



**HAL**  
open science

# Contribution à l'étude et à l'amélioration de la qualité de service sur un réseau maillé WiFi

Rémi Soulage

► **To cite this version:**

Rémi Soulage. Contribution à l'étude et à l'amélioration de la qualité de service sur un réseau maillé WiFi. Automatique. Université Montpellier II - Sciences et Techniques du Languedoc, 2011. Français. NNT: . tel-00750468

**HAL Id: tel-00750468**

**<https://theses.hal.science/tel-00750468>**

Submitted on 12 Nov 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ MONTPELLIER 2  
SCIENCE ET TECHNIQUE DU LANGUEDOC

---

THÈSE

---

En vue de l'obtention du grade de

**DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER 2**

Formation Doctorale : **SYSTÈMES AUTOMATIQUES ET MICROÉLECTRONIQUES**

Ecole Doctorale : **INFORMATION, STRUCTURES, SYSTÈMES**

Présentée et soutenue par

**Rémi SOULAGE**

jeudi 1 décembre 2011

---

---

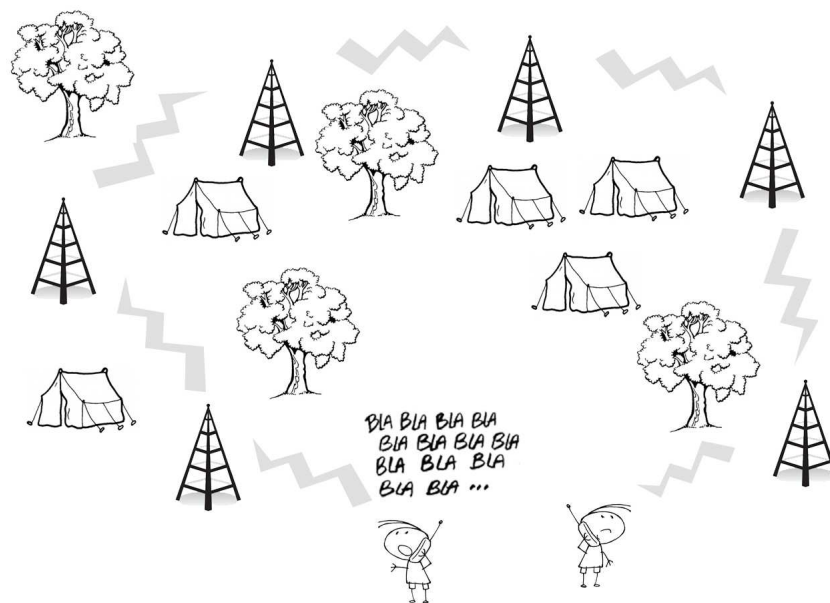
**CONTRIBUTION À L'ÉTUDE ET À L'AMÉLIORATION DE  
LA QUALITÉ DE SERVICE SUR UN RÉSEAU MAILLÉ WIFI**

---

**Jury**

<b>David ANDREU</b> Maître de Conférences, Université Montpellier 2, LIRMM	Co-directeur de thèse
<b>Yves BOULOT</b> Osmozis	Invité
<b>Philippe FRAISSE</b> Professeur, Université Montpellier 2, LIRMM	Directeur de thèse
<b>Michel MISSON</b> Professeur, Université Clermont-Ferrand	Rapporteur
<b>Gérard TREMBLAY</b> Docteur, Osmozis	Examineur
<b>Thierry VAL</b> Professeur, Université de Toulouse	Rapporteur
<b>René ZAPATA</b> Professeur, Université Montpellier 2, LIRMM	Président





*A ma femme, à ma famille...*



Cette thèse est le fruit d'une collaboration CIFRE entre :



UMR 5506 - CC477  
161, rue Ada  
34392 Montpellier Cedex 5



ZA Les Baronnes  
34730 Prades-le-Lez



# Remerciements

*Ces travaux ont été encadrés par le LIRMM, Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier, Unité mixte de recherche CNRS et de l'Université Montpellier II. Je remercie la direction de ce laboratoire de m'avoir permis de faire ces travaux de recherche.*

*Je remercie mes rapporteurs Michel MISSION et Thierry VAL d'avoir accepté ce rôle et de venir respectivement de Clermont-Ferrand et de Toulouse. Je les remercie aussi pour l'organisation des rencontres L2I qui m'ont permis de présenter les évolutions de ce travail de recherche, mais aussi d'apprendre à partir des discussions et des travaux des autres participants.*

*J'adresse mes sincères remerciements à mon directeur de thèse Philippe FRAISSE, que j'ai rencontré il y a onze ans, cuisinant une paëlla à la Salvetat-sur-Agout. Merci de m'avoir accompagné durant mon DUT, encadré pour le projet d'aspirateur autonome, trouvé un stage de chronométrage chez qui j'ai travaillé par la suite, fait la publicité du DEA Robotique en boîte de nuit, et enfin amené à rencontrer l'entreprise Osmozis qui m'a permis ce travail de thèse et pour qui je travaille aujourd'hui. Un grand merci pour ce long encadrement et ces encouragements notamment dans la dernière ligne droite de rédaction.*

*J'exprime ma gratitude à mon co-encadrant David ANDREU qui m'a fait découvrir les réseaux de Petri, en cours, en stage et enfin en thèse! Merci pour ces multiples réunions qui ont largement fait avancer ce travail, pour ces lectures, relectures, rerelectures et corrections, ainsi que pour tous ces conseils. Merci d'avoir toujours été présent pour m'aider au niveau scientifique mais aussi administratif.*

*Merci à René ZAPATA d'avoir accepté de faire partie de ce jury et d'évaluer ce travail. Finalement, les cours d'automatique de DEA m'ont été utiles dans ce travail de recherche.*

*Je ne saurais assez remercier Gérard TREMBLAY, Directeur Général d'Osmozis, docteur en hyperfréquence (si, si) et Monsieur le Président Yves BOULOT, grand théoricien et prévisionniste, pour m'avoir offert ce poste de premier employé de la société, notamment de premier installateur avec une échelle en bois sur la place de la Comédie. Cette aventure m'a permis de faire cette thèse mais aussi de découvrir le fonctionnement*



*et la gestion d'une société à partir de sa création jusqu'à aujourd'hui. Elle m'a aussi permis de parcourir la France pour goûter à la daube de marcassin, visiter le Croisic et les Prairies du Lac... Je remercie aussi tous les employés : Christophe papa du hotspot, Cyril et Guilhem installateurs d'hotspots, Yoël acheteur de pièce d'hotspot, PH, David, Simone... et tous les nombreux autres, pour leurs bonnes humeurs et pour ce travail en osmose dans une bonne ambiance.*

*Je souhaite également remercier Olivier PARODI qui a commencé sa thèse en même temps que moi, mais l'a terminée bien avant, pour tous les bons moments passés ensemble, et pour les quelques pauses à la cafétéria du labo. Merci pour ton manuscrit à partir duquel je m'inspire de l'organisation.*

