à  $NAV = 0$  (événement correspondant à la libération « virtuelle » du canal). C'est à ce moment qu'on peut lancer la procédure d'accès.

La figure 2.5 montre les différentes étapes du mécanisme. Nous remarquons que les envois des paquets RTS, CTS et des paquets de données sont espacés par des durées SIFS.

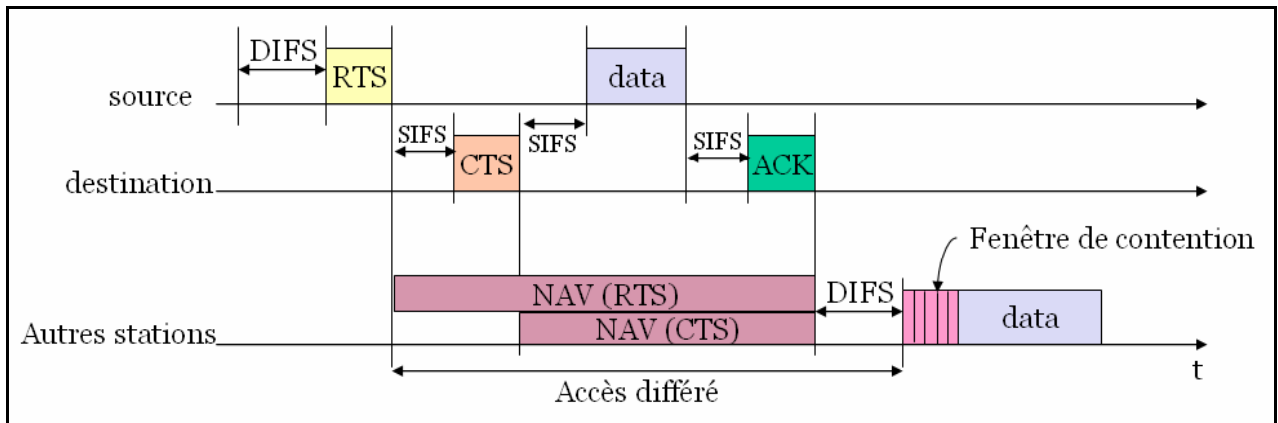


Figure 2. 5 Mécanisme d'écoute virtuelle VSC

Bien entendu, l'envoi des trames RTS/CTS n'élimine pas complètement le risque d'avoir des collisions. Cependant, s'il y a collision, elle se produit entre des trames de petite taille, ce qui améliore la qualité des transmissions. Dans ce sens, le mécanisme VCS cesse d'être intéressant lorsque les trames de données à envoyer sont de petite taille. Un seuil appelé *RTSThreshold* a donc été défini et le mécanisme VCS ne sera adopté que si la taille des trames à émettre est supérieure à ce seuil.

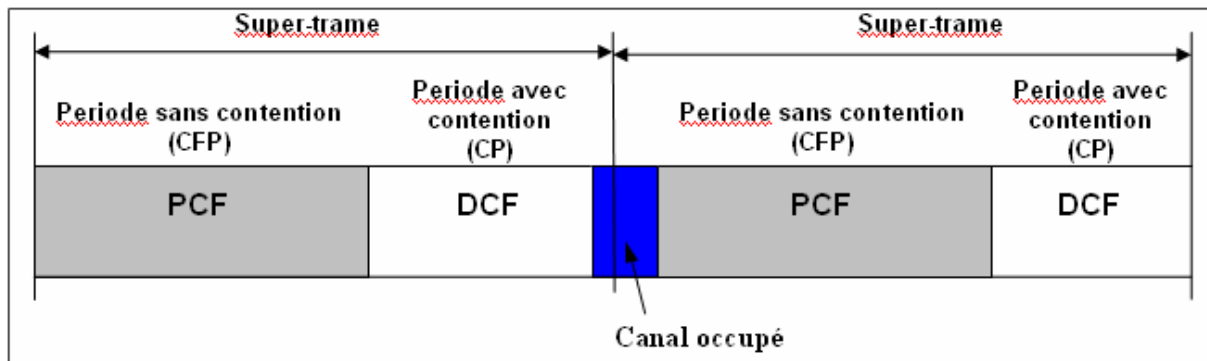
## 2.2 Le protocole PCF (Point Coordination Function)

Il s'agit d'une méthode basée sur un algorithme centralisé permettant de transmettre des données synchrones. Il s'agit d'une méthode optionnelle ne pouvant être utilisée qu'en mode

infrastructure. Le point d'accès prend le contrôle et choisit les stations qui peuvent transmettre leurs données. Pour cela, il définit un PC (Point coordinator) qui lui permet de communiquer avec station du BSS. Le PC est chargé donc d'ordonner les transmissions et de distribuer le droit d'utilisation du support. Le PC détermine deux périodes de temps :

- La CP (Contention Period) : c'est une période de temps avec contention durant laquelle la méthode d'accès DCF est utilisée.
- La CFP (Contention Free Period) : c'est une période de temps sans contention durant laquelle la méthode d'accès utilisée est PCF [ ].

Pour cette technique d'accès, le temps est divisé en intervalle de répétition du CFP, appelé aussi super-trame PCF. La figure ci-dessous illustre la succession de deux super-trames PCF :



**Figure 2. 6 Succession de deux super-trames PCF**

Au début de la période sans contention, le PC acquiert le contrôle du support et garde ce contrôle pendant toute cette période. Si le support est libre pendant une durée PIFS, le PC commence par envoyer une trame Beacon pour informer les stations de la durée de cette période ce qui leur permet de mettre à jour leur compteur NAV.

Ensuite, le PC attend un SIFS et après il commence à envoyer les données aux stations destinataires par l'intermédiaire des trames CF-Down tandis que les stations émettrices utilisent les trames CF-Up. Les différentes trames de données transmises sont espacées par des intervalles SIFS. La période CFP se termine par l'émission d'une trame CF-End. La figure 2.7 illustre le déroulement de la transmission pour les deux périodes CFP et CP.

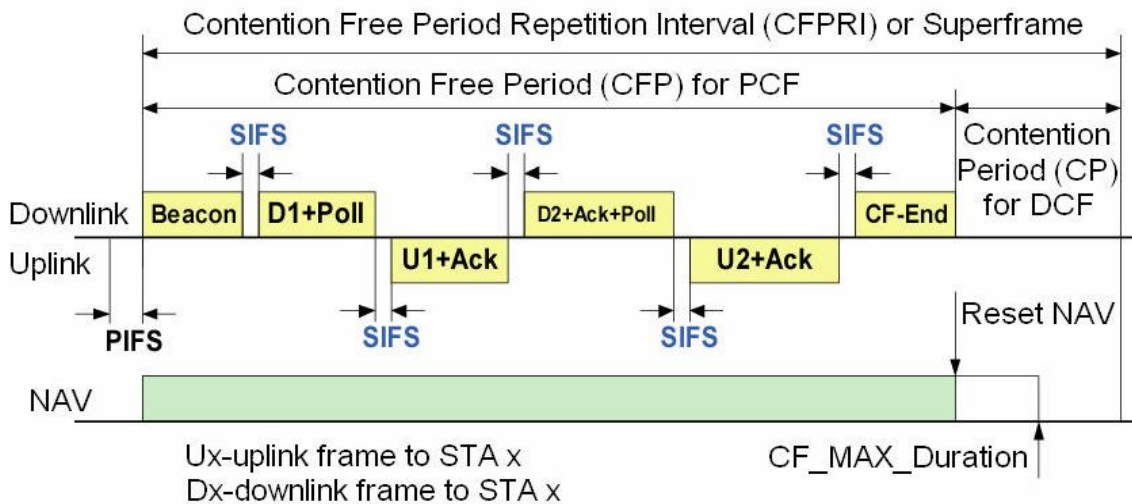


Figure 2. 7 Mécanisme de transmission pour les deux périodes CFP et CP

## 2.3 Limites en terme de qualité de service des mécanismes DCF-PCF et du standard 802.11e

### 2.3.1 Limites de DCF (Distributed Coordination Function)

L'inconvénient de la technique DCF est qu'elle ne fournit pas de garantie de qualité de service. Les différentes stations du BSS et les différentes classes de trafic ont la même probabilité d'accès au support. Cette technique ne comporte donc aucun mécanisme de différenciation qui garanti le débit et le délai pour les trafics de hautes priorités puisque  $CW_{MIN}$  et  $CW_{MAX}$  par exemple dépendent des caractéristiques de la couche physique. Finalement, il est important de signaler que cette technique supporte uniquement les services de type Best-effort

### 2.3.2 Limites de PCF (Point Coordination Function)

Le mode PCF a été conçu par IEEE pour supporter principalement des applications multimédia temps réel. Ceci étant, trois facteurs réduisent ces performances en terme de QoS :

- La complexité du plan de polling a pour conséquence de détériorer les performances en terme de qualité de service pour les trafics de haute priorité.
- L'alternance entre les périodes CP et CFP ne se déroule pas toujours sans problèmes : les transmissions en mode DCF peuvent mettre du temps à se terminer ce qui peut retarder l'envoi de la trame Beacon pour passer en mode PCF. Ce retard provoque

des délais supplémentaires sur les différentes transmissions en mode PCF, et détériore la qualité de service.

- Le PC n'est pas en mesure d'estimer avec exactitude la durée des transmissions des stations «élues» ; cette durée dépend en effet des différentes méthodes de codage de la couche physique. La fragmentation des trames influe également sur cette durée.

### 2.3.3 La norme 802.11e

Afin de gérer la priorité d'accès au support et garantir la qualité de service pour les trafics multimédia, le groupe IEEE 802.11 a développé de nouveaux mécanismes dans le but de garantir une certaine qualité de service.

Ce standard repose sur deux mécanismes d'accès : EDCF (Enhanced DCF) qui fonctionne durant la période CP et HCF (Hybrid Coordination Function) qui fonctionne durant les deux périodes. La station qui utilise le standard 802.11e est appelée ESTA (Enhanced STation) et la station jouant le rôle de contrôleur d'accès est appelée HC (Hybrid Coordinator).

#### 2.3.3.1 EDCF

L'EDCF est une évolution du DCF, qui ajoute un système de gestion de priorités lors de l'accès au support. L'accès se fait selon le niveau de priorité de la trame, toujours en faisant intervenir les espaces inter-trames.

L'EDCF fournit des accès différenciés pour différents types de trafic. Il définit huit niveaux de priorités (entre 0 et 7) par l'intermédiaire des catégories de trafic, ou TC (*Traffic Category*). Les espaces inter-trames ne sont plus identiques pour toutes les trames de données. EDCF introduit en effet le AIFS (*Arbitration IFS*) qui est en fait fonction de la classe de trafic : à chaque catégorie de trafic (TC) correspond un AIFS donné, de plus en plus petit à mesure que le degré de priorité est grand. La valeur minimale de AIFS correspond au DIFS.

De la même manière, les tailles limites de la fenêtre de contention  $CW_{min}$  et  $CW_{max}$  diffèrent selon la catégorie de trafic.

Chacune des catégories de trafic TC correspond à une file d'attente FIFO où l'accès des paquets au canal se fait en mode DCF, avec AIFS [TC] au lieu de l'habituel DIFS, et ( $CW_{min}[TC]$ ,  $CW_{max}[TC]$ ) au lieu des ( $CW_{min}$ ,  $CW_{max}$ ) identiques pour tous les types de trafic dans le DCF classique. (le mécanisme sera détaillé dans le chapitre 3)

### 2.3.3.2 HCF

Il s'agit d'un mécanisme centralisé qui fonctionne durant les périodes CFP et CP. Ceci permet à HCF de ne pas attendre la période CFP pour interroger les stations et leurs donner la permission de transmettre. De plus, la station HC peut accéder au support après un temps SIFS qui est inférieur à PIFS (utilisé dans PCF). En conséquence, la priorité d'accès associée au coordinateur est plus importante que celle des autres stations, quel que soit le type des trames qu'il souhaite émettre (dans PCF, une station désirant émettre une trame de type ACK est plus prioritaire que le PC). Les transmissions avec HCF peuvent se résumer de la manière suivante : la station HC interroge les stations en fonction des paramètres liés à la qualité de service; celles-ci répondent avec ou sans les données qu'elles souhaitent envoyer. Deux scénarios de transmission sont alors possibles : soit le HC transmet les données reçues à leurs destinataires dans un ordre qu'il définit ; soit il distribue des transmissions opportunity (Intervalle de temps pendant lequel une station a le droit d'émettre) TXOP aux ESTA en fonction de leurs priorités, afin qu'elles puissent transmettre les données elles-mêmes [2]. La figure 2.11 illustre la structure d'une super-trame HCF :

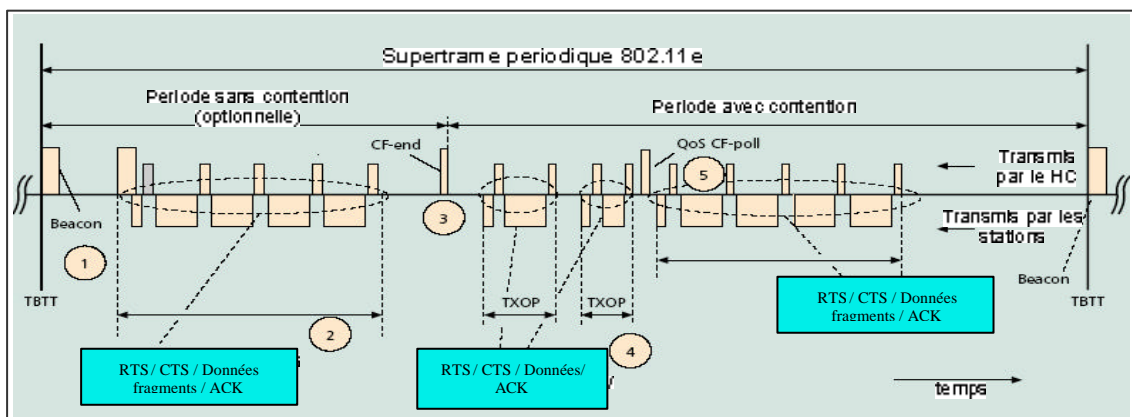


Figure 2. 8 Structure d'une super-trame HCF

### CONCLUSION

Au niveau de ce chapitre, nous avons détaillé les modes d'accès au support dans les réseaux sans fils IEEE 802.11 DCF et PCF.

Ces deux modes ont montré certaines limites tel que l'absence de mécanismes de différenciation de services ce qui a mené à la naissance du nouveau standard 802.11e. Nous avons présenté ce standard ainsi que les deux mécanismes d'accès EDCF et HCF réservés pour cette norme. Ces mécanismes assurent une certaine qualité de service pour les réseaux locaux sans fils WLANs.

## Chapitre 3 : Etude des mécanismes de différenciation de services

---

IP, malgré ses atouts, possède des limites en matière de qualité de service (Qos). Il traite en effet les paquets sans aucun mécanisme de différenciation de service. Les flux temps réel sont traités comme les flux de messagerie en Best effort delivery. Ce protocole privilégie la simplicité au détriment de la fiabilité. Le temps de transmission peut donc être long et surtout inconstant d'où un phénomène de gigue non négligeable. Depuis 1989, de nombreuses propositions ont été faites par l'IETF afin de pallier ces inconvénients. Le développement du multimédia, l'exigence croissante des applications sensibles, la voix sur IP ont précipité ces dernières années le développement de protocoles spécifiques correspondant aux couches 4 à 7 du modèle OSI permettant d'avoir une qualité de service adaptée à la spécificité des flux de données.

L'IETF propose désormais l'utilisation de deux grandes catégories de services pour assurer la qualité de service (Qos) : les services intégrés (IntServ) et les services différenciés (DiffServ).

IntServ gère les services de manière indépendante alors que DiffServ rassemble plusieurs applications simultanément. Intserv est souvent utilisé pour le réseau d'accès, les réseaux de faible taille pour des flux de données connus et prévisibles. DiffServ est un modèle extensible ciblant les réseaux de grande taille où il y a beaucoup de flots à gérer. Dans ce chapitre, nous commençons par détailler le mode EDCF dont nous évaluerons les performances dans le chapitre suivant. Ensuite, nous décrivons les mécanismes de différenciation de service au niveau IP : à savoir IntServ et DiffServ. En dernière phase, nous proposerons une architecture de qualité de service, associant DiffServ à l'EDCF dont l'évaluation des performances sera présentée dans le chapitre suivant.

### 3.1 Le mécanisme de différenciation de service niveau MAC : EDCF (Enhanced Distributed Coordination Function)

Il s'agit d'une amélioration du DCF (Distributed Coordination Function) qui ajoute un système de priorité pour la gestion de l'accès au support. Ce dernier se fait alors selon le niveau de priorité de la trame. Selon la définition du dernier draft de la norme 802.11e [7], la couche MAC au niveau d'une station est formée de quatre files de transmission dont chacune fonctionne comme une entité de Backoff en mode DCF. La structure de cette couche est illustrée par la figure 3.1.

La norme IEEE 802.11e a donc défini, au niveau MAC, quatre catégories d'accès : AC (Access Catégorie) relatives aux applications traitées dans les couches supérieures. Chaque catégorie de trafic constitue une file d'attente FIFO. Elles sont notées respectivement :

- **AC\_VO** : pour les applications temps réels tel que la voix
- **AC\_VI** : pour les applications vidéo
- **AC\_BE** : pour le trafic " Best Effort "
- **AC\_BK** : pour le trafic Background

Pour introduire la notion de différenciation entre les différentes AC, Chaque catégorie de trafic possède son propre DIFS, on parle donc de AIFS (Arbitration IFS). Ces catégories de trafic, gèrent huit niveaux de priorités allant de 0 à 7 relatives à la norme 802.11D. Les correspondances entre ces priorités et les catégories d'accès sont récapitulées aussi au niveau de la figure 3.1. En outre, il est important de signaler que les tailles limites de la fenêtre de contention diffèrent selon la classe de trafic. On parle alors de  $CW_{Min} [AC]$  et  $CW_{Max} [AC]$ . Chaque AC détient son propre compteur de Back-off qui est désormais compris entre 1 et  $1 + CW [AC]$ .

Quand deux ACs finissent en même temps leur durée de Backoff, alors c'est le paquet de plus haute priorité qui sera transmis, les autres entités doivent augmenter leurs fenêtres de Backoff.



Pour sa version actuelle, la norme 802.11e a aussi introduit le paramètre TXOP (Transmission Opportunity). Il s'agit d'un intervalle de temps pendant lequel une station a le droit d'émettre. Au niveau de la trame balise, l'AP annonce aussi à chaque AC la limite de l'intervalle TXOP (TXOPLimit [AC]) tout en définissant aussi la date de début de transmission. Durant un TXOP, la station peut transmettre plusieurs MPDUs pour un seul AC. Ces MPDUs sont espacés d'un SIFS de leurs acquittements. Cette transmission de plusieurs MPDUs est notée CFB (Contention Free Burst). La figure 3.3 présente la structure du CFB :

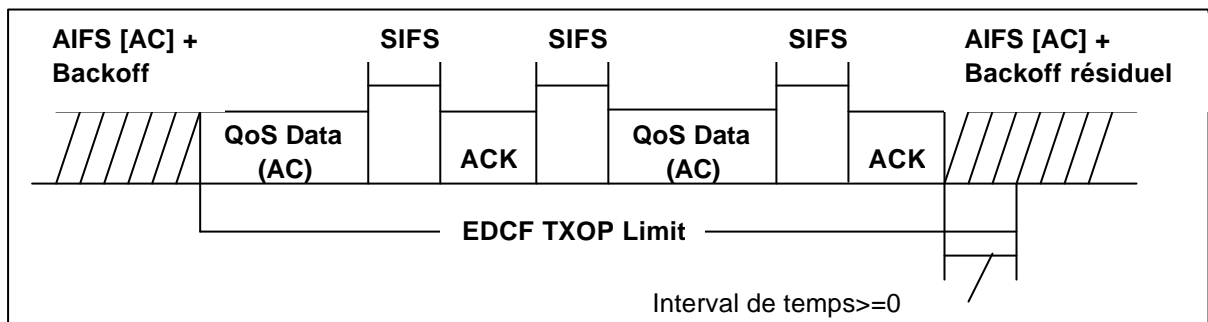


Figure 3. 3 Structure temporelle du CFB

## 3.2 Les mécanismes de différenciation de service niveau IP : IntServ / DiffServ

### 3.2.1 Le protocole à intégration de service IntServ

Les applications traditionnels non temps réels comme FTP se sont longtemps satisfaites du service best effort. Mais avec l'arrivée des communications multimédias, de nombreuses applications sont devenues sensibles au délai si bien que le service best effort traditionnel ne suffit plus. Bien que certaines applications soient adaptatives, il est souvent nécessaire de fournir de nouvelles classes de service offrant une meilleure qualité de service (en terme de bande passante, délai ou pertes). Ces nouvelles classes de service s'ajoutent au best effort traditionnel pour créer un Internet à intégration de services.

#### 3.2.1.1 Service proposés par IntServ

Un mécanisme explicite est utilisé pour signaler les exigences de qualité de service par flot aux éléments du réseau (hôtes, routeurs ou sous-réseaux). Les éléments du réseau, selon les ressources disponibles, implémentent l'un des services Intserv en fonction du type de qualité de service souhaité pendant la transmission des données. Le modèle distingue plusieurs types de services, en fonction du délai de transit par paquet

##### 3.2.1.1.1 Service à contrôle de charge (CL)

Le service à contrôle de charge (Controlled-Load) est destiné aux applications temps réel adaptatives, qui sont très sensibles à la congestion dans le réseau. Le service à contrôle de charge offre une seule fonctionnalité à ces applications : En effet, il fournit un débit comparable à ce qu'on aurait dans un environnement de réseaux peu chargés (sans congestion). Le contrôle de charge n'accepte pas de paramètres de qualité de service spécifiques comme paramètres de contrôle, tels que la perte de paquets ou le délai.

### 3.2.1.1.2 Service garanti (GS)

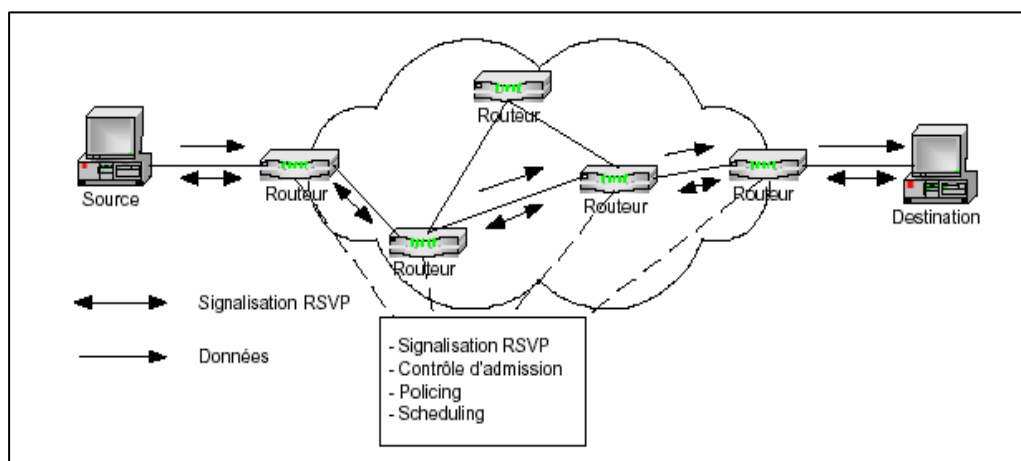
Le service garanti (Guaranteed Service) est un service quantitatif qui fournit des garanties de bande passante et des bornes de délai. Il est destiné aux applications temps réel, ayant des exigences de qualité de service très strictes. Le service GS contrôle uniquement le délai maximum ; il ne contrôle pas le délai minimum ni la gigue. Le délai est constitué du délai fixe et du délai d'attente. Le délai fixe est une propriété du chemin, il est déterminé lors de l'établissement du chemin par des mécanismes tels que RSVP, alors que le délai d'attente est déterminé par le service GS.

### 3.2.1.2 Architecture de IntServ

L'architecture Intserv repose sur deux principes fondamentaux :

- le réseau doit être contrôlé et soumis à des mécanismes de contrôle d'admission,
- des mécanismes de réservation de ressources sont nécessaires pour fournir des services différenciés.

Le modèle IntServ définit une architecture capable de prendre en charge la qualité de service en définissant des mécanismes de contrôle complémentaires sans toucher au fonctionnement IP. C'est un modèle basé sur un protocole de signalisation RSVP. Dans le modèle présenté par la figure 3.4, les routeurs réservent les ressources pour un flot de données spécifiques en mémorisant des informations d'état. Il est important de rafraîchir périodiquement les informations au cas où il y a eu changement de la route emprunté par le flot. En effet, il est inutile de continuer à réserver les ressources sur un routeur qui ne fait plus partie du chemin emprunté.



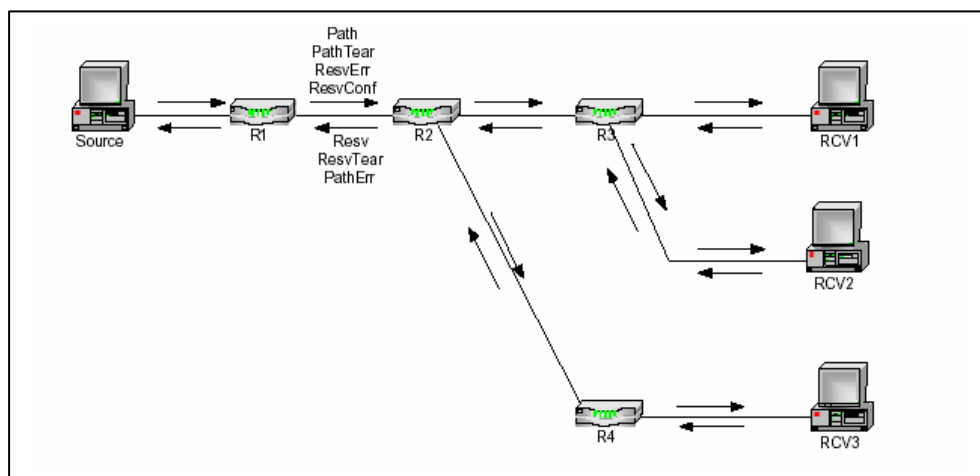
**Figure 3. 4 Architecture de IntServ**

### 3.2.1.3 Le protocole RSVP (Resource Reservation Setup Protocol)

Le protocole de signalisation RSVP est un mécanisme dynamique conçu pour effectuer des réservations de ressources explicites dans une architecture Intserv. RSVP est utilisé uniquement pour la communication des paramètres de qualité de service, il ne comprend pas l'information qu'il transporte dans les requêtes de qualité de service.

- RSVP est initié par une application au début d'une session de communication. Une session est identifiée par l'adresse IP de destination, le type de protocole de la couche transport et le numéro de port de destination. Chaque paquet RSVP contient les détails sur la session à laquelle il appartient. L'affectation des ressources demandées par l'intermédiaire de RSVP pour un flot donné est indépendante de RSVP, elle dépend d'IntServ.
- RSVP établit et maintient un état logiciel entre les nœuds constituant le chemin réservé. Par opposition à la réservation d'un chemin statique, cet état logiciel est caractérisé par des messages de rafraîchissement envoyés périodiquement le long du chemin pour maintenir l'état.
- RSVP fournit aussi une qualité de service dynamique tenant compte des modifications de ressources ; celles-ci peuvent être dues au destinataire, à l'émetteur ou encore à de nouveaux membres dans un groupe multicast.

Le protocole RSVP définit sept types de messages, les principaux étant les messages PATH et RESV. Ces deux messages assurent le fonctionnement de base de RSVP. Les autres messages RSVP sont utilisés soit pour fournir des informations sur l'état des réservations, soit pour annuler explicitement les réservations le long d'un chemin d'une session de communication.



**Figure 3. 5 Sens des messages RSVP**

### 3.2.1.4 Les Limites de IntServ :

Au niveau technique, la faiblesse principale de l'architecture IntServ est sa non-résistance au facteur d'échelle. Le nombre de flux qui peuvent bénéficier d'une réservation est assez limité, en particulier dans les routeurs du coeur du réseau. Ces équipements doivent traiter des milliers des flux simultanément [12], et le coût introduit par la gestion d'états et l'ordonnancement par flux peut entraîner une réduction considérable de leur performance.

Au niveau économique, la réservation de ressources est incompatible avec le système de tarification forfaitaire habituellement utilisé dans l'Internet. Si aucun contrôle n'est effectué, un seul utilisateur pourrait réserver la totalité de la bande passante au détriment des connexions établies par d'autres utilisateurs. Un nouveau système de tarification, basé sur la quantité de ressources réservées, devrait être implanté dans le réseau. Avec l'introduction du protocole COPS [13], une telle action est imaginable à l'intérieur d'un domaine. Par contre, pour que le modèle puisse être implanté à grande échelle, l'échange des informations concernant la tarification doit être sécurisé.

Malgré l'existence des standards qui définissent les modifications nécessaires pour faire de l'Internet un réseau sécurisé [12], leur mise en oeuvre n'est pas encore au niveau nécessaire pour pouvoir transporter des informations si importantes que celles concernant la tarification. La standardisation du modèle IntServ s'est pratiquement achevée en 1996. Pourtant, le modèle n'a jamais été implanté à grande échelle dans le réseau. Les faiblesses énumérées précédemment empêchent son déploiement. De nouvelles propositions cherchant à offrir des garanties strictes dans le réseau sont actuellement à l'étude. Elles se basent sur une architecture hybride IntServ-DiffServ [12].

### 3.2.2 Le protocole à Différenciation de service DiffServ

L'architecture à Différenciation de Services (DiffServ) résulte des efforts menés pour résoudre les problèmes de complexité et de passage à l'échelle posés par IntServ. Le passage à l'échelle devient possible en offrant des services à des agrégats plutôt qu'à chaque flot et en repoussant le traitement par flot aux extrémités du réseau. L'objectif est de laisser le cœur du réseau aussi simple que possible. [9]

L'avantage de DiffServ est qu'il n'y a plus nécessité de maintenir un état des sources et des destinations dans les routeurs de cœur du réseau, d'où une meilleure *scalability*

#### 3.2.2.1 Principe de l'architecture à Différenciation de Services

La différenciation de services est principalement réalisée grâce au champ Differentiated Service (DS ou DSCP : Differentiated Services Code Point) dans l'en-tête IP et au comportement associé (Per-Hop Behavior ou PHB). [9]

DSCP est le champ qui identifie le traitement que le paquet doit recevoir. Ce champ est codé sur 6 bits et fait parti des 8 bits codant le champ TOS d'IPv4 ou le champ COS d'IPv6, laissant une marge de codage pour les classes de service d'historique et d'autres qui viendront dans le futur.

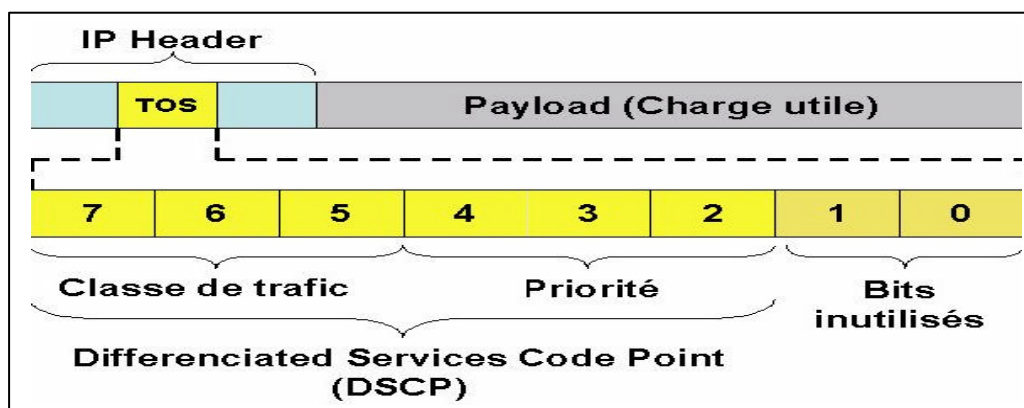
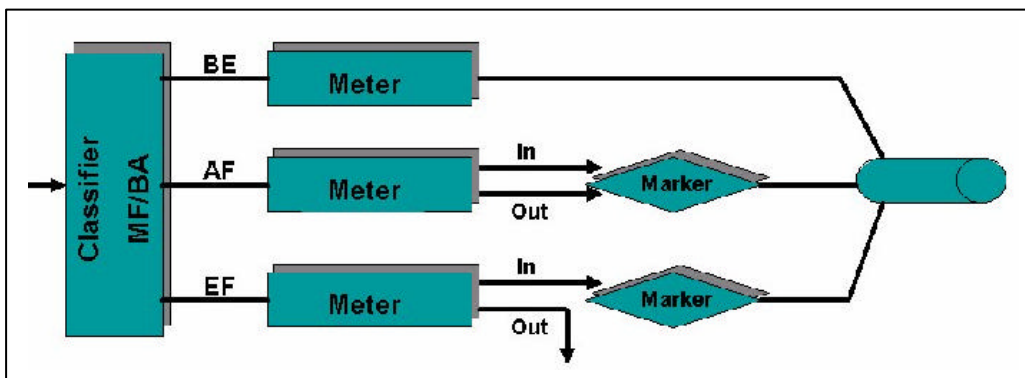


Figure 3. 6 Entête d'un datagramme IPV4

DiffServ divise le réseau en domaines. Un domaine est un groupe de nœuds qui fonctionnent avec un ensemble commun de politiques d'allocation de service et de définitions de PHB. Un domaine DiffServ est constitué de deux types d'éléments fonctionnels, les éléments de bordure et les éléments du cœur du réseau [10]

- Les éléments de bordure sont responsables de la classification des paquets et du conditionnement du trafic en fonction des accords de service (Service Level Agreement ou SLA) entre les domaines voisins. Un SLA est un accord bilatéral entre des domaines, négocié statiquement ou dynamiquement. Selon les SLAs, des spécifications de service (Service Level Specification ou SLS) appropriées sont affectées aux nœuds de bordure. Le SLS contient des paramètres tels que la capacité de transmission, la taille des rafales et le débit crête. Les nœuds de bordure contiennent les éléments suivants (voir figure 3.8) [10]:
  - un métreur qui mesure le trafic afin de vérifier sa conformité par rapport au profil établi,
  - un marqueur qui positionne le champ DS à une valeur déterminée (le champ DS identifie l'agrégat auquel le trafic appartient),
  - un régulateur, pour retarder le trafic afin qu'il n'excède pas le profil,



**Figure 3. 7 Traitement d'un paquet par un edge router**

- Les éléments du cœur du réseau ne sont responsables que du transit des paquets. À chaque nœud, les paquets sont traités selon le PHB invoqué par l'octet DS dans l'en-tête du paquet. Les routeurs de cœur ne voient plus des flots utilisateurs mais des classes (voir figure 3.9). Les routeurs appliquent un traitement aux paquets en fonction de leur champ DS. Ce dernier décrit un PHB traduit par exemple par :
  - le niveau de priorité du paquet si le routeur gère les paquets suivant un mécanisme de priorités (avec une file d'attente par niveau de priorité);
  - la proportion minimale de la capacité du routeur associée à chaque classe de paquets si la file est gérée suivant une discipline WFQ (*Weighted Fair Queueing*)

- la probabilité de rejet du paquet en fonction de sa classe (et du niveau d'occupation de la file) si le routeur gère le trafic suivant une politique RED (*Random Early Discard*)

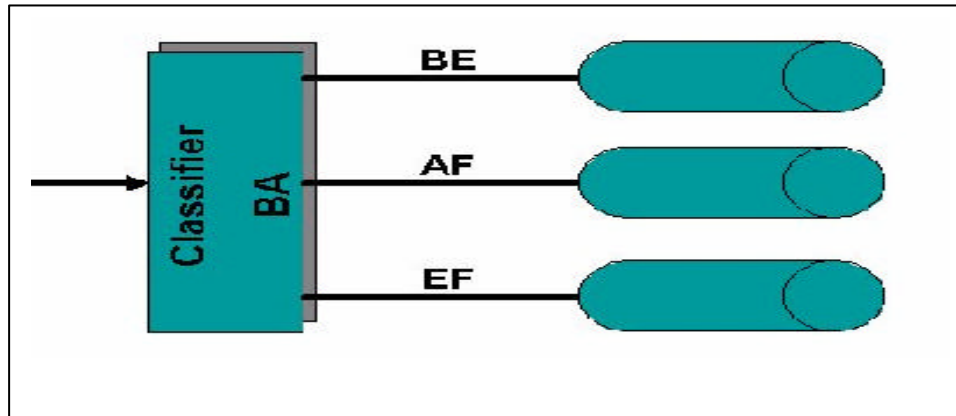


Figure 3. 8 Traitement d'un paquet par un core router

### 3.2.2.2 Classes de services :

#### 3.2.2.2.1 Best Effort

Le principe du Best Effort se traduit par une simplification à l'extrême des équipements d'interconnexion. Quand la mémoire d'un routeur est saturée, les paquets sont rejetés. Le principe de bout en bout de l'Internet est aussi adopté pour le contrôle de flux grâce à différents algorithmes comme le *Congestion Avoidance* introduit dans TCP. Les principaux inconvénients de cette politique de contrôle de flux sont un trafic en dents de scie composé de phases où le débit augmente puis est réduit brutalement et une absence de garantie à long terme.