*J'adresse de sincères remerciements à la cousine Anne-Claire ainsi qu'à Laurent pour la relecture et la correction de cette thèse, comme quoi il n'est pas nécessaire de comprendre pour vérifier l'orthographe et la grammaire.*

*Merci à mes amis et cousins de Ceilhes : Benjamin, Fabrice, Marco, Popo, les membres du foyer ainsi que tous les autres pour les quelques fêtes cet été qui m'ont permis de me reposer et d'être plus performant sur la rédaction de ce manuscrit.*

*Une pensée va naturellement à une bande de Chacal : le Catalan, le Def, le Brousse, le Yass, Docteur Lole, le Rrray ainsi qu'à leurs femmes pour ceux qui en ont, avec qui j'ai passé de longues années d'études et de très bons moments sur Montpellier.*

*Je ne saurais terminer sans rendre hommage à mes parents, à mes frères, à ma famille et ma belle famille pour leurs soutiens, ainsi qu'à Claire, ma femme, pour sa patience et sa gentillesse, sans oublier notre petit Théo qui me laisse terminer ces quelques lignes avant de se réveiller.*

# Table des matières

<b>Remerciements</b>	<b>vii</b>
<b>Table des matières</b>	<b>ix</b>
<b>Introduction</b>	<b>1</b>
<b>1 Le WiFi maillé, des principes au terrain</b>	<b>5</b>
1.1 Pourquoi le WiFi maillé?	5
1.1.1 Les éléments d'un réseau WiFi	6
1.1.2 Topologie	7
1.2 La communication	9
1.2.1 Adresses des nœuds contenus dans un paquet	9
1.2.2 Choix des chemins	9
1.3 L'accès à la couche physique	10
1.3.1 Présentation de la méthode d'accès	10
1.3.2 Les paquets de contrôle RTS/CTS	12
1.3.3 La qualité de service (QoS)	13
1.4 Evaluation de performance, notions de métrologie	13
1.4.1 Introduction	13
1.4.2 Le débit	13
1.4.3 La latence	14
1.4.4 La gigue	14
1.4.5 La qualité de service	14
1.4.6 Le temps d'utilisation du canal WiFi	15
1.4.7 Mesure des ces indicateurs	15
1.5 Analyse par expérimentation	15
1.5.1 Introduction	15
1.5.2 Présentation de l'expérience	15
1.5.3 Résultats	17
1.5.4 Problèmes observés	22
1.5.5 Conclusion	23
<b>2 Etat de l'art</b>	<b>25</b>
2.1 Solutions orientées couche physique	25
2.1.1 Le WiFi maillé sur plusieurs canaux	25

2.1.2	Changement de modulation basé sur des chaînes de Markov . . .	27
2.2	Solutions protocolaires pour le WiFi maillé . . . . .	27
2.2.1	Approches visant l'équilibrage de charge par les point d'accès . .	27
2.2.2	Approche traitant du problème du nœud caché . . . . .	27
2.2.3	Approches visant à améliorer la qualité de service de certains flux	32
2.3	Positionnement . . . . .	33
2.3.1	Identification des problèmes . . . . .	33
2.3.2	Identification des leviers . . . . .	34
2.3.3	Problème de l'expérimentation . . . . .	35
2.3.4	Simulation . . . . .	35
2.4	Simulateurs existants . . . . .	36
2.4.1	Simulateurs réseaux classiques . . . . .	36
2.4.2	Choix du formalisme de modélisation . . . . .	38
2.4.3	Joueurs de réseaux de Petri . . . . .	39
2.5	Modélisation et Réseaux de Petri . . . . .	40
2.6	Conclusion . . . . .	41
<b>3</b>	<b>De la modélisation à la validation du simulateur</b>	<b>43</b>
3.1	Introduction . . . . .	43
3.1.1	Objectif de la modélisation . . . . .	44
3.1.2	Présentation du formalisme de modélisation . . . . .	44
3.2	Structuration du modèle . . . . .	47
3.2.1	Une approche modulaire . . . . .	47
3.2.2	Présentation des modules . . . . .	49
3.3	Granularité . . . . .	51
3.3.1	Granularité au niveau de la couche radio . . . . .	51
3.3.2	Granularité au niveau de la couche liaison de données . . . . .	51
3.3.3	Granularité au niveau supérieur à la couche liaison de données .	52
3.4	Description . . . . .	52
3.4.1	Configuration de la topologie d'une installation WiFi maillé . . .	52
3.4.2	Modèle <i>Network</i> . . . . .	56
3.4.3	Modèle <i>Radio</i> . . . . .	59
3.4.4	Modèle <i>MAC</i> . . . . .	65
3.4.5	Modèle <i>LLC</i> . . . . .	94
3.4.6	Modèle <i>UDP</i> . . . . .	98
3.4.7	Modèle <i>ICMP</i> . . . . .	100
3.5	Validation . . . . .	103
3.5.1	Énumération des mécanismes à valider . . . . .	103
3.5.2	Présentation du cas d'étude . . . . .	107
3.5.3	Graphiques générés . . . . .	108
3.5.4	Simulation d'une requête ICMP sur le cas d'étude . . . . .	110
3.5.5	Simulation de requêtes ICMP en concurrence avec des requêtes UDP . . . . .	113
3.5.6	Utilisation du mécanisme de contrôles RTS/CTS . . . . .	117
3.5.7	Simulation d'émission de Beacon . . . . .	118
3.5.8	Conclusion . . . . .	119

<b>4</b>	<b>Exploitation du simulateur</b>	<b>121</b>
4.1	Introduction . . . . .	121
4.1.1	Synthèse des leviers disponibles . . . . .	121
4.1.2	Rappel des problèmes évoqués dans l'état de l'art . . . . .	122
4.2	Simulation dans une configuration de référence . . . . .	122
4.2.1	Présentation de la simulation . . . . .	122
4.2.2	Résultats . . . . .	123
4.3	Ralentissement de l'injection de flux au niveau de la source du trafic . . . . .	128
4.3.1	Rappel du rôle des paramètres $AIFS$ , $CW_{min}$ et $CW_{max}$ . . . . .	128
4.3.2	Calcul de probabilité des gains de période de contention . . . . .	129
4.3.3	Application de ces paramètres sur le simulateur . . . . .	130
4.3.4	Bilan des actions sur l'injection de trafic . . . . .	138
4.4	Ralentissement de l'injection du flux au niveau de la source et du premier relais . . . . .	139
4.4.1	Augmentation des paramètres $AIFS$ sur les nœud 1 et 2 pour un flux du nœud 1 à 4 . . . . .	140
4.4.2	Augmentation des paramètres $AIFS$ sur les nœud 1 et 2 pour un flux du nœud 1 à 6 . . . . .	142
4.4.3	Bilan de l'impact de cet action . . . . .	144
4.5	Régulation du trafic par l'utilisation du levier $AIFS$ . . . . .	144
4.5.1	Principe du mécanisme de régulation local . . . . .	145
4.5.2	Identification des paramètres . . . . .	146
4.5.3	Résultats avec le régulateur pour un flux du nœud 1 à 4 . . . . .	147
4.5.4	Résultats avec le régulateur pour un flux du nœud 1 à 6 . . . . .	151
4.5.5	Proposition d'un mécanisme de régulation global . . . . .	153
4.6	Conclusion . . . . .	154
	<b>Conclusion</b>	<b>157</b>
	<b>Bibliographie</b>	<b>164</b>
	<b>Table des figures</b>	<b>165</b>
	<b>Liste des tableaux</b>	<b>169</b>
	<b>Glossaire</b>	<b>171</b>
	<b>Résumé</b>	<b>176</b>