#### 3.2.2.2.2 Expedited Forwarding

La classe « Expedited Forwarding » correspond à la valeur 101110 pour le DSCP et l'objectif est de fournir un service de transfert équivalent à une ligne virtuelle dédiée à travers le réseau d'un opérateur. Le contrat porte sur un débit constant. Les paquets excédentaires sont lissés ou rejetés à l'entrée pour toujours rester conforme au contrat. L'opérateur s'engage à traiter ce trafic prioritairement. Pour que le service soit performant, il faut qu'il ne présente qu'une faible partie du trafic total pour qu'aucun paquet marqué EF ne soit rejeté dans le cœur du réseau [11]

Pour atteindre ces performances, les paquets d'un service EF ne devraient pas subir de file d'attente ou passer par des files de très petite taille et strictement prioritaires. De plus, les flux

ne doivent avoir que très peu de perte, la gigue doit être minimale et la bande passante garantie.

D'une part, cela nécessite la mise en place d'un contrôle d'accès et, d'autre part, cela impose qu'à chaque noeud traversé, le taux maximal de trafic d'arrivée doit être inférieur au taux minimal de trafic de départ. Cette dernière assertion implique que, dans les noeuds internes, une bande passante minimale est disponible au service EF et que, dans les noeuds d'extrémité, un *traffic conditioning* est effectué.

Ce mécanisme de conditionnement est utilisé pour vérifier la conformité des flux utilisateurs. La conformité du trafic EF et AF par rapport à leurs profils est déterminée pour chacun par un *token bucket*). Dans ce cas, la taille et le débit du *bucket* sont à spécifier. Les paquets EF non conformes sont détruits tandis que les paquets AF non conformes sont marqués pour être jetés en cas de congestion.

#### 3.2.2.2.3 Assured Forwarding

Pour l'AF, il définit 4 classes de service et 3 priorités de rejets (appelées niveau de post-précédence) suivant que l'utilisateur respecte son contrat, le dépasse légèrement ou est largement en dehors. Les classes sont donc choisies par l'utilisateur et restent les mêmes tout au long du trajet dans le réseau.

Chaque classe peut être vue comme une file d'attente séparée en utilisant une certaine proportion des ressources du réseau.

Les étapes du traitement du paquet dans le cas de la transmission garantie (voir figure 3.10) :

- La première étape consiste à classer les paquets en fonction des 4 classes de priorités. Elle peut être réalisée sur l'hôte émetteur, ou sur le premier routeur d'accès
- La deuxième étape consiste à marquer les paquets en fonction de la priorité définie. Pour cela, on utilise le champ codé dans l'en-tête du paquet IP.
- La troisième étape consiste à faire passer les paquets à travers un filtre de canalisation/suppression qui peut retarder ou éliminer certains paquets pour donner aux 4 flux un comportement acceptable,

Avantages de l'*Assured Forwarding* : [11]

- Peut offrir une meilleure différenciation (classe et priorité)

- Le marquage à l'entrée du réseau est une opération moins coûteuse que le *shaping*
- Ne demande pas une coordination entre domaines
- Une facturation simple peut être utilisée.

Inconvénients de l'*Assured Forwarding* :

La qualité offerte dépend énormément du niveau d'agrégation et de la présence de flux concurrents

- Il n'existe aucune assurance de délai
- Il y a beaucoup de paramètres à régler
- 3 niveaux de priorité ne suffisent pas pour assurer une bonne différenciation sur des liens non-chargés
- Un mauvais dimensionnement rend inutile la présence de priorités sur des liens en congestion
- Le marquage ne suffit pas pour protéger TCP de UDP.

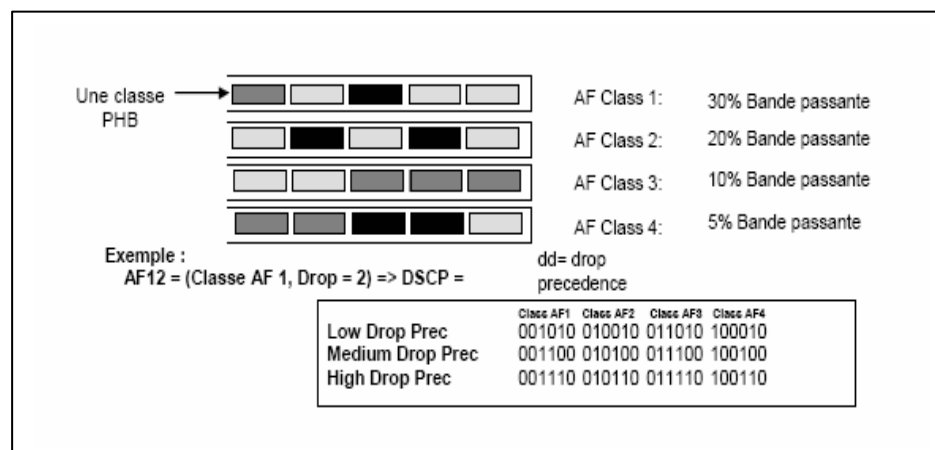


Figure 3. 9 Différenciation avec AF

### 3.2.2.2 Le traitement différencié des paquets :

Dans l'architecture DiffServ, le traitement différencié des paquets s'appuie sur plusieurs opérations fondamentales :

- Classification des flux en classes de services.
- Conditionnement du trafic.
- Gestion de la congestion : introduction des priorités au sein des classes (scheduling) et gestion du trafic dans une classe donnée (queue management).

### 3.2.2.2.1 Classification et conditionnement du trafic

La classification s'effectue suivant une ou plusieurs valeurs contenues dans l'en-tête IP (ex : @ source, @ dest, ...).

Une fois les paquets marqués, ils sont envoyés à leur destination puis à chaque routeur, ils reçoivent le service associé à leur classe.

En plus de cette classification/marquage, un mécanisme de profilage du trafic est mis en place. Il a pour objet la prise en compte du taux d'arrivée des paquets, afin de ne pas dépasser le seuil maximum de paquets pouvant être envoyés sur le réseau. Ainsi, un mécanisme de mesure du trafic permet de savoir si le flot de paquets entrant correspond au profil de trafic négocié. Si ce flot dépasse un certain seuil, certains paquets seront marqués comme moins prioritaires et seront automatiquement jetés en cas de congestion dans le réseau.

Le Conditionnement du trafic met donc en oeuvre 4 composants:

- Meter : mesure du trafic pour vérifier s'il est compatible avec le contrat et fourniture des infos aux autres composants
- Marker: affectation d'un DS qui peut être différent de celui qui est reçu.
- Shaper: lissage du trafic en retardant certains paquets de telle sorte qu'il respecte le débit contractuel
- Dropper: suppression de paquets dépassant le trafic contractuel.

### 3.2.2.2.2 Gestion de la congestion

Cette opération est assurée par les algorithmes d'ordonnancement servant à contrôler la distribution de ressources entre les classes de service. On peut donner en exemple 2 types d'ordonnanceurs : **PQ** (*Priority Queueing*) et **WRR** (*Weighted RoundRobin*).[13]

Le modèle PQ utilise plusieurs files d'attente logiques. Les paquets classifiés sont mis dans une file d'attente correspondant à la valeur du DSCP. Les files sont ensuite servies suivant un algorithme spécifique. Celle qui contient les paquets avec la plus haute priorité sera favorisée par rapport aux autres files.

Le modèle WRR utilise aussi plusieurs files d'attentes mais qui sont servies à tour de rôle. A chaque tour, on transmet un nombre de bits (ou de paquets) correspondant au poids de la file.

Lorsqu'il y a de la congestion sur le réseau, un mécanisme de priorité, tel que CBWFQ, est mis en place pour garantir la bande passante aux différentes classes de trafic.

### 3.2.2.3 Gestion des files d'attente

La stratégie de gestion des diverses files d'attente sur un routeur joue un rôle essentiel dans la différenciation des services, à travers le choix de l'algorithme qui place les paquets dans la queue de sortie, et le choix de la taille maximale de la queue. Plusieurs algorithmes ont été développés:

#### **First In, First out (FIFO):**

C'est la méthode standard de gestion de trafic entre une interface d'entrée et une interface de sortie. Les paquets sont placés dans la file de sortie dans l'ordre dans lequel ils sont reçus. Compte tenu des optimisations logicielles effectuées depuis le début, cette technologie peut être considérée comme la plus rapide du point de vue de la transmission en paquets par seconde alors que des techniques plus élaborées risquent de dégrader ces performances (11).

#### **Priority queuing (PQ)**

C'est la forme primitive de différenciation des services. Un trafic particulier peut être identifié et réordonné dans la file de sortie suivant un critère fourni par l'utilisateur dans la file de sortie.

#### **Class-Based Queuing (CBQ)**

Ce mécanisme, utilisé pour éviter qu'une seule classe de trafic ne monopolise les ressources, définit plusieurs files de sortie avec une priorité et un total de trafic autorisé. Le trafic est extrait de chaque queue suivant une rotation.

#### **Weighted Fair Queuing (WFQ)**

Cet algorithme donne un traitement prioritaire aux flux de faible volume et permet aux flux de volume important d'utiliser la place qui reste. Pour cela, il trie et regroupe les paquets par flux, puis met ceux-ci en file d'attente suivant le volume de trafic dans chaque flux.

### 3.3 Architecture de différenciation de service niveau IP et/ou MAC

#### 3.3.1 Position du problème :

Pour que l'Internet fournisse des services flexibles avec la garantie de paramètres de performance (délai, gigue, débit et taux de perte de paquets), il doit nécessairement d'une part intégrer une notion de priorité à l'aide des mécanismes d'ordonnancement, de gestion et de contrôle dans le réseau et d'autre part assurer une qualité de service globale dans tous les domaines et réseaux d'accès traversés. Un réseau backbone avec qualité de service (QoS – Quality of Service) peut s'appuyer sur l'architecture **DiffServ** (Differentiated Services) [17] qui offre une différenciation par priorité basée sur une classification des paquets à l'entrée du réseau et un traitement différencié à l'intérieur. Une gestion soutenue de la qualité de service par l'architecture **DiffServ** impose un dimensionnement adéquat du réseau et une configuration optimale des paramètres qui interviennent dans la garantie de la qualité de service. La gestion de bout en bout de la qualité de service implique la présence de mécanismes spécifiques de gestion de qualité de service à chaque niveau (réseau backbone et réseaux d'accès). Le goulot d'étranglement se situe bien souvent au niveau des réseaux d'accès. L'introduction de la QoS dans ces réseaux devient donc une nécessité, typiquement dans un environnement sans fil. Cependant, elle doit être faite dans deux cas : pour le trafic entrant dans le réseau d'accès et pour le trafic sortant.

Il est donc nécessaire d'avoir des schémas de différenciation non seulement dans tous les domaines qui constituent le réseau backbone, mais aussi dans les réseaux d'accès où se situent souvent les goulots d'étranglement (voir figure 3.11).

Afin de remédier aux limites posées par EDCF, à savoir l'influence du flux le plus prioritaire sur le moins et les contraintes de qualité de service posés par les flux multimédia (débit et délai, nous proposons ainsi une solution de couplage entre EDCF et DiffServ (le choix de Diffserv et non IntServ revient au fait que la solution IntServ correspond au réseau d'accès, tandis que la solution DiffServ semble plus appréciée pour l'intérieur du réseau lorsqu'il y a beaucoup de flots à gérer)

Dans cette partie, nous présenterons une étude sur l'interopérabilité entre les schémas de qualité de service des réseaux d'accès et du réseau backbone. Nous proposons une solution de couplage entre DiffServ et les schémas de qualité de service dans les réseaux sans fil 802.11.

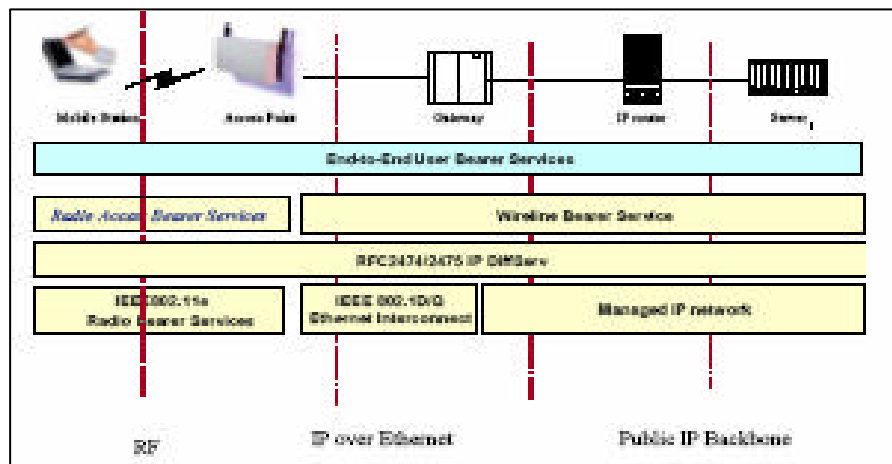


Figure 3. 10 Structure du réseau bout en bout

### 3.3.2 Solution proposée : association des paramètres de DiffServ et l'EDCF :

Le schéma de qualité de service étant défini dans le réseau d'accès, il est maintenant indispensable de l'associer à l'architecture de QoS DiffServ introduite dans le *backbone*. L'interopérabilité entre le champ DSCP et le schéma de QoS dans 802.11e consiste à faire correspondre les niveaux de priorité entre DiffServ et 802.11e. Deux problèmes se posent pour ce couplage. Le premier consiste à déterminer les champs dans la trame 802.11e qui reflètent la différence de service définie par le DSCP de DiffServ. Le second est de définir la compatibilité et les similarités entre les services DiffServ et les services de 802.11e. Il est important aussi de définir à quel instant le marquage des trames pour la qualité de service est effectué

Pour l'architecture DiffServ, le champ DSCP indique non seulement la nature du trafic (important, moins important, pas important) reflétant le niveau de priorité à la perte mais aussi la classe de service à laquelle appartient le paquet. Par conséquent, il nous faut aussi pour la norme 802.11e des indicateurs pour identifier la nature du trafic et le niveau de priorité de chaque trame. Concernant 802.11e, une interface de programmation applicative spécifique a été définie pour demander la qualité de service au niveau de la couche MAC. Elle permet d'identifier si la trame requiert une qualité de service ou non en indiquant la valeur appropriée sur quatre bits (1000) dans le champ *Subtype*. Cette valeur ne définit pas le niveau de priorité de la trame qui est essentiel pour la mise en correspondance dans les différentes files d'attente. Nous utilisons trois bits dans le champ QoS Control pour spécifier le trafic à travers le sous-champ TID (Traffic Identifier). Considérons le sens ascendant où le trafic quitte le réseau d'accès pour aller vers le réseau *backbone*. Dans ce cas, il est nécessaire de marquer les trames avant de les envoyer [18].

Nous proposons dans le tableau 3.1 le codage des bits pour les trames qualité de service. Le nombre limité de bits (3) pour identifier les catégories de trafic nous impose d'avoir seulement 3 classes de service avec 3 niveaux de priorité dans chacune pour le comportement AF. Deux combinaisons sont prises pour les trafics EF et *best-effort*. Ceci rentre bien dans la conception du trafic que nous avons définie (trafic important, moins important et pas important). Le troisième niveau de priorité de toutes les classes de service AF est codé avec "000" et le trafic concerné est considéré comme du *best-effort*. Dans notre solution, le niveau "Gold Low" est plus élevé que le « Silver High ».

Le couplage défini dans le tableau 3.1 permet d'harmoniser la QoS dans les réseaux d'accès et la QoS dans le backbone. La notion de précedence ou la priorité à la perte est ainsi identique pour l'architecture DiffServ et 802.11e. En d'autres termes, le couplage permet à une classe de service d'observer le même comportement dans les deux niveaux de réseaux. Ceci améliore la gestion de bout en bout de la qualité de service des classes de service. Ainsi, pour le réseau global constitué de réseaux d'accès et de plusieurs domaines DiffServ, la qualité de service de bout en bout est déterminée par la partie du réseau qui offre les performances les moins bonnes.

Octet/2	2	6	6	6	2	6	2	0-2312	4
Frame control	Duration/ID	Adress 1	Adress 2	Adress 3	Sequence control	Adress 4	QoS control	Frame body	FCS

**Entête MAC 15 bits (12-15 TCID)**

Bits 12	Bit 13	Bit 14	Bit 15
	Priority selector		0 / 1

**Figure 3. 11 Champ Qos de la trame 802.11**

DIFFSERV			MAC 802.11		
PHB	DSCP	Champ subtype	CHAMP TCID	QoS champ	File d'attente correspondante
EF	101110	1000	111 Prenum		7
AF11	001010	1000	010	Gold	6

			High	
AF12	001100	1000	001 Gold Low	5
AF13	001110	1000	000 Best Effort	0
AF21	010010	1000	100 SilverHigh	4
AF22	010100	1000	011 SilverLow	3
AF23	010110	1000	000 Best Effort	0
AF31	011010	1000	110 Bronze High	2
AF32	011100	1000	101 Bronze Low	1
AF33	011110	1000	000 Best Effort	0
AF41	100010	-	-	-
AF42	100100	-	-	-
AF43	100110	-	-	-
BE	000000	0000 OR 1000	000 Best Effort	0

**Tableau 3. 1 Couplage de DiffServ et schéma de Qos pour la norme 802.11**

De ce fait, la QoS de bout en bout  $Q_{e2e}$  d'une classe de service donnée peut être présentée de la manière suivante  $Q_{e2e} = \min(Q_{DF}, Q_{WL})$  [19]

Où,  $Q_{DF}$  est la QoS obtenue par la classe de service dans **DiffServ**.

$Q_{WL}$  est la QoS obtenue par la classe de service dans le réseau d'accès 802.11

$j = 1, \dots, D$  ;  $D$  est le nombre de domaines **DiffServ** traversés.

$k = 1, \dots, E$  ;  $E$  est le nombre de réseaux **802.11e** participant au réseau global

### 3.3.3 Exemple de mapping :

Le tableau ci-dessous présentera un exemple de mapping entre les deux champs DSCP/TCID :

Classe de trafic	exemple	DSCP	TCID
Classe 1	Voix	(101)xxx EF	0
Classe 2	Vidéo	(100)xxx AF4X	1
Classe 3	Signaling bearer	(010)xxx AF2X	2
Classe 4	Transmission de données	Default	3

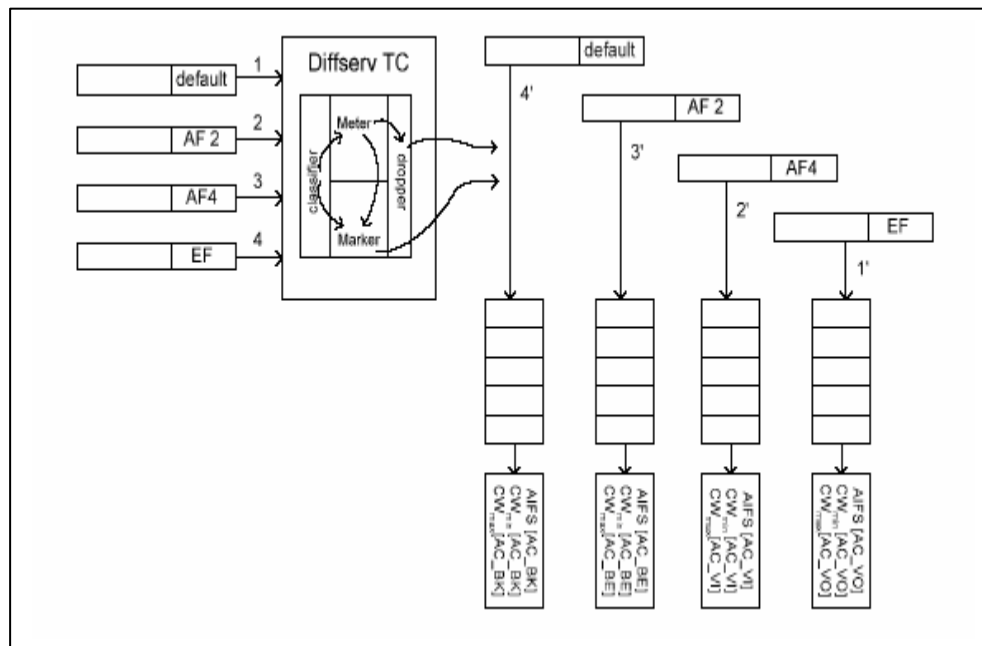
**Tableau 3. 2 Exemple de Mapping**

L'architecture de différenciation de services DiffServ de l'IETF [17] repose sur un modèle simplifié dans lequel le trafic qui rentre dans le réseau est classé et éventuellement régulé aux frontières du réseau, avant de se voir affecter des agrégats de comportement. Les paquets identifiés par leur point de code DSCP marqué en amont, recevront un traitement préférentiel par le réseau. Un routeur supportant DiffServ doit avoir la possibilité d'initialiser ou de réinitialiser le DSCP, en se basant sur les services associés à l'utilisateur ou à l'application.

Dans ce modèle, quand les paquets IP arrive au domaine Diffserv appelé conditionneur du trafic (TC), où ils sont classifiés, ils acquièrent une valeur DSCP et enfin ils sont lissés selon la priorité des valeurs DSCP. Par la suite, ces paquets IP sont encapsulés dans des trames MAC 802.11e et envoyés vers des queues de priorités (AC) de type 802.11e avec des valeurs TCID.

Par exemple, quand des paquets IP arrivent à un Diffserv TC suivant l'ordre des DSCP, AF4, AF2, EF et BE, ils sont lissés respectivement en des classe EF, AF4, AF2 et BE selon leurs priorités. Ensuite ils seront encapsulés en des trames MAC 802.11e.

La figure 3.12, illustre une architecture de qualité de service combinant DiffServ et les classes de trafic associé à la norme 802.11e.



**Figure 3. 12 Qos combinant DiffServ et 802.11e**

#### **Conclusion :**

Les architectures basées sur le modèle de services différenciés telle que **DiffServ** s'appuient sur des mécanismes d'ordonnancement, de prévention et de gestion de congestion pour offrir des services flexibles. Une gestion soutenue de la qualité de service par l'architecture **DiffServ** impose un dimensionnement adéquat du réseau et une configuration optimale des paramètres qui interviennent dans la garantie de la qualité de service. Il reste que la qualité de service dépend des services fournis par les réseaux d'accès où le sans fil occupe une place prépondérante. Nous avons analysé une solution pour introduire la qualité de service dans les réseaux d'accès sans fil. Elle repose sur une modification de la couche MAC dans le réseau 802.11 pour modifier le comportement de la méthode d'accès au canal.

Notre but ultime était d'aboutir à la gestion de bout en bout de la qualité de service. Nous avons contribué à cette gestion dans le réseau *backbone* et proposé des solutions pour l'introduire dans les réseaux d'accès sans fil.

## Chapitre 4 : Simulation et évaluation des performances de DiffServ et 802.11e sous NS-2

---

L'utilisation d'un réseau réel dans une évaluation est difficile et coûteux. Le réseau réel n'offre pas la souplesse de varier les différents paramètres de l'environnement et pose en plus le problème d'extraction de résultats ; c'est pour cela la majorité des travaux d'évaluation de performances utilisent le principe de la simulation vu les avantages qu'ils offrent.

Dans ce chapitre, nous allons d'abord présenter la plate-forme logicielle que nous avons utilisé pour les simulations (NS-2), ensuite, nous allons présenter les modélisations des réseaux ad-hoc à simuler tout en mettant en relief les principes que nous avons adoptés pour intégrer les modèles de qualité de service (DiffServ et la norme 802.11e) dans ces réseaux. Pour ce fait, nous allons définir les paramètres à évaluer et la mise au point du contexte de simulation et nous exposeront enfin, les résultats trouvés et nous donnerons leurs interprétations.

### 4.1 Présentation du Network Simulator

Network Simulator [16] est l'un des outils de simulation les plus populaires au sein de la communauté scientifique. Développé par le département des techniques informatiques à l'université de Berkeley en Californie, NS-2 offre un moteur de simulation de réseaux pour permettre à l'utilisateur de décrire un réseau et de simuler des communications entre ses différents nœuds.

Il s'agit d'un simulateur d'événements discrets orienté objet écrit en C++ avec une interface utilisateur en OTCL (Object Tool Command Language). A travers ces deux langages, il est possible de modéliser tout type de réseau et de décrire les conditions de simulation : la topologie du réseau (LAN, sans-fil, etc...), les caractéristiques des liens physiques, le type de

trafic qui circule, les routeurs et les mécanismes d'ordonnancement à appliquer, les protocoles utilisés, les communications qui ont lieu, etc... Ce dernier a pour essentiel point fort de pouvoir intégrer de nouvelles fonctionnalités et mettre à jour sa bibliothèque : chaque année, une nouvelle version de NS-2 apparaît.

La simulation avec NS-2 passe en général par trois phases :

- Définition de la topologie du réseau : on définit les nœuds et les connexions. On peut définir sur chaque lien, le délai, la bande passante, le fait qu'il soit simplexe ou duplexe et le type de file d'attente se trouvant à son extrémité.
- Spécification du scénario de la simulation : l'utilisateur spécifie les différents agents de la communication qui vont agir pendant la simulation (qu'elle nœud envoie les données, les routeurs actifs et ceux inactifs). Il spécifie aussi la succession des différentes opérations (à l'instant  $t_1$ , envoi des données ; à l'instant  $t_2$  arrêt d'émission). Il spécifie enfin les différents comportements que prend le réseau vis-à-vis de tel ou tel événement.
- Exploitation des résultats : cette dernière phase consiste en un recueil des statistiques de la simulation. Ces dernières peuvent être exploitées directement par NS-2 ou par l'un des outils qui l'accompagnent (outil de tracé graphique : xgraph, outils d'animation de la simulation : nam) ou bien elles seront archivées pour une utilisation ultérieure au moyen d'autres outils de traitements statistiques.

OTCL, dérivé de TCL, est un langage interprété qui ne demande pas de compilation. Il est principalement utilisé pour concaténer des objets, accéder aux objets à partir de l'interpréteur et configurer des simulations (début et arrêt des évènements, perte réseau, rassemblement de statistiques). Il permet une utilisation facilitée du simulateur, son utilisation est rapide et assez conviviale. C++ est utilisé pour créer les classes de base et pour traiter un grand nombre de données (tel que calcul des tables de routage, mouvement des mobiles, protocoles, de files d'attentes ...). La dualité entre ces deux langages s'explique par le fait que NS doit être d'une part efficace dans la manipulation et la gestion de grandes quantités de données et d'autre part rapide pour le changement de scénarios de simulation.

## 4.2 Modèle de l'EDCF implémenté

### 4.2.1 Présentation de L'EDCF

Pour notre application, la version choisie du NS est 2.26. Cette version intègre uniquement la méthode d'accès DCF relative à la norme IEEE 802.11.

Son implémentation exige :

- des modifications de certains fichiers NS tels que :
  - /ns-2.26/Makefile.in.
  - /ns-2.26/tcl/lib/ns-lib.tcl.
  - /ns-2.26/tcl/lib/ns-default.tcl.
  - /ns-2.26/tcl/lan/ns-mac.tcl
- Suppression de certains fichiers tels que :
  - tcl/lib/ns-mobilenode.tcl .
- Intégration des fichiers décrivant le code de la norme 802.11 e.

L'EDCF définit les paramètres suivants [15] :

- Quatre queues à priorité sont uniquement définies au lieu de 8.
- Le facteur de persistance PF est neutralisé à 2.
- La valeur minimale des AIFS est passée de SIFS à DIFS.
- Les paramètres relatifs aux queues ( $AIFS$ ,  $CW_{MIN}$ ,  $CW_{MAX}$ , TXOP-limit) ont acquis de nouvelles valeurs.

Les files d'attente EDCF se caractérisent par des priorités. Ces priorités sont associées à l'une des quatre files d'attente des catégories d'accès EDCF par une instance définie au dessus de ces files. Cette instance dérive de la classe *Priqueue* de NS [13], elle est noté *priq*. Sous cette instance, dans un fichier nommé *priority.tcl*, les quatre files d'attente EDCF sont définis. Chacune de ces files est une extension du type *Drop-Tail* de NS [10], auxquelles sont associés les paramètres qui figurent dans le tableau 4.1.

Notons que, dans notre cas, la priorité '0' est définie comme la plus haute priorité.

AC	CW <sub>MIN</sub>	CW <sub>MAX</sub>	AIFS	TXOP-limit
BK (3)	31	1023	7	0
BE (2)	31	1023	3	0
VI (1)	15	31	2	6.016ms
VO (0)	7	15	2	3.264ms

Tableau 4. 1 Les paramètres des ACs de l'EDCF implémenté (draft novembre 2003)

## 4.2.2 Paramètres de simulation

### 4.2.2.1 Paramètres pour la création des scénarios

On procède tout d'abord à la définition des différents paramètres du réseau à simuler. Ces paramètres, tels qu'ils sont définis au niveau du script TCL [16], sont :

- *Val (chan)* : Type du canal : sans fil (*Channel/Wireless Channel*)
- *Val (prop)* : indique le modèle de la propagation radio utilisé (Propagation/TwoRayGround). La propagation se fait dans l'espace libre qui atténue le signal de  $1/d^2$  où  $d$  est la distance entre les nœuds.
- *Val (netif)* : Interface physique pour accéder au réseau : (*Phy/WirelessPhy*).
- *Val (ant)* : Antenne Omnidirectionnelle (*Antenna/Omni Antenna*).
- *Val (LL)* : indique le type de la couche LLC utilisée
- *Val (mac)* : indique le modèle de la couche MAC utilisé. Dans nos simulations, nous avons deux modèles : le modèle existant IEEE 802.11 (*Mac/802.11*) et le modèle que nous avons introduit de l'IEEE 802.11e (*Mac/802.11e*)
- *Val (ifq)* : précise le modèle de la file d'attente au niveau de chaque nœud. Nous avons utilisé, dans nos simulations deux types de queue : le type drop Tail (*Queue/DropTail/PriQueue*) et le type DTail que nous avons défini (*Queue/DTail/Priq*)
- *Val (ifqlen)* : Taille max des files d'attente. Sa valeur par défaut est de 50 paquets dans la file. La taille définie par la suite est relative à chaque type de service.

- **Val (rp)** : indique le protocole de routage utilisé. Nous avons choisi aléatoirement le protocole DSDV pour toutes les simulations. Ce choix n'a pas d'importance, puisqu'il n'influe pas sur les résultats obtenus. Mais, il est nécessaire de le mentionner.
- **Val (nn)** : indique le nombre des noeuds du réseau.
- **Val (x)** : précise la longueur de la topologie fixée à 100 m.
- **Val (y)** : précise la largeur de la topologie fixée à 100 m.
- **MAX\_SPEED** : Vitesse maximale avec laquelle un noeud peut se déplacer. Nous l'avons fixé à 1m/s.
- **PAUSE\_TIME** : temps maximal durant lequel un noeud reste au repos.
- **SIMULATION\_TIME** : Durée de simulation en seconde. Nous l'avons fixés à 190 secondes pour tous les scénarios.

### 4.2.2.2 Définition des types de trafic

Dans nos simulations, nous allons considérer trois types de trafic : voix, vidéo et données. Les paramètres relatifs à chaque type de trafic sont illustrés dans le tableau suivant :

Service	Trafic	Taille de paquet (octets)	Débit (Kbps)
Voix	CBR	500	400
Vidéo	CBR	500	400
Données	CBR	500	400

**Tableau 4. 2 Les paramètres des trafics simulés**

La taille des files d'attente pour la voix et la vidéo sont fixées respectivement à 30Kbit et 38Kbit pour éliminer les paquets de délai excessif puisque les services voix et vidéo sont sensibles au délai, mais tolère un peu de pertes. Pour le service transmission de données, la taille des files d'attente est infinie pour assurer un taux de perte nul comme ce service est tolérant au délai, mais il nécessite une transmission sans pertes.

La différenciation de services est assurée en associant chacun des services à une catégorie d'accès relative à l'IEEE 802.11e. Selon la définition des catégories d'accès et les caractéristiques des services, l'association se fait de la manière suivante

Service	Catégorie d'accès	Priorité
Voix	AC_VO	0
Vidéo	AC_VI	1
Donnés	AC_BK	3

**Tableau 4. 3 Association trafic/AC**

### 4.3 Modèle de DiffServ

La configuration du modèle DiffServ dans un réseau dans NS2 passe par les étapes suivantes:

- Etablir la nature des routeurs du réseau : routeur de bordure (edge) ou routeur (core)

*Exemple de code:* \$ns simplex-link \$edge \$core 1Mb 1ms dsRED/edge

\$ns simplex-link \$core \$edge 1Mb 1ms dsRED/core

- Configurer les files d'attentes suivant le modèle DiffServ au niveau des liens entre les différents routeurs.

*Exemple de code :* set qEC [[ \$ns link \$edge \$core ] queue]

- Ajouter les DiffServ Policy. Ces derniers permettent de spécifier pour chaque trafic le niveau de service permis dans le réseau.

Cette différenciation de services s'établit en deux étapes :

- Configurer les entrées des policy tables en indiquant pour chaque trafic le policer approprié ainsi que ces paramètres accompagnants.

(Code point, CIR, PIR, CBS, ..). Cette étape se réalise au niveau des Files d'attentes des edge node .

-Le policer utilisé dans notre travail est :

TokenBuket	Inital code point	CIR	CBS
------------	-------------------	-----	-----

Où : CIR est un débit exprimé en bits/s, et CBS est exprimé en octet.

*Exemple de code:* \$qEC addPolicyEntry [\$s1 id] [\$dest id]

TokenBucket 10 \$cir0 \$cbs

\$qEC addPolicerEntry TokenBucket 10 11

Où:

10: Initial code point.

11 : Downgraded code point N°1.

- Configurer les paramètres d'entrées aux tables PHB selon le policer adopté. Cette étape se réalise au niveau des core node .

*Exemple de code :*

```
$qEC addPHBEntry 10 0 0
```

```
$qEC addPHBEntry 11 0 1
```

Où :

10 : initial code point.

0 : indice la file d'attente physique.(pour les deux lignes)

0 : indice de la file d'attente virtuelle.

11 : downgraded code point N°1.

- Configurer les paramètres relatifs à chaque file d'attente, ainsi que l'Ordonnanceur des files. Les ordonnanceurs implémentés dans NS2 sont: WRR (Weighted Round Robin) WIRR (Weighted Interleaved Round Robin), Round Robin (RR), et Priority (PRI).

## 4.4 Paramètres de Simulation

### 4.4.1 Débit utile (throughput)

C'est le débit total en réception. Il correspond à une somme discrète des débits utile mesurés successivement sur des intervalles de même longueur  $\Delta_t=0.5s$ . Ainsi, le débit utile calculé ici est une fonction discrète de temps. Mais vue que  $\Delta_t$  est relativement petite, nous pouvons l'assimiler à une fonction continue du temps.

Donc, nous définissons le débit utile à l'instant  $T_n$  par :

$$\text{débit utile}(T_n) = \frac{\text{nombre de bits reçus entre } T_{n-1} \text{ et } T_n}{\Delta_t} \quad \text{telque } n \in [1, p]$$

Avec :  $p = [E(\text{durée de la simulation} / \Delta_t)] + 1$

L'opération de calcul du nombre de bits reçus est effectuée en introduisant, dans le script TCL, un agent de type **Loss Monitor** attaché à la source. Cet agent est une sorte de puits intelligent capable de compter les données reçues ainsi que les pertes.

#### **4.4.2 Le taux de pertes**

Ce taux de pertes est modélisé par le nombre de bits perdus en fonction du temps. Comme pour le débit utile, l'agent *Loss Monitor* est responsable d'enregistrer le nombre de paquets perdus dans la variable associée *n\_lost*. Ainsi, le taux de pertes est modélisé par :

$$\text{Taux de pertes } (T_n) = \text{nombre de paquets perdus entre } T_{n-1} \text{ et } T_n \times \text{taille du paquet} \times 8$$

*telque*  $n \in [1, p]$

Où *p* est défini de la même manière que pour le débit utile.

Pour certaines courbes relatives à ce paramètre, nous avons choisi de représenter le nombre de paquets perdus en fonction du temps pour des raisons de clarté de figure.

#### **4.4.3 Le délai**

C'est le temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire. Le délai tient compte du délai de propagation le long du chemin et du délai de transmission induit par la mise en file d'attente des paquets dans les systèmes intermédiaires.

Pour calculer ce délai, nous avons filtré le fichier trace par un programme écrit en langage AWK [18]. Ce programme, calcule le délai pour chaque paquet émis :  $t = t_r - t_e$  où :  $t_r$  est le temps de réception et  $t_e$  est le temps d'émission.

Ainsi, toutes les courbes de délai qui suivent représentent le délai en fonction du temps d'émission.