# Introduction

## Contexte

Le WiFi est une certification délivrée par la WiFi Alliance [All11]. Ce consortium assure le respect de la norme de réseau sans fil, ratifiée il y a une dizaine d'années par IEEE sous la référence 802.11. Cet ensemble de protocoles de communication est présent aujourd'hui sur de nombreux équipements informatiques. Il est grandement exploité comme réseau d'entreprise ou domestique afin d'éviter de déployer des câbles. Cette technologie se met aussi en place dans les lieux publics (aéroports, gares, hôtels...) afin de permettre aux utilisateurs d'ordinateurs portables d'accéder à Internet lorsqu'ils sont en déplacement. Dans ce cas, le service WiFi est apporté par des hotspots auxquels l'accès est souvent payant. De plus, des téléphones mobiles WiFi permettent de transporter de la voix sur Internet à moindre coût.

Pour des applications à temps critique telles que la vidéo ou la téléphonie, une qualité de service doit être assurée. C'est pourquoi une annexe nommée 802.11e a été ajoutée en 2005. Des études ont montré l'efficacité de ces nouvelles fonctionnalités sur des communications audio et vidéo.

Osmozis, société innovante créée fin 2005, est opérateur de télécommunication. Son objectif est le déploiement d'accès Internet sans fil sur de grandes surfaces, telles que les campings, résidences de vacances, ports de plaisance ou parcs d'exposition. Cette société a déjà installé et exploite plusieurs milliers d'équipements en service permettant notamment aux estivants, en vacances dans un mobilhome, qui n'ont que la télévision, la piscine, la plage et le soleil, d'avoir un accès Internet.

Cependant, l'utilisation classique du WiFi se limite à une centaine de mètres en extérieur ou une cinquantaine de mètres en intérieur autour d'un émetteur central appelé point d'accès. Ces éléments peuvent être connectés entre eux en filaire afin d'étendre le réseau. Mais cette contrainte nécessite de creuser des tranchées pour les connecter en réseau alors que ces grands espaces vont jusqu'à plusieurs dizaines d'hectares.

Dans ce contexte, le WiFi maillé est utilisé. En effet, il permet d'établir les liens entre chaque point d'accès en WiFi afin que chacun d'eux relaye les paquets jusqu'à leurs destinations. Cette annexe nommée 802.11s est en cours de normalisation. Dans ce cas, la qualité de service est affectée lors des multiples relais de paquets par chaque point d'accès du chemin déterminé car une contrainte établie par la société Osmozis est que chaque hotspot n'ait qu'une interface WiFi et communique donc sur le même canal. Il est donc nécessaire d'améliorer la qualité de service des transmissions, surtout quand il s'agit de communications audio ou vidéo. Pour ceci, les propositions d'amélioration

doivent pouvoir être appliquées sur le matériel utilisé. C'est pourquoi les leviers sur lesquels nous jouerons doivent être concrets.

Lors d'expérimentations, il est aisé de se rendre compte de la baisse de la qualité de service, mais la mise en évidence des problèmes de communication nécessite l'enregistrement et l'analyse des transmissions radio au niveau de tous les nœuds qui sont parfois éloignés. De plus, ces enregistrements doivent être synchronisés temporellement. Enfin, l'environnement peut varier et comparer des résultats à des moments différents peut laisser un doute sur un changement d'état de celui-ci. C'est pourquoi nous optons pour un processus de recherche "Modélisation, Simulation, Amélioration". Les outils existants n'étant pas adéquats, nous proposons un nouveau simulateur, utilisant les réseaux de Petri comme formalisme de modélisation et de simulation. L'évolutivité de celui-ci permet des raffinements successifs en fonction des points sur lesquels l'étude doit être approfondie.

## Problématiques scientifiques visées

Les questions principales abordées lors de ce travail de recherche concernent les causes de cette diminution de qualité de service sur un réseau WiFi maillé, et les parades que nous pouvons mettre en place pour éviter ces détériorations. Pour répondre à la première question, il est nécessaire de s'interroger sur la méthodologie permettant d'observer en détail ces causes, et de trouver quels outils peuvent nous aider dans cette recherche.

Ensuite, lorsque ces problèmes sont identifiés, la recherche de leviers concrets, modifiables sur un matériel réel, est une étape importante. La compréhension de l'effet obtenu en jouant sur ces leviers est une autre problématique qui amène à se demander s'il existe une action automatique à effectuer sur ces leviers.

Enfin, savoir si les actions portées sur les leviers sont efficaces sur d'autres topologies ou avec d'autres types de flux amène à étudier si ces actions sont généralisables dans différents contextes.

## Objectifs

Ce manuscrit a tout d'abord pour objectif de décrire les réseaux WiFi maillés avec les éléments à améliorer. Ceci est présenté dans une première partie. Après une description du fonctionnement, une section métrologie permet d'exposer les différentes mesures qui reflètent la qualité de service. Ce chapitre se termine par une approche expérimentale qui démontre d'une part l'altération de la qualité de service causée par ce type de réseau, mais aussi la difficulté de recherche et de compréhension des causes de cette altération en utilisant des prototypes.

Ensuite, la deuxième partie propose un état de l'art qui permet dans un premier temps de faire une synthèse des problèmes rencontrés dans la littérature et d'étudier la faisabilité de leurs solutions par rapport à nos contraintes. Ainsi, à partir de cette étude, nous nous positionnons en exprimant les problèmes pris en compte et les leviers disponibles. Une recherche de simulateurs est exposée pour dresser une synthèse de l'existant. Au regard de ce bilan et de nos choix, la mise au point d'un nouveau simulateur basé

sur un modèle formel évolutif permet de répondre à nos besoins. Elle est suivie d'une recherche de joueurs de réseau de Petri, formalisme de modélisation qui nous avons choisi. Enfin, une dernière étude porte sur la littérature où le réseau de Petri est utilisé pour modéliser une communication WiFi.

Un des objectifs de ce manuscrit est la description du modèle de réseau WiFi maillé. Celui-ci est présenté dans la troisième partie qui se termine par une validation de ce modèle.

Enfin, la quatrième partie présente la première analyse, suivie d'une première amélioration proposée à l'aide de ce simulateur. En observant en détail les files d'attente des nœuds intermédiaires, nous remarquons que la saturation et l'encombrement de celles-ci provoquent de fortes baisses de qualité de service. Une approche automatique et réseaux permet alors de palier à ce problème.