### **4.5 Simulation : résultats et interprétations**

#### **4.5.1 Introduction**

Dans cette partie, nous allons aborder la partie simulation : la création des scénarios en mode infrastructure, les simulations des différents paramètres de qualité de service introduits au niveau de la partie précédente pour les deux modes EDCF et DiffServ l'interprétation des résultats des simulations pour ces deux mécanismes de différenciation de service

#### **4.5.2 EDCF (résultats et interprétations) :**

Pour mettre en évidence le mode EDCF, nous avons considéré un WLAN en mode infrastructure constitué par une station voix, une vidéo et une donnée, tous mobiles. Chaque station émet vers le point d'accès. La figure suivante présente la topologie du réseau relatif à ce scénario :

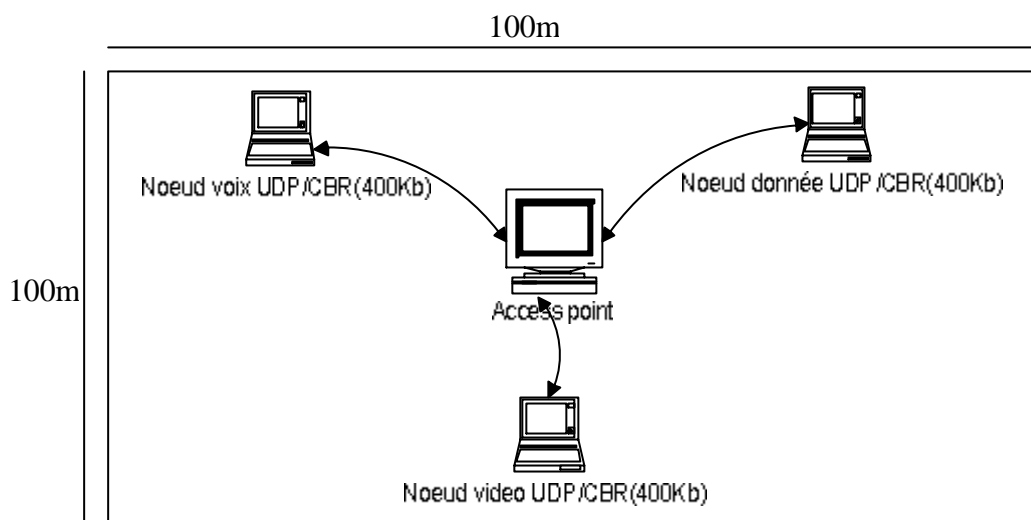


Figure 4. 1 Scénario EDCF

Les stations mobiles, sont des agents UDP, auxquels nous avons attaché des sources de trafic CBR de paramètres relatifs au service fourni. La station voix commence à émettre à  $t=0s$  et sera arrêté à  $t=140s$ , la station vidéo commence son émission à  $t=40s$  jusqu'à  $160s$  enfin la station donnée émet à  $t=60s$  jusqu'à la fin du temps de simulation  $190s$ .

A partir des fichiers traces générées pour la topologie nous avons tracé les courbes relatives aux paramètres de qualité de service.

La figure 4.2 représente la courbe de débit utile avec EDCF pour chacun des trafics considérés.

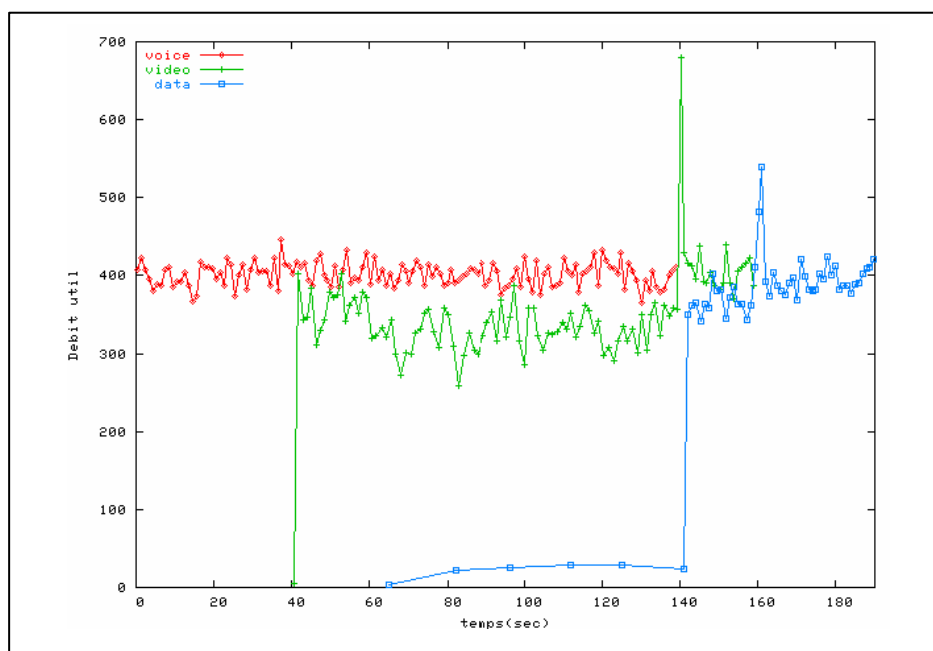


Figure 4. 2 Courbe de débit avec EDCF (Kbits/s)

L'analyse de la courbe précédente montre bien que l'EDCF permet de favoriser les trafics les plus prioritaires. Plus le trafic est prioritaire, plus il est servi. Donc notre hypothèse à vérifier à ce niveau est la suivante «le débit utile du trafic le moins prioritaire se dégrade si le débit utile du trafic le plus prioritaire augmente ».

La figure 4.2 confirme bien l'hypothèse précédente. Le débit utile de la voix ne diminue pas même lorsque la station vidéo commence son émission à  $t=40s$ , à savoir cette dernière n'augmente que lorsque la station voix achève son émission.

A  $t=60s$ , la station donnée commence à émettre avec un débit très faible pendant toute la durée de l'émission de la station voix et elle n'est très bien servie que lorsque celle-ci est arrêtée (à  $t=140s$ ) pour atteindre une valeur acceptable.

Ainsi notre hypothèse sera plus défendue au niveau des autres figures à savoir le délai et les pertes générés au niveau de chaque type de trafic.

Ainsi, les pertes générées au niveau de la station donnée diminuent lorsque celle de la voix achève son émission. Celle-ci passe de 200paquets par secondes jusqu'à à peu près 10 paquets par secondes à  $t=140s$ .

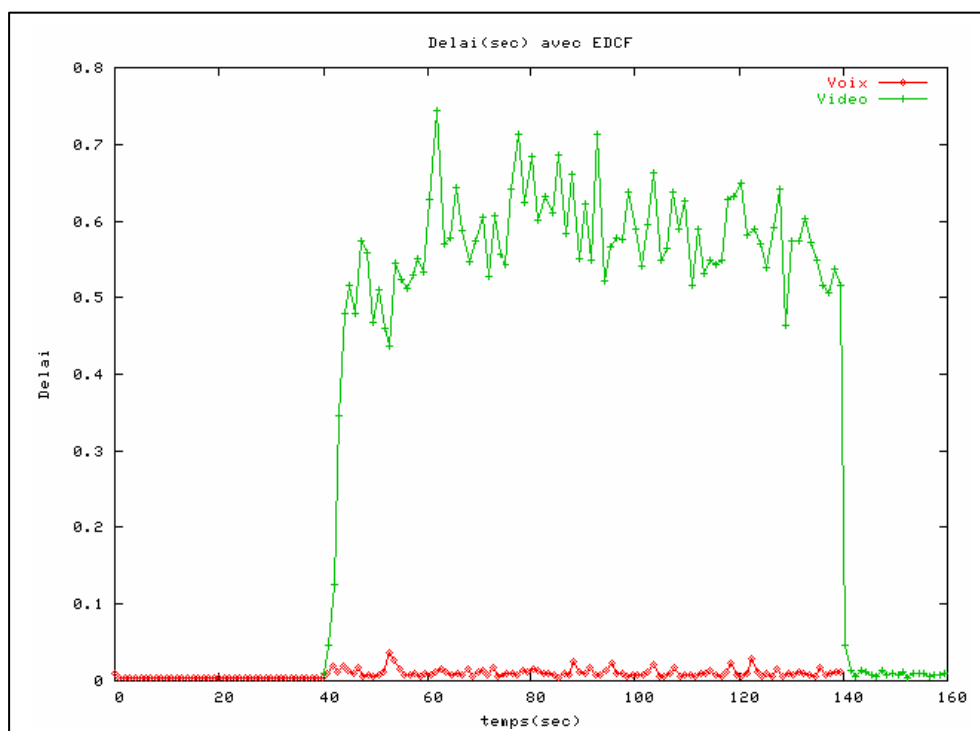


Figure 4. 3 Courbe de délai avec EDCF

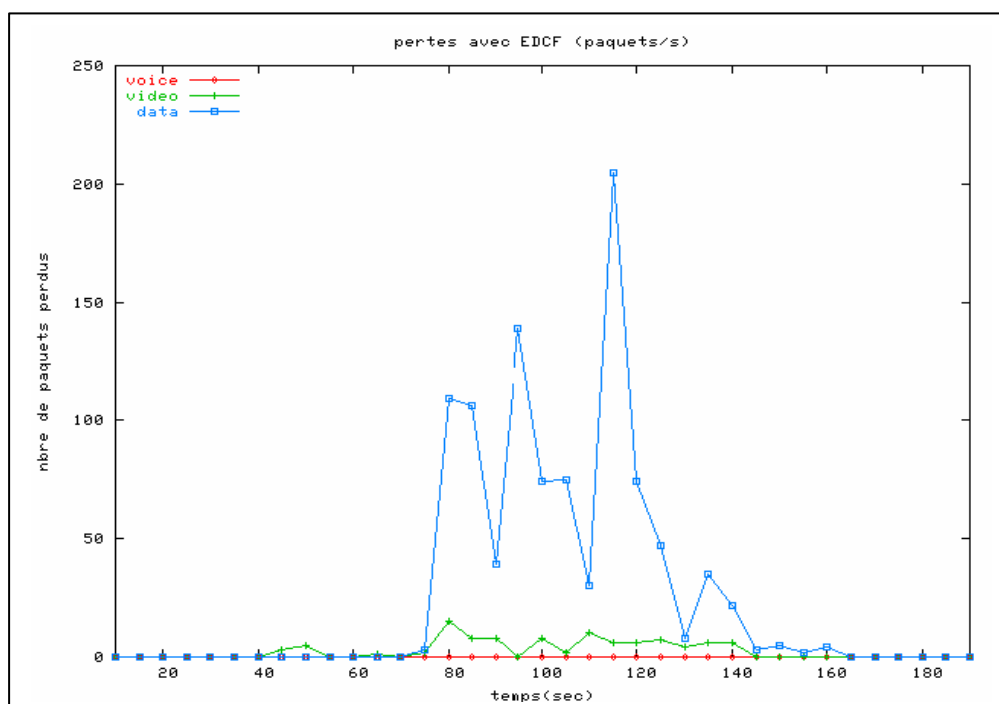


Figure 4. 4 Courbe de pertes avec EDCF

### 4.5.3 DiffServ (résultats et interprétations) :

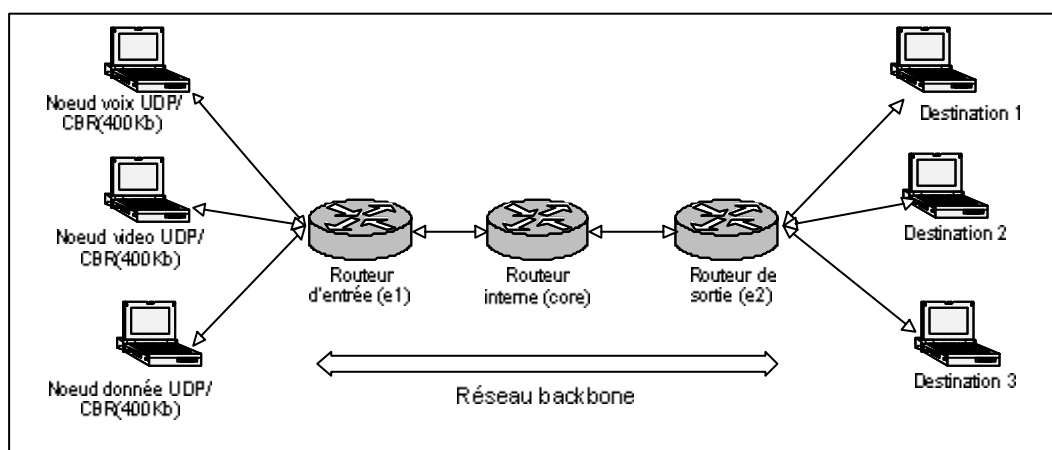
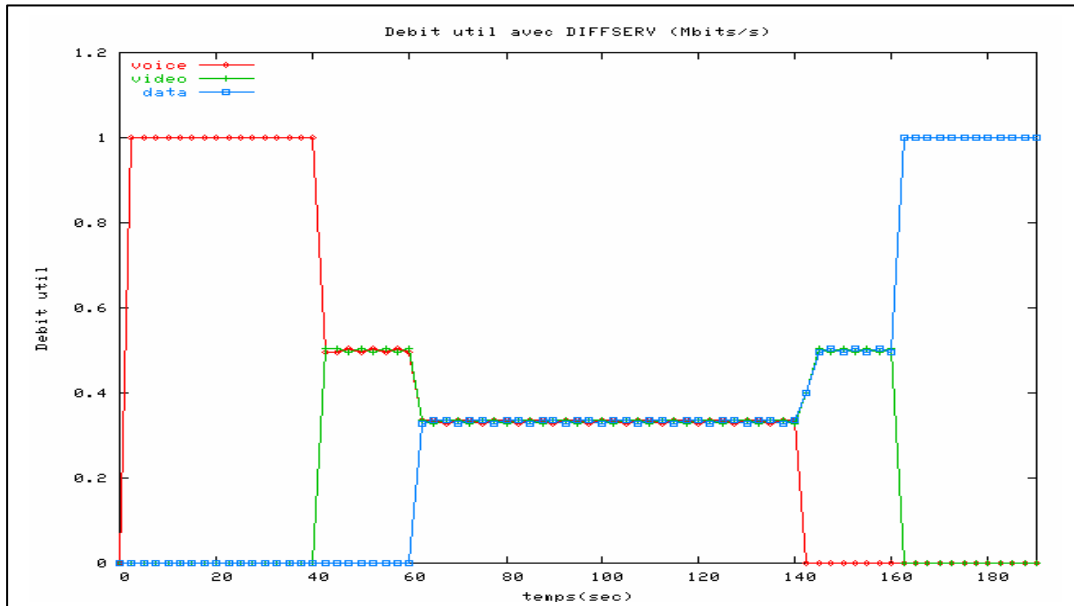


Figure 4. 5 Topologie avec DiffServ

Pour mettre en évidence le modèle Diffserv nous avons simulé la topologie décrite ci-dessus. Les stations mobiles, sont des agents UDP, auxquels nous avons attaché des sources de trafic CBR de paramètres relatifs au service fourni. La station voix commence à émettre à  $t=0s$  et sera arrêté à  $t=140s$ , la station vidéo commence son émission à  $t=40s$  jusqu'à  $160s$  enfin la station donnée émet à  $t=60s$  jusqu'à la fin du temps de simulation  $190s$ .

A partir des fichiers traces générées pour la topologie nous avons tracé les courbes relatives aux paramètres de qualité de service.

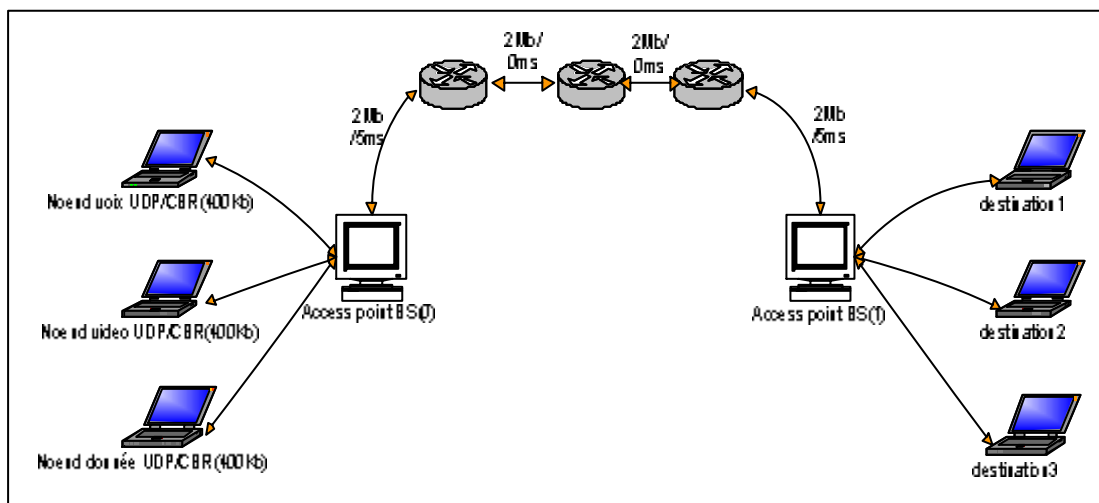


**Figure 4. 6 Courbe de débit utile avec DiffServ**

La figure 4.6 confirme très bien la viabilité d'un service Diffserv pour les flux voix, vidéo et donnée, à savoir que la distribution de la bande passante est équitable.

D'après la courbe, pendant [0s,40s] le trafic voix est le seul servi alors il exploite la bande passante entière. Ce débit se réduit à la moitié dès que la station vidéo commence son émission, ensuite il atteint le 1/4 lorsque la station donnée est activée.

### 4.5.4 Couplage de DiffServ et EDCF (résultats et interprétations) :



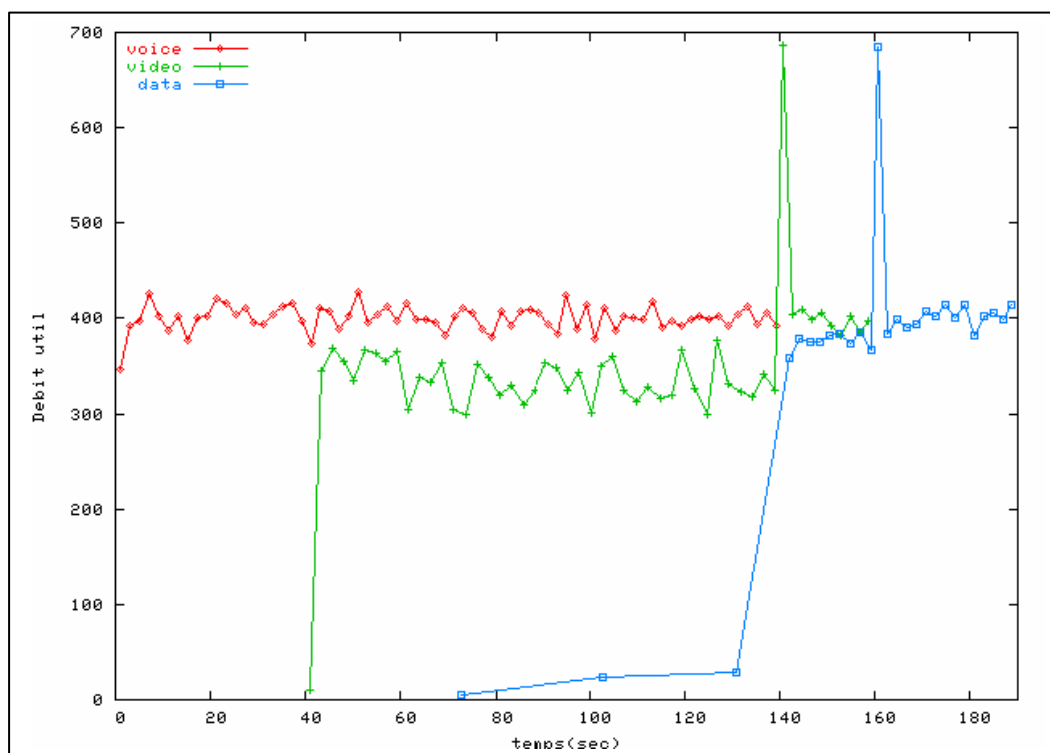
**Figure 4. 7 Topologie à simuler : gestion de bout en bout**

Afin de montrer l'utilité d'avoir la qualité de service de bout en bout, nous avons simulé la configuration définie dans la figure 4.7 le réseau global est constitué de deux réseaux d'accès filaires et d'un domaine DiffServ.

Notre modèle de simulation est composé de trois stations mobiles qui émettent des flux cbr0, cbr1 et cbr2 modélisant respectivement la voix, la vidéo et les données. Les sources transmettent respectivement à  $t=0s$ ,  $40s$  et  $60s$  avec un débit de  $400Kb$ .

Les paramètres de simulation dans l'environnement 802.11 sont celles définies dans le tableau 4.1.

A partir des fichiers traces générées pour la topologie nous avons tracé les courbes relatives aux paramètres de qualité de service suivants : débit utile, délai et taux de pertes de paquets.



**Figure 4. 8 Débit utile avec Qos (Kbits/s)**

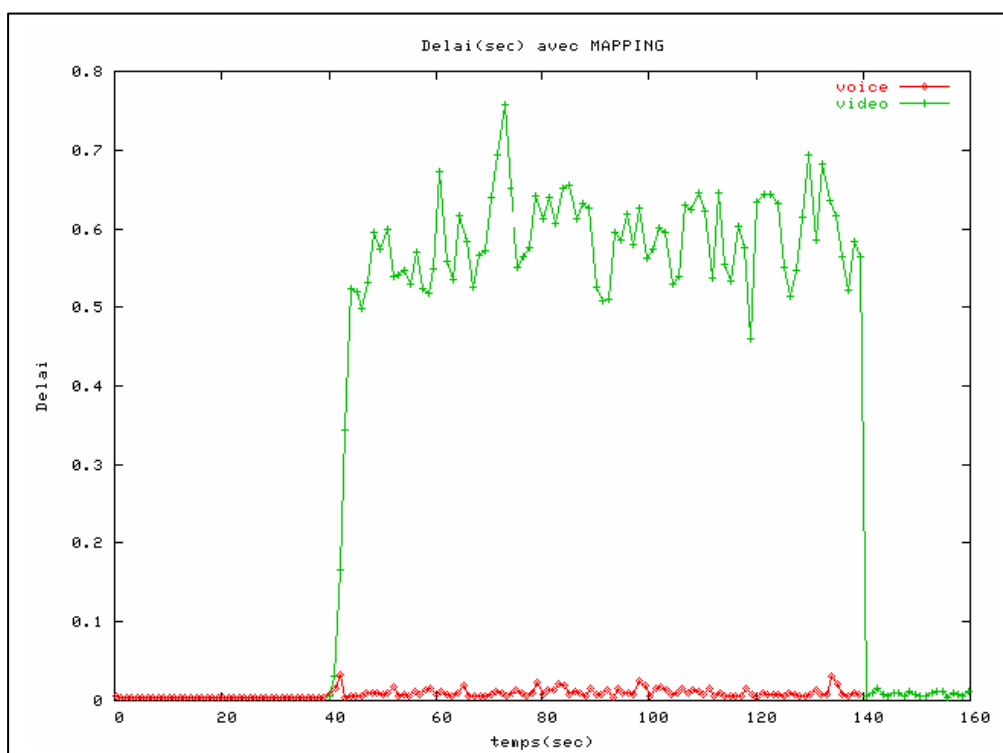
L'architecture proposée gère très bien la gestion de la qualité de service de bout en bout. Les résultats montrent une nette amélioration des performances du trafic donnée (trafic le moins prioritaire) par rapport aux autres trafics, à savoir voix et vidéo.

L'analyse de la courbe présentant le débit utile de chaque type de trafic en fonction du temps, montre bien l'amélioration du débit du trafic donnée qui subit une croissance pendant

l'intervalle [65s, 130s] pour atteindre 370 Kbits/s à  $t=140s$  (temps d'arrêt de la station voix) avec un débit de l'ordre de 40Kbits/s sans introduction de la qualité de service.

A  $t=165s$ , après que la station voix achève son émission le débit utile du trafic donnée atteint 700 Kbits/s avec gestion de la qualité de service par contre il ne dépasse pas 550Kbits/s sans le couplage.

En conclusion on peut dire que la combinaison de la différenciation de service au niveau IP et MAC fournit un meilleur traitement pour les paquets de faible priorité dans les files d'attente, et accroît sa probabilité d'accès au canal.



**Figure 4. 9 Délai avec Qos de bout en bout**

La figure 4.9 illustre le délai de bout en bout des flux voix et vidéo. On observe un délai nul à  $t=60s$  (début de transmission du trafic donnée) au cas où aucun mécanisme de différenciation n'est utilisé, ce délai augmentera par l'introduction de la gestion de qualité de service de bout en bout pour atteindre la valeur de 10 ms.

En contre partie, le délai du trafic vidéo subit une légère diminution pour passer de 0.75s (à  $t=60s$ ) à 0.69s avec gestion de qualité de service.

Nous avons finalement évalué les pertes au niveau de chaque type de trafics, présentée dans la figure 4.10. Après avoir activé la station donnée, les pertes sont de l'ordre de 100 paquets/s

pour le cas sans Qos (à  $t=75s$ ) pour se réduire à 30 paquets/s en cas d'introduction de la nouvelle architecture. Ces pertes peuvent atteindre la valeur de 30 paquets/s lorsque la station voix cesse d'émettre ; mais généralement ce taux ne dépasse pas 70 paquets/s en cas d'introduction de la gestion de qualité de service de bout en bout.

En conclusion, le couplage défini permet d'harmoniser la qualité de service dans les réseaux d'accès et dans le backbone. En effet, il permet à une classe de service d'observer le même comportement dans les deux niveaux de réseaux ceci afin d'améliorer la gestion de bout en bout de la qualité de service des classes de trafics.

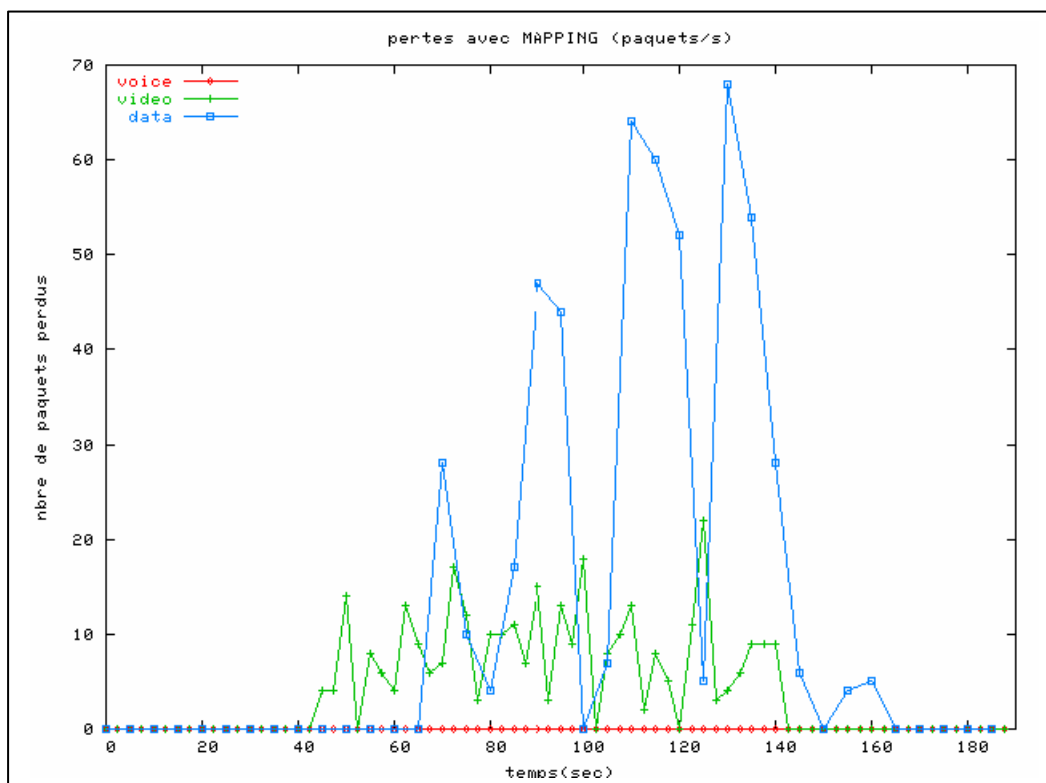


Figure 4. 10 Courbe de pertes avec Qos de bout en bout

### 4.5.5 Evaluation des performances des approches simulées :

Afin de mettre en évidence l'apport du couplage de l'EDCF et l'architecture DiffServ, nous avons simulé un réseau local sans fil (WLAN) constitué par six stations voix, deux stations vidéo et deux stations données. Chaque station émet vers une station puit. Les stations voix commencent leurs émissions à  $t=0s$  et s'arrêtent à  $t=140s$ , les stations vidéo commencent à  $t=40s$  et s'arrêtent à  $t=160s$  alors que celles des données commencent à  $t=60s$  et s'arrêtent à  $t=190s$ . Toutes les stations gardent les mêmes paramètres ainsi définis dans le tableau 4.1.

L'analyse de la figure 4.11 prouve l'apport du «mapping». Cependant lorsque le médium est équitablement partagé entre les stations, tous les paquets ont la même probabilité pour accéder au canal. Inversement, quand seule la différenciation de service MAC est utilisée pour contrôler l'accès au lien, les paquets les moins prioritaires (trafic donnée) sont retardés par les paquets les plus prioritaires (trafic voix).

Cette figure montre une amélioration des débits voix et donnée. En utilisant le «mapping» le trafic voix révèle une augmentation du débit de 150 Mb à 350 Mb, à l'instant  $t=2s$ . Concernant le trafic donnée, le débit augmente de 500Mb à 600 Mb.

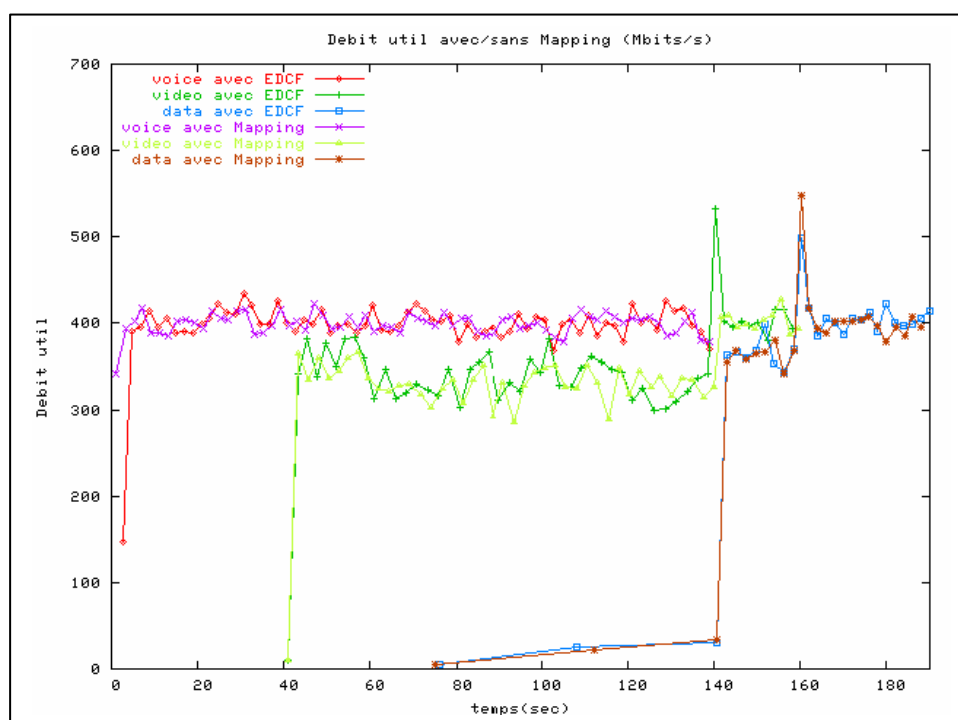
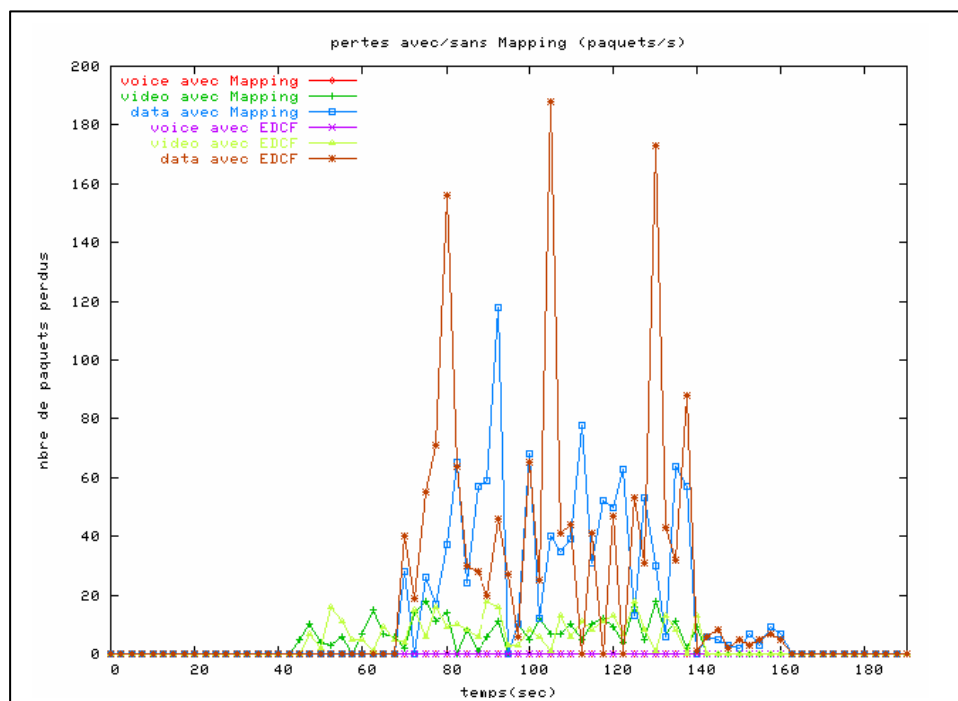


Figure 4. 11 Débit utile avec/sans Mapping

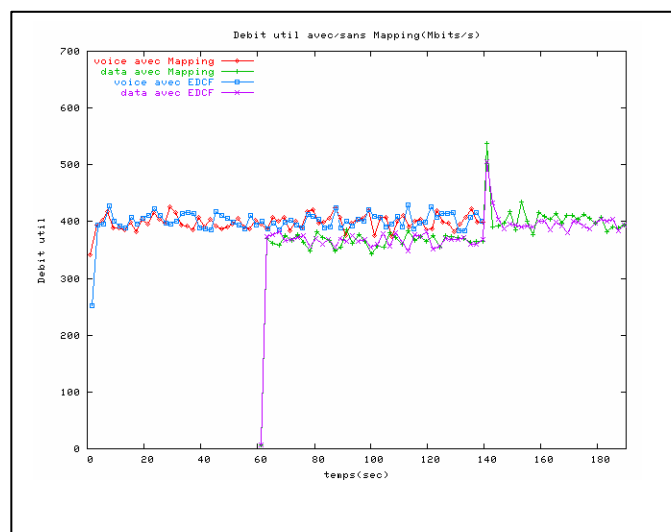
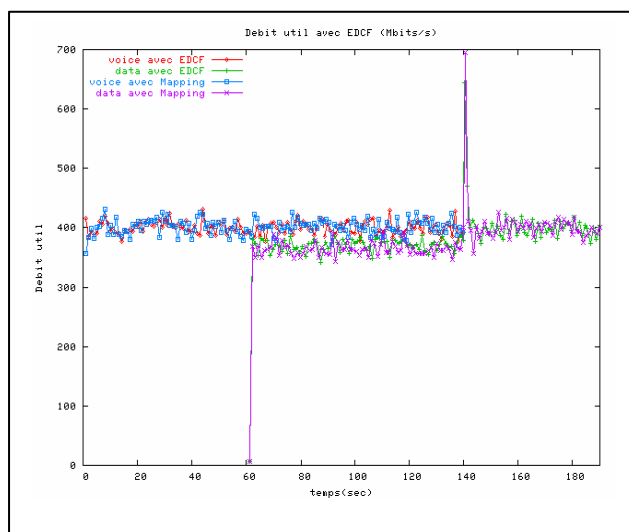
En observant la figure relative aux pertes, on remarque que le taux de pertes diminue en passant de l'EDCF au «mapping». Ce taux passe de 100 paquets/ secondes pour atteindre 60 paquets avec mapping. Cette diminution revient au fait que le trafic donnée est mieux servi avec mapping même en présence d'un grand nombre de stations voix.



**Figure 4.12 Pertes avec/sans Mapping**

Pour mieux montrer l'apport du mapping nous avons simulé trois topologies afin de mettre en évidence l'influence du trafic le plus prioritaire sur le moins prioritaire en terme de débit et pertes : la première comporte deux stations voix et une seule station données, la deuxième comporte quatre stations voix et une seule station données et la troisième comporte six stations voix et une seule station données.

La figure 4.13 illustre respectivement l'évolution des débits utiles des deux trafics voix et données pour les trois topologies considérées.