# Chapitre 1

## Le WiFi maillé, des principes au terrain

### 1.1 Pourquoi le WiFi maillé ?

La norme IEEE 802.11 [IEE99a], correspondante au WiFi signifiant Wireless Fidelity, a été proposée en 1997 afin de mettre en place un réseau sans fil sur quelques dizaines ou centaines de mètres. Elle se situe au niveau des couches physique et liaison du modèle OSI. Par-dessus, la pile TCP/IP peut être utilisée et peut ouvrir les portes à l'Internet. Depuis, de nombreuses annexes ont été ajoutées à cette norme. Certaines définissent la couche physique 802.11a [IEE99b] 802.11b [IEE99c] 802.11g [IEE03] 802.11n [IEE09] afin d'utiliser différentes fréquences ou d'augmenter le débit. D'autres réglementent l'utilisation des bandes de fréquences en fonction des pays. Certaines définissent de nouveaux mécanismes d'accès au médium comme 802.11e [IEE05] qui améliore la qualité de service.

Aujourd'hui, ce standard est utilisé dans une très large majorité d'ordinateurs portables, certains téléphones mobiles ainsi que certaines consoles de jeux. Tous ces équipements peuvent se connecter au réseau WiFi de l'entreprise ou à celui des box ADSL domestiques.

Afin d'étendre la couverture du réseau sans fil, une idée serait d'augmenter la puissance d'émission. Cependant, celle-ci est limitée par les autorités, en particulier pour des contraintes sanitaires en fonction de la fréquence et du pays. De plus, certains équipements terminaux comme les téléphones mobiles ont une puissance d'émission plus faible pour limiter le coût et la consommation énergétique. Une autre possibilité serait d'augmenter le nombre de points d'accès et de les répartir sur la zone à couvrir. Pour assurer une couverture totale en extérieur, la distance entre deux points d'accès peut dépasser la centaine de mètres. Il devient alors difficile de relier les éléments en filaire sur de telles distances comme le prévoit la norme 802.11.

Le WiFi maillé, bientôt normalisé par l'annexe 802.11s [IEE11], propose une solution en permettant aux points d'accès de communiquer entre eux sans fil. Ceux-ci pourront répéter les paquets de proche en proche afin d'atteindre une destination qui pourra être à l'autre bout du réseau. Ainsi, la superficie pouvant être couverte par un tel réseau est

beaucoup plus grande.

### 1.1.1 Les éléments d'un réseau WiFi

L'élément de base d'un réseau WiFi en mode infrastructure est le point d'accès (Access Point AP). Il peut-être connecté à un réseau Ethernet, ou être intégré dans un routeur. C'est lui qui relaye tous les paquets entre les différentes stations et le réseau auquel il est connecté (par exemple Ethernet). Le service fourni par un AP est appelé Basic Service Set (BSS) comme présenté dans la figure 1.1. Une Station (STA) peut se connecter à un AP. Celui-ci va répéter les paquets qui lui sont transmis afin de les acheminer jusqu'à leur destinataire qui peut être une autre STA, ou se situer sur le réseau Ethernet sur lequel est connecté l'AP. Le système de distribution (Distribution System DS) permet d'acheminer les données entre AP. Il peut être de type Ethernet et peut être connecté à Internet. Une STA peut communiquer avec d'autres STA en mode Ad-hoc, sans AP. Dans ce mode, chaque STA peut communiquer directement avec les autres, et le service est nommé Independent BSS (IBSS).

Le service fourni par tous les AP connectés sur le même DS est appelé Extended Service Set (ESS). Il peut couvrir des zones importantes. Un ESS est identifié par son nom (Service Set Identifier SSID). Il est émis par tous les AP qui font partie du ESS. Il pourra être visible par les stations afin de leur permettre de se connecter. Les AP sont distingués par des identifiants uniques appelés Basic Service Set Identifier (BSSID). Formé de 48 bits, c'est généralement l'adresse MAC de la carte WiFi de l'AP.

La norme 802.11 définit un ensemble de fonctions nommé Medium Access Coordination Function (MCF) qui décrit comment accéder au système de communication.

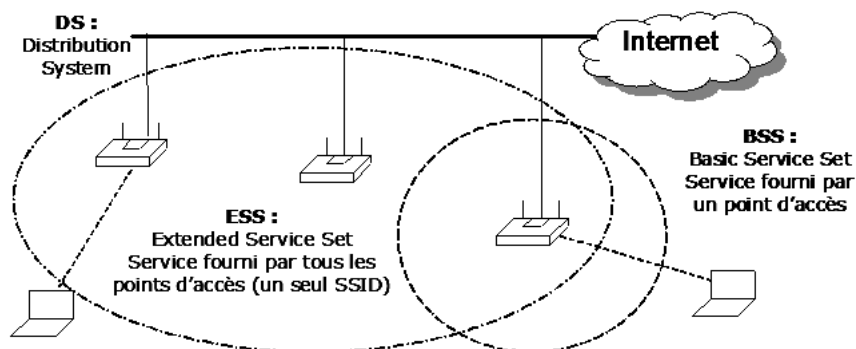


FIGURE 1.1 – Système de distribution WiFi proposé par IEEE 802.11

De nouvelles définitions permettent de présenter l'annexe 802.11s. L'élément de base du réseau WiFi maillé est le MP (Mesh Point). Il doit implémenter au minimum les fonctionnalités de recherche de voisin, de sélection de canal et créer des associations avec ses voisins afin de relayer des informations.

Plusieurs MP forment le WDS (Wireless Distribution System). C'est le système d'acheminement de la donnée entre tous les éléments du réseau maillé. Contrairement au DS de la norme 802.11, celui-ci est maintenant sans fil.

Un MP qui implémente la fonctionnalité de AP est appelé MAP (Mesh Access Point). Celui-ci va permettre à des STA qui respectent la norme 802.11 de base de se connecter au réseau maillé.

Le MPP (Mesh Portal) est un MP qui offre la fonctionnalité de se connecter sur un autre type de réseau, par exemple un réseau filaire qui conduit à Internet. La norme 802.11s définit un WDS avec une suite de nouveaux protocoles qui permettront la communication entre les éléments d'un réseau maillé. Ceux-ci devront apprendre la topologie du réseau de manière à déterminer automatiquement les chemins entre chaque élément. Deux procédures d'initialisation sont définies. D'une part, la STA va s'associer avec un MAP (802.11 traditionnel) ; et d'autre part, le MP va s'associer avec ses voisins.

### 1.1.2 Topologie

La norme 802.11 définit deux topologies : le mode infrastructure, présenté figure 1.2, où un AP joue le rôle d'élément central entre les STA afin de synchroniser toutes les stations, et le mode ad-hoc, présenté figure 1.3, où chaque STA dialogue directement avec les autres.

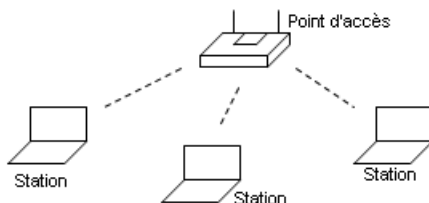


FIGURE 1.2 – Mode infrastructure

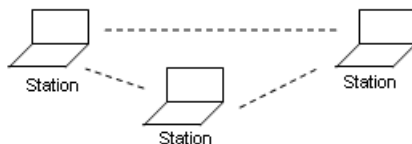


FIGURE 1.3 – Mode ad-hoc

Avec du WiFi maillé [HDM<sup>+</sup>10], chaque MAP permet à la fois de fournir un accès au réseau pour une station comme le ferait un point d'accès standard, mais aussi de se connecter à d'autres points d'accès. Deux points d'accès peuvent établir un lien WDS s'ils peuvent communiquer directement entre eux. Pour ceci, la puissance du signal WiFi, dépendante entre autre de la distance, doit permettre à des paquets d'être transmis d'un point d'accès à un autre, avec un nombre limité de paquets erronés ou perdus. Le débit physique utilisé sur ce lien est lui aussi dépendant de la qualité du lien.

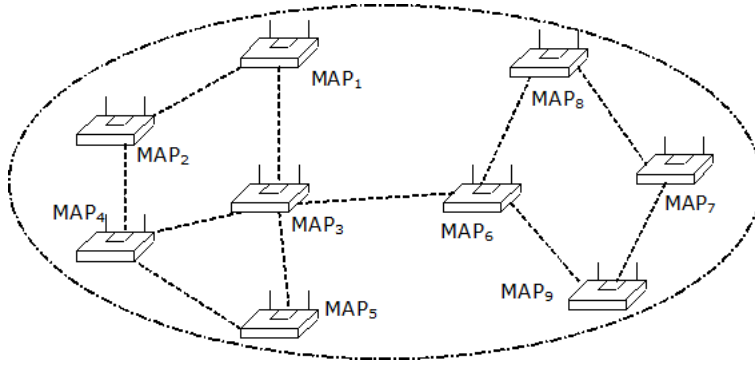


FIGURE 1.4 – Exemple de réseau maillé

Comme présenté sur la figure 1.4, de proche en proche, tous les points d'accès peuvent être connectés au même réseau WiFi maillé. De cette manière, si une station peut se connecter à un MAP, elle pourra communiquer avec tout le réseau. Si un MAP est connecté à Internet, la station le pourra également.

Lorsqu'un nœud émet une trame, celle-ci ne sera reçue que par les nœuds étant dans une zone à proximité. En effet, la possibilité de réception dépend du rapport signal sur bruit. La puissance du signal WiFi diminue en fonction de la distance et de la présence d'obstacle ; et le bruit dépend de l'environnement dans lequel d'autres appareils peuvent utiliser le canal radio.

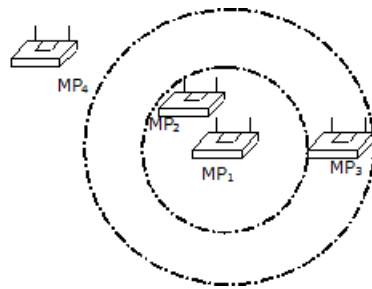


FIGURE 1.5 – Exemple de zone de réception

Par exemple sur la figure 1.5, lorsque le nœud  $MP_1$  émet, le nœud  $MP_2$  peut avoir une probabilité de 95% de recevoir correctement le paquet, alors que le nœud  $MP_3$  peut avoir une probabilité de 40%, et le nœud  $MP_4$  ne détectera jamais de paquet de ce nœud. Le nœud émetteur devra choisir un débit physique adapté au lien. Dans cet exemple, si le paquet est destiné à  $MP_2$ ,  $MP_1$  pourra choisir un débit physique de  $54 \text{ Mbit.s}^{-1}$ , ou un débit plus faible s'il est destiné à  $MP_3$ .

Pour définir une topologie, nous devons donc considérer d'une part si deux nœuds sont à portée, et d'autre part, s'ils sont à portée, une probabilité de succès de transmission ainsi qu'un débit physique. De plus, il est possible dans des cas particuliers qu'une transmission ait une probabilité de succès et un débit physique d'un nœud  $MP_1$  à un nœud  $MP_2$ , et une autre probabilité et un autre débit du nœud  $MP_2$  vers le nœud  $MP_1$ . En effet, une asymétrie peut être due à un environnement bruité.

## 1.2 La communication

### 1.2.1 Adresses des nœuds contenus dans un paquet

Lorsqu'un paquet transite entre une station et un point d'accès, trois champs d'adresse sont utilisés dans l'entête du paquet : BSSID, nœud source, nœud destination.

Afin d'être relayé, un paquet qui transite sur le réseau WiFi maillé contient 4 adresses MAC identifiant les acteurs de ce relais de paquet. Le nœud source est celui qui est à l'origine de l'introduction de ce paquet dans le réseau maillé WiFi, c'est l'émetteur d'origine. Le transmetteur est celui qui est en train d'émettre le paquet (donc potentiellement un nœud intermédiaire). Le receveur (aussi potentiellement intermédiaire) est le nœud qui doit recevoir le paquet. Enfin, le destinataire est le dernier récepteur du paquet.

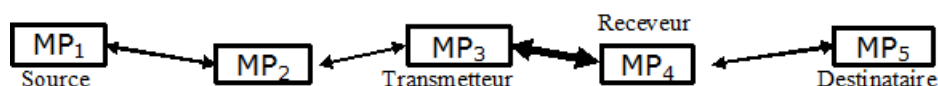


FIGURE 1.6 – Désignation des nœuds

Dans l'exemple exposé figure 1.6 où un paquet allant de  $MP_1$  à  $MP_5$  est en train d'être émis par  $MP_3$  et d'être reçu par  $MP_4$ , la source est  $MP_1$ , le transmetteur courant est  $MP_3$ , le receveur courant est  $MP_4$ , et le destinataire est  $MP_5$ .

### 1.2.2 Choix des chemins

Dans un réseau maillé, il peut y avoir plusieurs possibilités de chemins entre deux nœuds. Ce choix ressemble au choix de routage dans un réseau IP avec des boucles. Le terme "route" étant utilisé au niveau de la couche réseau, nous utiliserons le terme "chemin" pour spécifier par quels nœuds sont relayés des paquets quand la réémission se fait au niveau de la couche liaison.

De nombreux algorithmes de choix de chemin existent. Des protocoles sont proactifs comme DSDV (Destination Sequenced Distance Vector) [URKRZ<sup>+</sup>08] ou OLSR (Optimized Link State Routing) [DRFM08]. Ceux-ci sont adaptés à des petits réseaux. D'autres protocoles sont réactifs comme DSR (Dynamic Source Routing) [RD09] ou AODV (Ad Hoc On-Demand Vector) [SL09]. Ceux-là sont adaptés à des grands réseaux. En effet, un tel algorithme doit pouvoir reconfigurer les chemins si un nœud tombe en panne ou si la qualité d'un lien est modifiée.