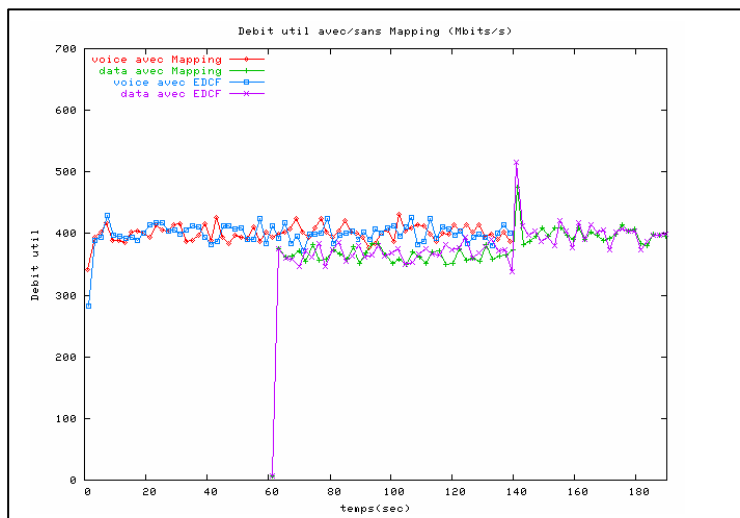
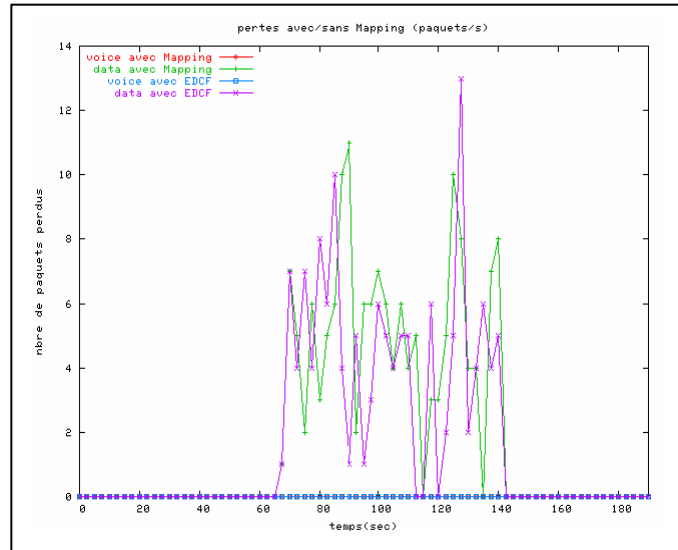
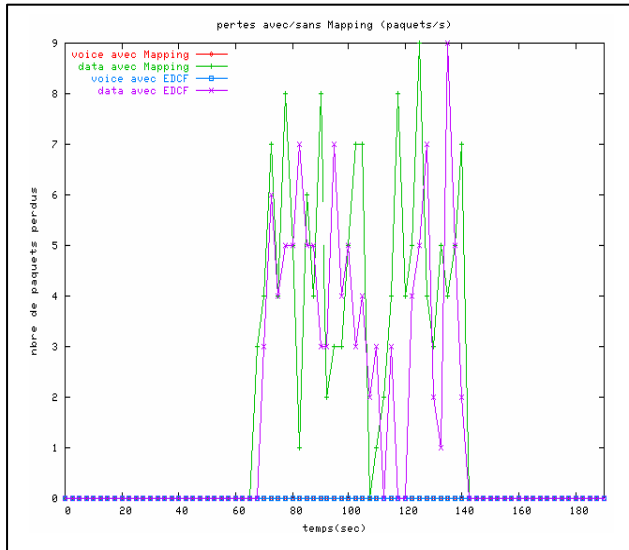
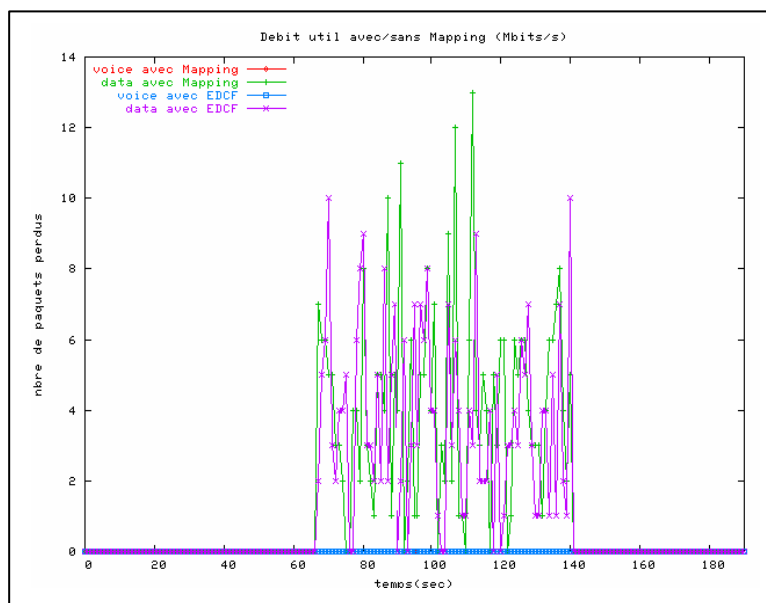


Figure 4. 13 Débit utile avec/sans mapping

La figure 4.13 confirme très bien notre hypothèse précédente. Le débit utile de la voix augmente en introduisant de nouveaux noeuds alors que le débit du trafic données se dégrade avec et sans mapping.





**Figure 4.14 Pertes avec/sans mapping**

L'analyse de la figure 4.14 montre très bien que les pertes des trafics diminuent progressivement avec mapping même en augmentant le nombre des stations voix.

D'après tous ces résultats, il est clair que réaliser une différenciation de service IP dans les réseaux filaires ou mobiles fournit une meilleure qualité de service surtout pour les trafics les moins prioritaires. Les paquets prioritaires peuvent être retardés par des paquets moins prioritaires. Dans ce cas, les paquets les moins prioritaires peuvent aussi attendre plus de temps pour accéder au médium. Par conséquent, les paquets prioritaires sont encore plus pénalisés dans leur accès au lien.

Notre solution combine alors les avantages apportés par les fonctionnalités de qualité de service de chaque couche IP et MAC.

### **4.6 Conclusion :**

Notre but ultime était d'aboutir à la gestion de bout en bout de la qualité de service. Nous avons contribué à cette gestion dans le réseau *backbone* et proposé une solution pour l'introduire dans les réseaux d'accès sans fil. Notre dernière contribution a été donc une étude du couplage (mapping) entre DiffServ et les schémas de qualité de service dans le réseau d'accès 802.11e. Nous avons mis en correspondance les services de DiffServ et ceux issus des schémas de qualité de service dans 802.11e. La solution originale de couplage que nous avons proposée s'avère facilement implémentable grâce à une relation simple entre le champ DSCP de DiffServ et les champs des trames 802.11e réservés à la qualité de service.

## Conclusion générale

---

De nos jours, IEEE 802.11 est la norme des réseaux locaux sans fil la plus déployée mais elle ne garantit pas une qualité de service suffisante pour les différents types de service. De ce fait, Le groupe de travail 802.11 cherche à mettre en œuvre une nouvelle version de la norme 802.11 à savoir la norme 802.11e qui a pour but de garantir la qualité de service pour un grand nombre d'utilisateurs et pour les applications multimédias.

Dans ce contexte, l'introduction de la différenciation de service au niveau MAC afin d'améliorer la qualité de service s'avère important. L'objectif de ce projet de fin d'étude s'articule autour de cette analyse des problèmes de garantir de qualité de service dans les réseaux 802.11 et de l'apport de la norme 802.11e ainsi que le mécanisme de différenciation de service niveau IP et le couplage entre ces deux mécanismes. En effet, La qualité de service au niveau IP et MAC permet de différencier les flux et de garantir la qualité de service des applications temps réel sensibles au délai et pertes de paquets.

En premier lieu, nous avons introduit les notions de base concernant les réseaux locaux sans fil de type IEEE 802.11. Nous avons également présenté un aperçu sur la norme 802.11 avec ses deux couches physique et liaison de données.

En deuxième lieu, nous avons abordé les mécanismes d'accès au médium sans fil pour la norme IEEE 802.11, à savoir le mode distribué DCF et le mode centralisé PCF. A partir des limites de ces deux méthodes en terme de qualité de service, nous avons introduit la norme 802.11e avec ces deux méthodes d'accès EDCF (DCF amélioré) et HCF.

En troisième lieu, nous avons traité les mécanismes de différenciation de service niveau IP, à savoir les services intégrés et différenciés. A partir des problèmes de différenciation de services dans les WLANs nous avons mis le point sur l'insuffisance de l'EDCF d'où la nécessité de l'introduction de DiffServ (mapping). De ce fait nous avons présenté les principes du couplage.

Enfin, dans la dernière partie de ce projet, nous avons entamé la partie simulation sous NS-2.26 afin d'évaluer les performances de l'EDCF ainsi que celle de l'architecture proposée. D'abord, nous avons commencé par la définition de l'environnement de simulation basé sur la présentation du modèle de l'EDCF simulé et Diffserv, la définition des paramètres de simulation et des critères de qualité de service à simuler. Toutefois, nous avons montré que les performances des trafics de faible priorité se dégradent lorsque la charge des trafics de haute priorité augmente. A partir des simulations et des différentes courbes obtenues, nous avons conclu que le couplage fournit une amélioration sensible des performances du trafic le moins prioritaire.

Le sujet abordé dans ce projet pourra être encore amélioré.

---

# Bibliographie

---

## Spécifications ANSI/IEEE (release 1999)

- [1] ANSI/IEEE Std 802.11, *Wireless LAN Medium Access Control (MAC ) and Physical Layer (PHY) Specifications*, 1999 Edition Part11.

## Articles et livres

- [2] N. Prasad, A. Prasad, *WLAN Systems and Wireless IP for Next Generation Communication*, Edition Artech House 2002.
- [3] G. Pujolle, Davor Males, *Wi-Fi par la pratique*, Editions Eyrolles, premier triage 2002.
- [4] G. Pujolle, *Les réseaux*, Editions Eyrolles, deuxième triage 2000.
- [5] Stephan Mangold, Sunghyun Choi, Guido R.Hiertz, Ole Klein, Bernhard Walke, *Analysis of IEEE 802.11e for QoS support in wireless LANs*, IEEE Wireless Communications, Decembre 2003.
- [6] Wasan Pattara-Atikom, Prashant Krishnamurthy, Sujata Banerjee, *Distributed mechanisms for Quality of Service in Wireless LANs*, IEEE Wireless Communications, Juin 2003.
- [7] Daqing Gu, Jinyum Zhang, *QoS Enhancement in IEEE 802.11 Wireless Local Area Network*, IEEE Communications Magazine, Juin 2003.
- [11] [IETF98a] IETF, RFC2474: Définition des champs DS dans les en-têtes d'IPV4 et IPV6, décembre 1998 :

<ftp://ftp.rfc-editor.org/in-notes/rfc2474.txt>

- [12] [IETF98b] IETF, RFC2475: Définition de l'architecture de DiffServ, décembre 1998:  
<ftp://ftp.rfc-editor.org/in-notes/rfc2474.txt>
- [13] [UNIV-CAR] Ketan Mayer-Patel, Université de Caroline du nord, Etude du modèle DiffServ, été 2002
- [17] Seyong Park, Kyungtae Kim, Doug C. Kim, Sunghyun Choi, Sangjin Hong  
«Collaborative QoS Architecture between DiffServ and 802.11e Wireless LAN »
- [18] An AWK Tuorial , Version 1.0.4, 01 Janvier 2002
- [19] Ibrahima NIANG, Hossam AFIFI, Dominique SERET « Couplage de services différenciés pour une QoS de bout en bout »

### Sites Web

- [8] [http://www.cs.unc.edu/Courses/comp249-02/lectures/comp249\\_s02\\_18/sld001.htm](http://www.cs.unc.edu/Courses/comp249-02/lectures/comp249_s02_18/sld001.htm)
- [9] <http://www.urec.cnrs.fr/metrologie/article-qos.html>
- [10] <http://www.rennes.enst-bretagne.fr/~medina/these/these-medina.pdf>
- [14] Network Simulator, <http://www.isi.edu/nsnam/ns/>
- [15] <http://www-sop.inria.fr/planete/software>
- [16] Marc Greis : <http://www.isi.edu/nsnam/ns/tutorial>

## Annexe : Le simulateur NS-2

Une des manières d'évaluer un nouveau protocole est de l'implémenter dans un simulateur réseau pour effectuer des tests concrets. De nos jours, le simulateur réseau NS est une référence dans le domaine des réseaux et permet de comparer les protocoles entre eux de manière directe dans des environnements proches de ce qu'on trouve dans l'Internet. NS permet notamment de simuler des flux proches de la réalité sur les équipements et de visualiser l'échange des paquets.

Si le script écrit en OTCL permet une utilisation facile du simulateur, les routines sont elles écrites en C++ pour avoir une plus grande puissance de calcul. Un grand nombre de classes est prédéfini. Ces classes mettent en oeuvre plusieurs types de protocoles, de files d'attente, de sources et algorithmes de routage.

La figure ci-dessous présente de façon simplifiée le schéma de l'utilisation de NS.

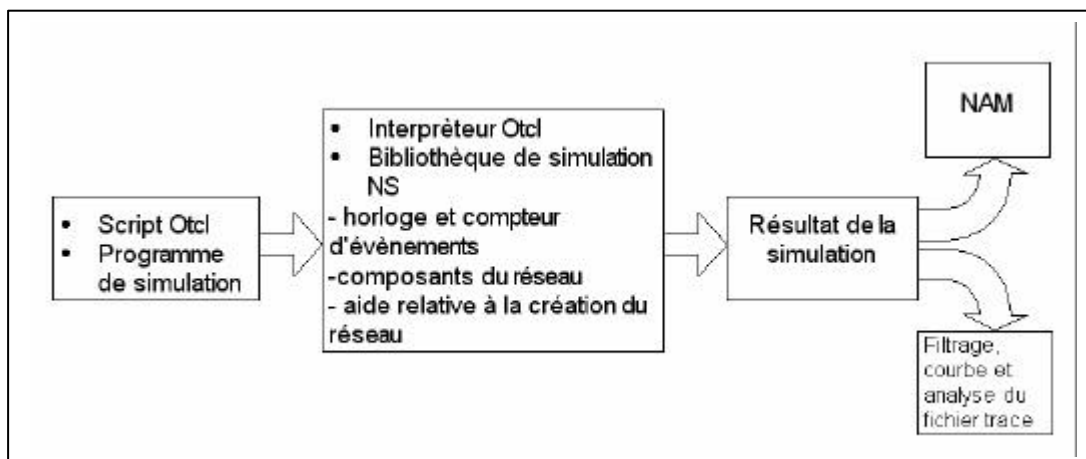


Figure A.1 schéma de l'utilisation de NS

L'utilisation de ces outils et l'exploitation des résultats requièrent certaines compétences: Connaissance du C++ et maîtrise des concepts de la programmation orientée objet pour l'édition des routines qui composent le simulateur.

- Connaissance du langage script OTCL pour la description des simulations et le retraitement des données.

- Maîtrise de l'outil indispensable à l'extraction des données des fichiers, AWK.
- Capacité à produire des graphes à partir des fichiers produits, à l'aide de Gnuplot par exemple.

Les concepteurs de NS-2 ne manquent pas de mettre en garde les utilisateurs sur l'aspect non achevé de NS-2. Il ne s'agit pas d'un produit fini, de nouveaux bogues sont souvent trouvés et d'autres constamment introduits par les utilisateurs programmeurs. Le logiciel est par essence en constante évolution de par la nature des réseaux et protocoles qui sont en perpétuelle évolution. Il sera donc demandé à l'utilisateur de vérifier la véracité de ses résultats et de s'assurer qu'ils ne sont pas erronés par une mauvaise implémentation de ses algorithmes. Il est donc désirable, même après cette démarche volontaire, que l'auteur d'une simulation n'oublie jamais d'y joindre le code dont elle est le fruit

## A.1 Pourquoi deux langages ?

Si NS-2 utilise deux langages c'est parce qu'il réalise deux types d'opérations. D'une part, le simulateur doit être efficace dans les simulations détaillées c'est à dire la manipulation des bytes, de paquets et d'en-têtes et dans l'application d'algorithmes sur de grandes quantités de données. Dans ce cas la vitesse d'exécution est primordiale et prend le pas sur le temps de compilation, l'utilisation du C++ s'impose. D'autre part, l'utilisateur souhaite changer rapidement ses scénarios de simulation, dans ce cas OTCL offre une bonne solution.

L'inconvénient de cette dualité est le dédoublement des objets et l'implémentation des fonctions qui doit veiller à la juste interaction entre les deux parties. Pour pouvoir être suivie par l'utilisateur, une variable doit exister à la fois en C++ et en OTCL. La figure suivante illustre cette notion de dualité.

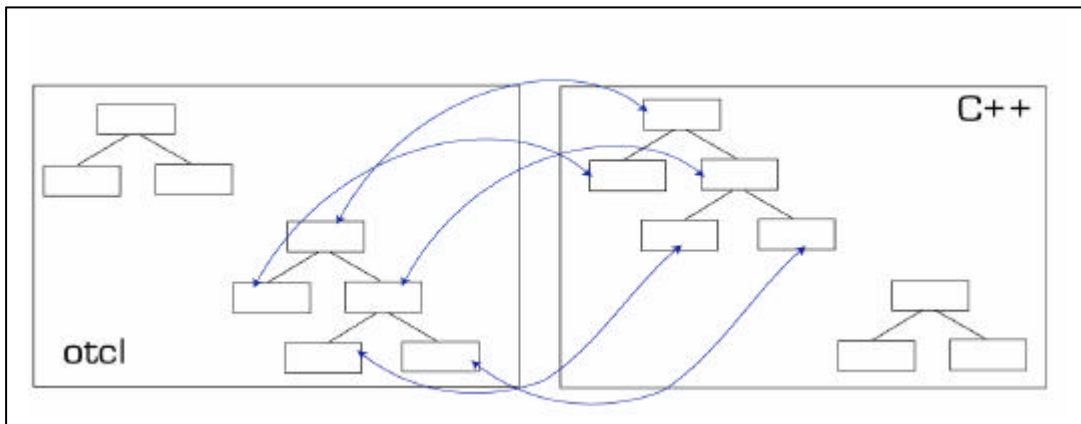


Figure A.2 Dualité entre OTCL et C++

## A.2 Le planificateur d'événements

C'est l'élément principal du simulateur. Il est implémenté à l'aide d'une liste chaînée des événements qui doivent se produire. Ces événements sont de plusieurs types : expiration d'un temporisateur (*timer*), arrivée d'un paquet, perte d'un paquet, démarrage d'une source, ...voici la définition d'un événement :

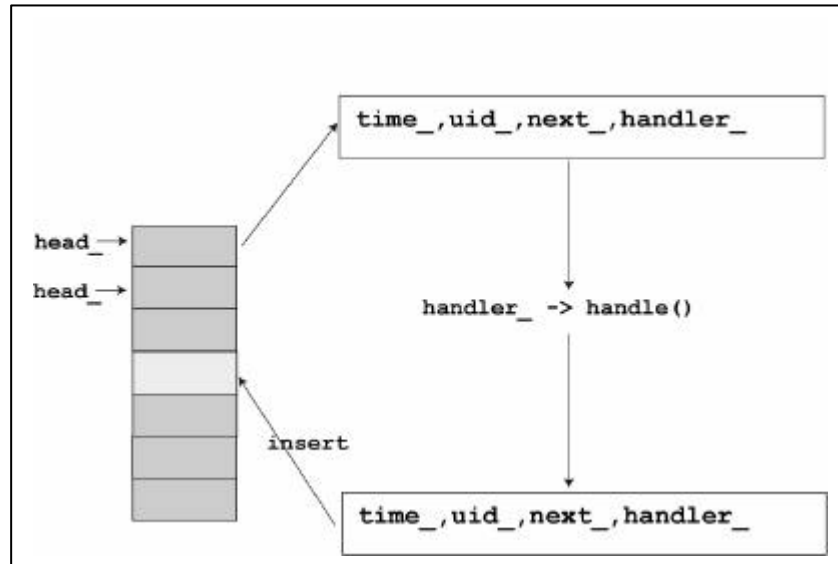
```

Class Event {
Public :
Event *next_ ; /*event list */
Handler * handler_ ; /* handler to call when event ready */
Double time_ ; /* time at which event is ready */
Int uid_ ; /* unique ID */
Event () : time_ (0), uid_ (0) {}
};

```

Un événement possède donc quatre éléments: un pointeur vers le prochain événement, un pointeur vers l'objet responsable du traitement de cet événement, l'instant où se produit l'événement et un identificateur unique pour chaque événement. La réalisation de l'événement se fait donc en appelant la fonction «handle» de l'objet responsable du traitement de l'événement. Les principaux utilisateurs du planificateur d'événement sont les composants du réseau (nœuds, liaisons,...) qui désirent simuler les délais de manipulation de paquets ou tout ce qui nécessite une synchronisation. Le déroulement du programme consiste à prendre systématiquement l'élément suivant de la liste du planificateur et à le traiter. La figure A3

illustre le traitement de l'événement qui se trouve en tête de la liste. Ce traitement peut provoquer l'insertion de nouveaux événements dans la liste.