La sélection de la méthode de choix de chemin ouvre de grandes perspectives de recherche et de nombreux travaux ont amené des améliorations sur le sujet. Ceci pourrait être automatisé et reconfigurable. Cependant, les nœuds de type point d'accès sont fixes et les stations des clients ne se déplacent que très peu. Nous considérerons donc la topologie comme étant fixe. De plus, nous ne gérerons pas les pannes d'un nœud. C'est pourquoi, dans cette étude, nous spécifierons les chemins de manière statique.

## 1.3 L'accès à la couche physique

### 1.3.1 Présentation de la méthode d'accès

La norme 802.11 définit deux modes d'accès à la couche liaison : DCF (Distributed Coordination Function) et PCF (Point Coordination Function). La méthode d'accès PCF propose une approche centralisée par un contrôleur nommé PC (Point Coordinator). Celui-ci définit une division temporelle du canal WiFi afin de garantir un temps de parole à chaque STA. Cependant, PCF n'est pas utilisé car les performances sont faibles lorsque le réseau contient beaucoup de STA. De plus, le PC doit être écouté de tous les nœuds, et il est probable qu'aucun nœud ne soit capable de jouer ce rôle dans un réseau maillé étendu.

La méthode d'accès DCF est basée sur CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Celle-ci prévoit que lorsqu'un élément du réseau veut émettre un paquet, il écoute le canal et attend qu'il soit libre. Contrairement à CSMA/CD (Carrier Sense Multiple Access with Collision Detection) que la norme Ethernet utilise, une interface sans fil ne peut pas écouter le canal et émettre en même temps. Le CSMA/CD permet de détecter et de signaler une collision entre deux paquets émis alors que CSMA/CA nécessite des accusés de réception pour approuver la bonne transmission du paquet.

Lorsque qu'un élément du réseau WiFi veut émettre un paquet, il attend que le canal soit libre. Pour ceci, il va écouter le canal pendant une durée nommée IFS (Inter Frame Space). L'unité de cette durée est le nombre de aSlotTime ( $20\mu s$ ). Pendant cette attente, s'il détecte une transmission, il va attendre la fin et à nouveau écouter pendant une durée de IFS. Comme présenté sur la figure 1.7, plusieurs durées sont possibles :

- SIFS (Short IFS) est la plus courte. Elle est utilisée avant les paquets de contrôle ACK (accusé de réception), avant CTS, ou avant un paquet suivant un dialogue RTS, CTS.
- PIFS (PCF IFS) est utilisée dans le cas de la méthode d'accès PCF.
- DIFS (DCF IFS) précède l'émission d'un paquet sans RTS/CTS, ou d'un RTS.
- EIFS (Extended IFS) remplace DIFS si le dernier paquet reçu est une collision ou s'il n'est pas reçu correctement.
- AIFS[AC] (Arbitration IFS) est défini dans la norme 802.11e. Il s'agit d'un tableau définissant une durée pour chaque classe de qualité de service définie dans la section suivante. La valeur correspondante au type du paquet va remplacer DIFS.
- Si AIFS est utilisé est si le dernier paquet n'est pas reçu correctement, la durée EIFS - DIFS + AIFS[AC] est utilisée.

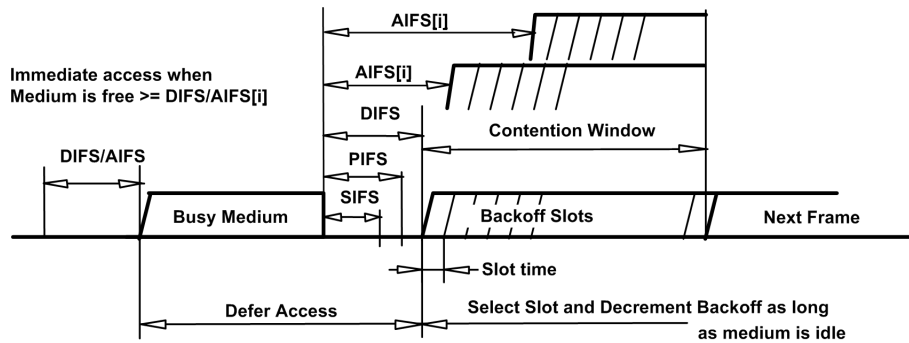


FIGURE 1.7 – Différents délais intertrames présentés dans la norme [IEE99a]

Si une transmission a été détectée pendant cette durée, l'émetteur compte à nouveau ce délai dès la fin de la transmission.

Si ce délai ne suit pas immédiatement une transmission, le paquet est émis sans attendre. Sinon, le nœud attend un nouveau délai nommé Backoff ou période de contention. En effet, si le médium est libre, il est très improbable que deux nœuds décident d'émettre exactement au même moment. Par contre, si le médium est occupé, il est fort probable que deux nœuds se synchronisent sur la fin de la transmission, attendent le même délai IFS, et émettent au même moment. Ceci produit alors une collision. C'est pourquoi ce délai aléatoire est introduit.

Le temps de Backoff est un nombre entier tiré aléatoirement entre 0 et  $CW$  (Contention Window) de constante  $aSlotTime$  ( $20\mu s$ ). Les constantes  $CW_{min}$  et  $CW_{max}$  sont définies pour chaque classe de qualité de service définie dans la section suivante. Lors de la première tentative d'émission, la valeur  $CW$  est égale à  $CW_{min}$ . En cas d'échec de transmission du paquet, une nouvelle tentative d'émission est programmée. A chaque nouvelle tentative,  $CW$  est élevé au carré, moins 1, jusqu'à  $CW_{max}$ . Si une transmission est reçue pendant ce délai, le processus est mis en pause, le temps de backoff attendu est mémorisé, et le reste du temps sera attendu après un temps IFS succédant la prochaine tentative d'émission. De cette manière, si le médium est souvent occupé, un paquet a de plus en plus de chance d'être émis s'il n'a pas encore été émis.

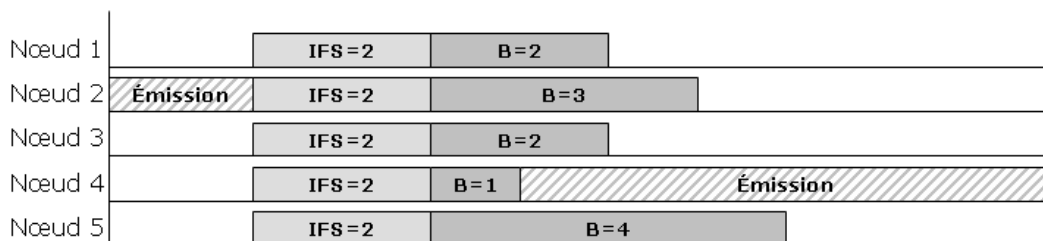


FIGURE 1.8 – Exemple de gain de période de contention

La figure 1.8 présente la compétition de cinq nœuds à la suite de l'émission du nœud 2. Après avoir attendu 2 unités de IFS, les cinq nœuds tirent une période de contention entre 0 et  $CW$  qui vaut 4. Le nœud 4 gagne avec le temps de backoff de 1 unité. Il



peut donc émettre. Les autres nœuds ayant détecté une transmission, ils attendent la fin avant de recommencer ce processus.