**Figure A.3 Planificateur et traitement des événements**

NS-2 dispose de deux types de planificateurs. L'un en temps réel et l'autre en temps non réel. Le planificateur en temps non réel est le planificateur par défaut. Le planificateur en temps réel doit permettre au simulateur d'interagir avec un véritable réseau. Une importante utilisation du planificateur est la programmation d'événements tels que l'activation d'une application FTP, le lancement d'une procédure ou la fin de la simulation. Un tel événement est de la classe «AtEvent», une sous-classe de la classe «Event» qui dispose d'une variable supplémentaire pour contenir la chaîne de caractères passée en argument. Un tel événement est traité de la même manière que ceux relatifs au traitement des paquets. Cependant, lorsque cet événement arrive en tête du planificateur, il est traité et la commande OTCL spécifiée par la chaîne de caractères est exécutée. Voici un exemple de simulation faisant appel à ce genre d'événement. Après 12 secondes de simulation, la procédure «finish» sera appelée.

```

Set ns [new Simulator]
Proc finish {} {...}
$ns at 12 "finish"

```

Le planificateur pourra aussi être utilisé pour appeler à des intervalles réguliers une procédure. Dans l'exemple suivant la procédure «record» enregistre dans un fichier la valeur de la variable «cwnd» toutes les 0.1 secondes.

```

Proc record {}{
Global ns tcp fichier
Set time 0.1
Set now [$ns now]
Puts $fichier "$now [$tcp set cwnd]"
$ns at [expr $now+$time] "record"
$ns at 0.0 "record"
$ns run

```

### A.3 Composants des réseaux

La topologie NS-2 est essentiellement composée de noeuds et de liens. Pour mieux comprendre les différents composants du réseau dans NS, on va présenter la hiérarchie des classes OTCL de NS à l'aide de la figure suivante

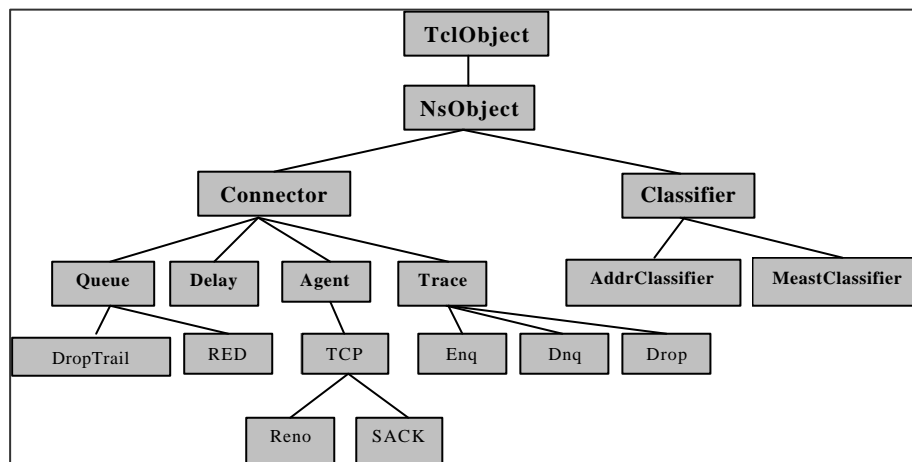


Figure A.3 les classes OTCL de NS

La racine de la hiérarchie est la classe TcObject qui est à la base de tous les objets OTCL de la librairie de NS (planificateur, composants réseau, objets NAM). La classe NsObject est une descendante de la classe TcObject et la base de tous les objets qui manipulent les paquets (noeuds, liaisons, files). Cette classe est encore subdivisée en deux sous-classes Connector et Classifier. Les objets offrant une seule sortie possible pour les paquets se trouvent dans la classe Connector et les autres dans la classe Classifier

### A.3.1 Nœud

Le nœud est un des premiers éléments de description de la topologie d'un réseau sous NS. Le nœud est composé de deux éléments, le classificateur d'adresse et le classificateur de port. Le rôle des classificateurs est de relayer les paquets sur la bonne liaison (*Addr Classifier*) ou sur le bon port (*Port Classifier*) si le paquet est destiné à un agent de ce nœud. La classification des paquets se fait sur la base de l'adresse de destination contenue dans le paquet. Cette adresse contient un champ qui détermine le nœud destinataire et un champ déterminant l'agent (le port).

Il existe deux types de nœuds: les nœuds "unicast" (figure A-5) et les nœuds "multicast" (figure A-6). Dans la structure multicast se trouve un replicator dont la fonction est de copier les paquets pour les acheminer vers plusieurs destinations.

La déclaration d'un nœud se fait de la manière suivante:

```

Set ns [new Simulator]
$ns node

```

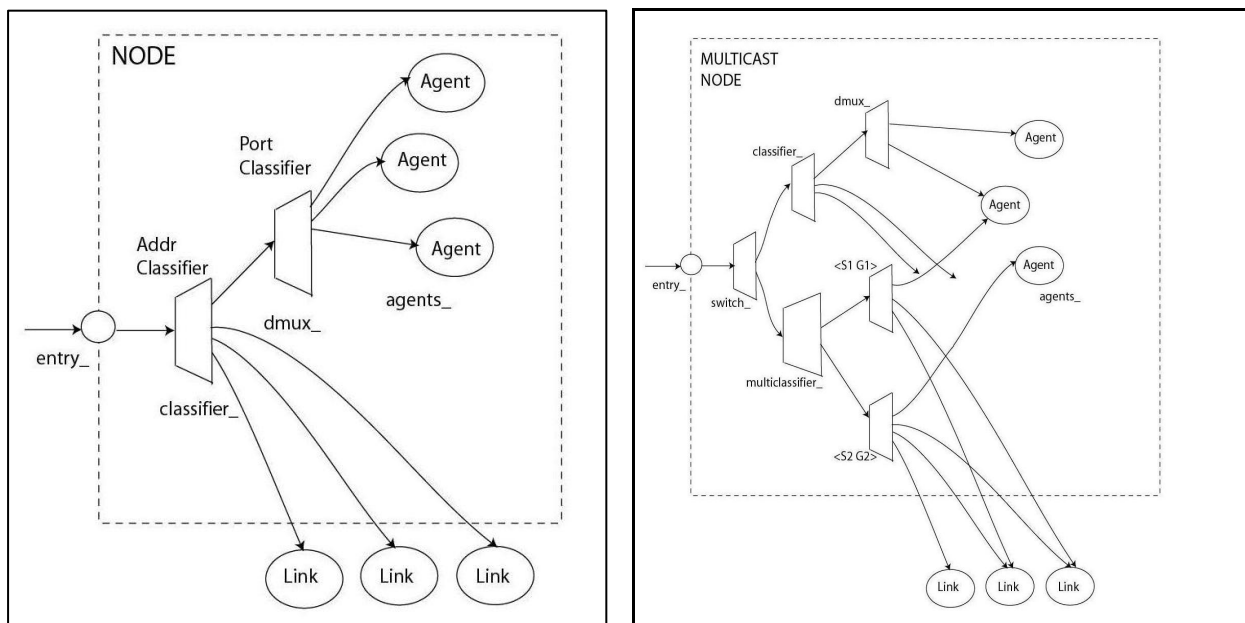


Figure A.4 Structure d'un nœud unicast et multicast

### A.3.2 Liens

Les liens servent à connecter les noeuds entre eux. Un lien est composé d'un ensemble de connecteurs comme présenté dans la figure A-7.

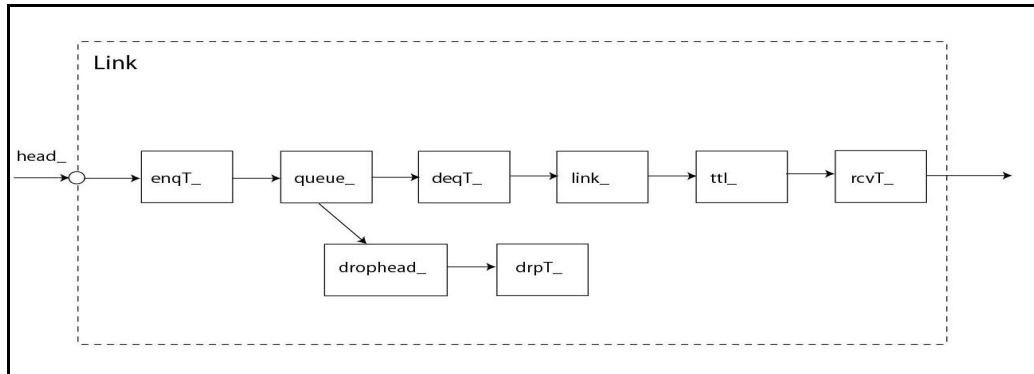


Figure A.5 Structure d'un lien dans NS

Chaque connecteur remplit un rôle différent, ils sont définis comme suit :

- Head point d'entrée du lien.
- Queue : référence à la file principale du lien. C'est où s'insère la file d'attente dont on aura choisi la discipline.
- TTL : référence à l'élément qui s'occupe du traitement du "time to live".
- DropHead : référence à l'objet qui traite les s paquets rejetés.
- Link : référence à l'élément qui modélise la ligne en terme de délai et de bande passante.
- enqT, deqT, drpT, rcvT : ces éléments ont pour rôle de suivre les paquets. Ils offrent la possibilité d'observer durant la simulation certains paramètres du lien (nombre de paquets reçus, nombre de paquets perdus,...).

La création d'un lien se fait à l'aide des commandes suivantes:

```

| $ns simplex-link <node0> <node1> <bandwith> <delay> <queuetype>
| $ns duplex-link <node0> <node1> <bandwith> <delay> <queuetype>

```

La première commande créera une liaison unidirectionnelle entre les noeuds node 0 et node 1 alors que la seconde créera une liaison bidirectionnelle. Les arguments que prennent ces commandes sont : **bandwith** la bande passante, **delay** le délai et **queue type** la discipline de la file d'attente implémentée sur ce lien.

### A.3.3 Files

La file est le lieu où les paquets sont stockés momentanément dans l'attente d'être transmis par le lien vers le noeud suivant. Le temps d'attente dépendra du niveau de congestion du lien. La gestion de cette file peut répondre à plusieurs politiques différentes. Voici celles implémentées dans NS-2 :

- DropTail : utilise la technique FIFO (First In First Out). C'est la file par défaut. Dès que la file est saturée, toutes les entrées supplémentaires seront automatiquement éliminées.
- RED : Random Early Detection.
- FQ: Fair Queuing.
- SFQ: Stochastic Fair Queuing.
- CBQ: Class Based Queuing.
- DRR: Deficit Round-Robin.

### A.3.4 Agents

On peut assimiler les agents de NS-2 à la couche transport du modèle OSI. Ce sont les agents qui construisent ou détruisent les paquets. Ce sont les points terminaux du réseau qui reçoivent ou délivrent les paquets des applications. A chaque agent est attribué un port. L'adresse d'un agent se compose du numéro de son noeud et de son port. Voyons comment créer un agent et l'attacher à un noeud. Voici la commande qui permet de créer un agent. Ci-dessous nous créons un agent "TCP source" et un agent "TCP récepteur".

```

| Set source [new Agent/TCP]
| Set destination [new Agent/TCPSink]

```

Ensuite ces agents doivent être attachés à un noeud, la source au noeud 1 et la destination au noeud2.

```

| $ns attach-agent $noeud1 $source
| $ns attach-agent $noeud2 $destination

```

Les agents peuvent ensuite être connectés entre eux.

```

| $ns connect $source $destination

```

### A.3.5 Applications

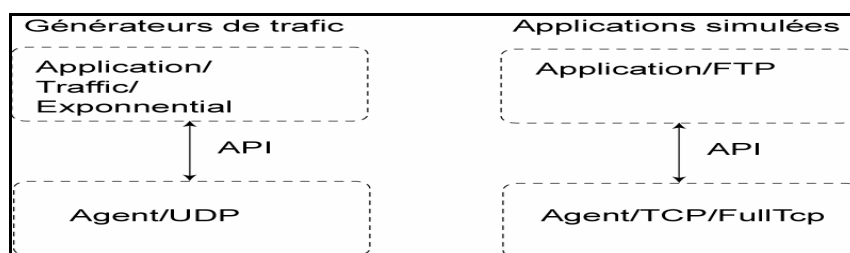
Au dessus de la couche transport se trouve l'application qui va générer le trafic. Il y aura au plus une application par agent.

Les applications que l'on peut utiliser au-dessus de TCP :

- **Application/FTP** : Application FTP qui génère autant de trafic que nécessaire pour l'agent TCP auquel il est attaché. L'application peut-être configurée pour n'envoyer qu'un nombre maximum de paquets.
- **Application/Telnet** : Les intervalles de temps entre l'émission de paquets sont pris exponentiellement avec une moyenne fixée par l'utilisateur.

Les applications que l'on peut utiliser au-dessus de UDP :

- **Application/Traffic/Exponential** : Source de trafic On/Off. En mode "on", des paquets sont envoyés a débits constant. En mode "off", aucun trafic n'est généré. Les moments d'émission et de silence sont tirés d'une distribution exponentielle. Ses paramètres de configuration sont les suivants: packet size pour la taille des paquets, burst time le temps moyen "on", idle time temps moyen "off" et rate débit durant les périodes "on".
- **Application/Traffic/Pareto** : Le trafic est généré selon une distribution de Pareto.
- **Application/Traffic/CBR** : Source de trafic constante. Ses paramètres de configuration sont les suivants: rate pour le débit, packet size la taille des paquets, maxpkts le nombre maximum de paquets à envoyer et random si l'on désire introduire du bruit dans les instants de départ.
- **Application/Traffic/Trace** : permet de déterminer le trafic généré à partir d'un fichier.

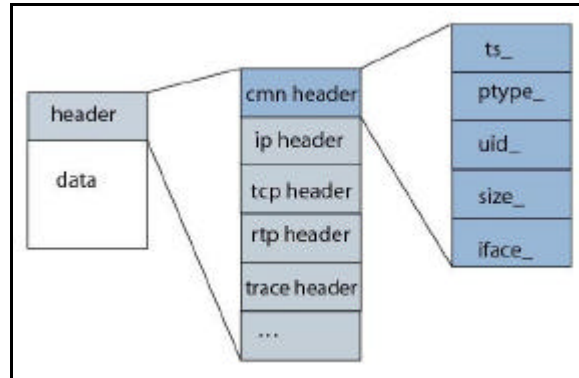


**Figure A.6 Applications pouvant tourner au-dessus de l'agent de transport**

### A.3.6 Paquets

Les paquets dans NS-2 sont constitués d'un empilement d'en-têtes et en option d'un pointeur vers des données. Chaque paquet comporte tous les en-têtes (voir figure A-9), même s'ils ne sont pas tous nécessaires. Cette technique entraîne un gaspillage de la mémoire mais simplifie

l'utilisation de l'en-tête. L'offset nécessaire pour accéder à chaque partie de l'en-tête est enregistré, ce qui permet à chaque agent d'accéder à la portion de l'en-tête qui l'intéresse.



**Figure 4.7 Format des paquets NS**

## A .4 Post simulation

### A.4 .1 Fichier trace

On peut demander à NS de récolter un certain nombre de données statistiques sur le déroulement de la simulation et de les sauvegarder dans un fichier «trace ». la commande qui permet l'obtention d'un tel fichier (appelé out.all) est la suivante :

```
Set trace [open out.all]
$ns trace-all $trace
Proc finish {} {
  Global trace
  # Close the file
  Close $trace
}
```

Ce fichier comportera tous les événements survenus lors de la simulation : l'arrivée d'un paquet à un nœud, départ d'un paquet d'un nœud, perte d'un paquet ou encore réception d'un paquet par un agent. Chaque paquet peut donc être suivi tout au long de son parcours. En traitant ce fichier de très grande taille on pourra en extraire l'information souhaitée : calcul du débit aux nœud, évaluation des pertes ... Chacune des lignes correspond à un événement survenu à un paquet.

L'étape à suivre est de transformer cet ensemble de données brutes en un sous-ensemble intéressant pour l'analyse et la compréhension de la simulation. Cela se fera par le traitement du fichier out.all avec un langage adéquat tel que AWK ou PERL.

#### A.4.2 Network Animator (NAM)

C'est un outil d'animation graphique basé sur Tcl/TK. Il permet de visualiser le déroulement d'une simulation en affichant la topologie du réseau et le déplacement des paquets. Son interface utilisateur est semblable à celui d'un lecteur CD et permet notamment le contrôle de la vitesse du déroulement de la simulation. De plus il offre aussi une représentation graphique d'un certain nombre de grandeurs d'intérêt comme le débit et les pertes de paquets sur les liens. Voici les commandes du script qui permettent de demander au simulateur la création d'un fichier de visualisation out.nam et le lancement de l'animation sur NAM

```
Set trace [open out.nam]
$ns namtrace-all $trace-nam
Proc finish {} {
  Global trace-nam
  #close the file
  Close $trace-nam
  # execute NAM on the trace-nam file
  exec nam out.nam &
}
```