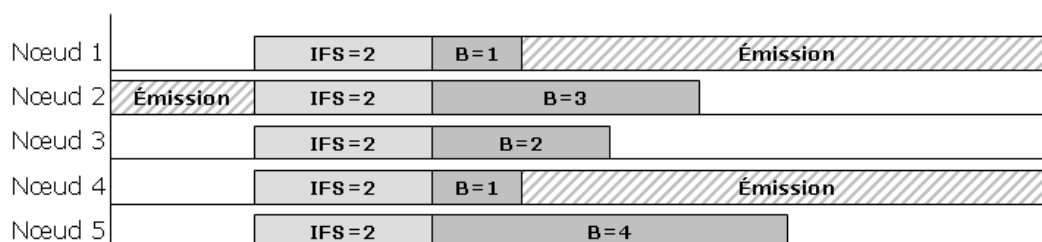


FIGURE 1.9 – Exemple de collision par tir du même backoff

Dans la figure 1.9, les nœuds 1 et 4 ont tiré le plus petit temps de backoff. Ils émettent ensemble, ce qui produit une collision. Les paquets devront donc être réémis.

A la suite de ces délais, le paquet est émis. Le récepteur répond avec un ACK (accusé de réception) qu'il émet après avoir attendu un délai de SIFS. Si le ACK n'est pas reçu, une nouvelle émission est tentée. Il peut y avoir jusqu'à 4 tentatives. Au cas où un accusé de réception serait perdu, tout nœud mémorise les identifiants des paquets reçus afin de ne pas utiliser de paquet en double.

### 1.3.2 Les paquets de contrôle RTS/CTS

Dans la norme 802.11, un point d'accès peut avoir plusieurs stations. Si une station veut communiquer avec une autre, le point d'accès est chargé de répéter le paquet. Les deux stations peuvent être lointaines et une station peut émettre sans savoir si l'autre est en train d'émettre. Ainsi, le point d'accès perçoit un signal incompréhensible résultant des deux signaux mélangés. Ce phénomène de collision est appelé problème du terminal caché.

Afin de l'éviter, les paquets de contrôle RTS/CTS permettent de réserver par annonce le médium. La station émet un RTS (Request To Send) contenant la durée de réservation nécessaire, et le point d'accès répond un CTS (Clear To Send) afin de confirmer qu'il a bien reçu le RTS et de répéter la durée de réservation du médium aux autres nœuds qui n'ont pas forcément reçu le CTS. L'autre station va donc se taire pendant l'émission de la première station, et le point d'accès pourra répondre par son accusé de réception. Pour optimiser les performances, l'utilisation de RTS/CTS est souvent conditionnée à un seuil minimum de taille de paquet.

Dans la figure 1.10, le CTS émis par AP va être reçu par STA 2, et va l'interdire d'émettre pendant la durée de l'émission du paquet de STA 1 ainsi que de l'ACK.



FIGURE 1.10 – Paquet de contrôle RTS/CTS

### 1.3.3 La qualité de service (QoS)

Depuis 2005, l'annexe 802.11e [IEEE05] ajoute à la norme WiFi une gestion de qualité de service sur la couche MAC. Ceci est nécessaire pour les applications à temps critiques comme la téléphonie, ou le transport de vidéo (streaming). Cette évolution est aussi appelée WMM (Wireless MultiMedia) ou WME (Wireless Multimedia Extension).

Tout d'abord, une classification de paquet est mise en place. Dans l'entête 802.11, un champ supplémentaire est spécifié afin de caractériser la qualité de service à appliquer au paquet. Il peut être un des quatre types suivants :

- VO : Vo-Ip (voix sur IP)
- VI : Video (vidéo)
- BE : Best Effort (au mieux)
- BK : BackGround (arrière plan)

Par défaut, ce paramètre est défini en fonction du paramètre TOS (Type Of Service) de l'entête IP (défini par la norme 802.1d). Par exemple, la consultation de page web sera en BK alors que la téléphonie sera en VO. Cette classification permet tout d'abord de prioriser l'émission de paquet suivant l'ordre de priorité : VO, VI, BE, BK.

Une nouvelle méthode d'accès nommé EDCA (Enhanced Distributed Channel Access) permet que les paquets d'une classe plus élevée aient plus de chance d'être prioritaires par rapport à d'autres paquets si deux nœuds venaient à vouloir émettre en même temps. En effet, pour chaque classe de qualité de service, un nouveau temps intertrame AIFS est défini et remplace le seul DIFS. De même, les constantes permettant de calculer les périodes de contention  $CW_{min}$  et  $CW_{max}$  sont définies en fonction de la classe de QoS. Ces délais seront plus faibles pour les paquets prioritaires.

## 1.4 Evaluation de performance, notions de métrologie

### 1.4.1 Introduction

Afin de qualifier une communication, que ce soit en simulation ou en expérimentation, il est nécessaire de définir les éléments que nous allons mesurer.

### 1.4.2 Le débit

Souvent mesuré en  $Kbit.s^{-1}$ , le débit indique la vitesse de transmission de données transitant entre deux points. Le débit physique est directement lié à la modulation sur la couche radio. La norme 802.11b définit des débits allant de 1 à 11  $Mbit.s^{-1}$ . 802.11a et 802.11g ont améliorés les débits de 6 à 54  $Mbit.s^{-1}$ . Et récemment, 802.11n définit des débits allant jusqu'à 300  $Mbit.s^{-1}$ . Cependant, l'encapsulation de données, l'ajout de temps intertrame, et les collisions définissent un débit utile sur la couche liaison plus faible. Enfin, sur un réseau WiFi maillé, au niveau de la couche transport (par exemple UDP), le débit mesuré prend en compte la répétition de paquet et l'acheminement de bout en bout.

### 1.4.3 La latence

La latence est le délai entre le dépôt dans la queue d'émission d'un paquet et la réception du paquet dans un autre nœud. Ceci peut aussi être appelé le retard. Le RTT (Round Trip Time) est le temps d'aller-retour d'un paquet. Il peut être mesuré avec l'utilitaire "ping" qui permet d'émettre un paquet à un destinataire qui va répondre. Le nœud source va pouvoir mesurer le temps entre le moment où il a émis le paquet, et le moment où il a reçu la réponse.

### 1.4.4 La gigue

La gigue  $J$  est la moyenne des variations de délai entre deux paquets successifs. Cette valeur est importante pour qualifier une communication audio. Elle est notamment définie dans la RFC 1889 qui normalise le protocole RTP (Real Time Protocol). Pour un paquet  $i$ , le délai  $D_i$  est calculé à partir de  $R_i$  qui est la date d'arrivée du paquet  $i$ , et  $S_i$  qui est la date d'envoi du paquet  $i$ . La date d'envoi  $S_i$  peut être calculée à partir de  $T_i$  qui est le numéro d'échantillon contenu dans le paquet RTP. Par exemple, en utilisant le codec audio G.711, la fréquence d'échantillonnage  $F$  est de 8000Hz.

$$S_i = \frac{T_i}{F} \quad (1.1)$$

$$D_i = (R_i - R_{i-1}) - (S_i - S_{i-1}) \quad (1.2)$$

$$J_i = J_{i-1} + \frac{|D_i| - J_{i-1}}{16} \quad (1.3)$$

La moyenne est calculée par un estimateur du premier ordre, et le gain  $\frac{1}{16}$  permet une bonne réduction du bruit.

### 1.4.5 La qualité de service

Pour qu'une conversation audio puisse être comprise par les interlocuteurs, la qualité de celle-ci est importante. Cependant, cette qualité est dépendante de plusieurs caractéristiques. Tout d'abord, l'encodage du signal et sa compression vont altérer plus ou moins cette qualité, et définir la taille et la fréquence d'émission de paquet. Le choix et la qualité engendrés par le codage d'une conversation audio dépassent le cadre de notre étude, c'est pourquoi nous nous limiterons à comparer les débits nécessaires à quelques exemples de codec. Si le débit disponible sur le médium est inférieur au débit nécessaire pour la transmission du flux audio, la qualité va être fortement altérée.

Une autre mesure importante permettant de qualifier une transmission audio est la latence. En effet, si le délai est perceptible, les interlocuteurs devront attendre entre chaque échange. Enfin, la gigue, qui correspond aux variations de la latence, va fortement influencer la qualité de la communication. Si certains paquets arrivent en retard, c'est-à-dire après la date où ils auraient été nécessaires, le tampon d'entrée devra être augmenté, ce qui augmentera le délai. Toutes ces contraintes sur le médium vont altérer la communication audio. La qualité devra être suffisante pour que la conversation soit audible.

De même pour la transmission de vidéo, ces paramètres vont en influencer la qualité. Ces transmissions peuvent être unidirectionnelles pour de la vidéosurveillance, ou bidirectionnelles pour de la vidéoconférence.

#### 1.4.6 Le temps d'utilisation du canal WiFi

L'utilisation d'un canal WiFi est limitée dans une zone donnée. En effet, il ne peut y avoir qu'une transmission à la fois à un instant donné. La somme de toutes les durées de trames pendant une période définie va donc indiquer combien de temps le canal a été occupé sur la durée de la période. De plus, les délais intertrames peuvent être déduits en fonction du type de trame. Cette mesure permet d'évaluer l'occupation du médium.

#### 1.4.7 Mesure des ces indicateurs

Afin d'évaluer la qualité de service présente sur un réseau WiFi maillé, ces indicateurs peuvent être mesurés par une expérimentation. En effet, les valeurs mesurées sont des valeurs réelles et reflèteront la qualité de service disponible.

### 1.5 Analyse par expérimentation

#### 1.5.1 Introduction

Pour évaluer la qualité de service, nous avons mis en place 3 points d'accès WiFi relais présentés sur la figure 1.11. Cette première approche expérimentale, sur le terrain, va permettre d'avoir un premier aperçu de la qualité d'un réseau maillé. En effet, les valeurs mesurées sont des valeurs réelles. De plus, cette expérimentation permet de mesurer l'impact d'un premier levier qui est l'activation de la norme de qualité de service 802.11e par le paramètre WMM.

Une manière d'évaluer les performances est de créer des interfaces logiques de type "espion" sur chaque point d'accès, de manière à enregistrer tous les paquets étant émis et reçus par la carte WiFi. Cette méthode permet d'obtenir tous les paquets transmis et reçus par le nœud "espion" avec la date. Cependant, tous les points d'accès ne peuvent pas être synchronisés temporellement avec une précision permettant de déduire les différents délais de transmission (la dizaine de microseconde). De plus, les paquets qui ne sont pas transmis correctement ne sont pas visibles par l'interface "espion". Nous ne pouvons donc pas avoir toutes les données pour reconstituer le trafic. Enfin, cette méthode d'analyse ne donnerait qu'une image statique une fois saisie. C'est-à-dire que nous ne pourrions pas jouer différents scénarii pour chercher la bonne stratégie de gestion de la qualité de service, ou tout autre levier favorisant son amélioration.

#### 1.5.2 Présentation de l'expérience

Afin d'évaluer la qualité de service d'un réseau maillé 802.11, nous avons mis en place 3 points d'accès WiFi relais (figure 1.11). Ils sont équipés d'une seule interface WiFi physique composée d'un chipset Atheros. Un seul canal WiFi va donc être utilisé.

Le pilote permet de créer plusieurs interfaces logiques de type point d'accès, WDS, client ou ad-hoc.

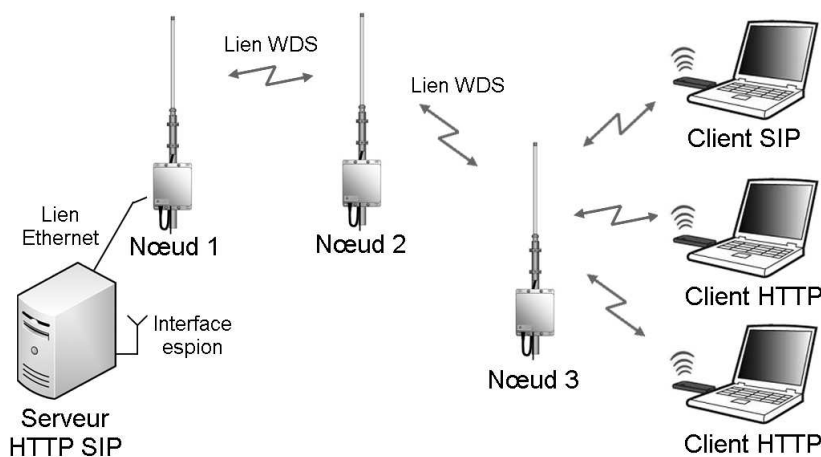


FIGURE 1.11 – Topologie du réseau expérimental

Le premier est connecté en Ethernet à un ordinateur jouant le rôle de serveur HTTP et SIP. Deux liens WDS sont mis en place : le premier entre les nœud 1 et 2, et le second entre les nœud 2 et 3. De plus, le nœud 3 joue le rôle de point d'accès sur la même interface radio que celle utilisée pour communiquer avec le nœud 2.

Dans chacun des nœuds, les interfaces sont configurées en point réseau de manière à fonctionner sur la couche liaison et être transparent au niveau de la couche IP. Le client SIP se connecte au serveur de manière à mettre en place un flux RTP sur UDP transportant les données audio de manière bidirectionnelle. Les clients HTTP téléchargent un gros fichier en HTTP.

Pour chaque expérimentation, 2 captures de trame sont mises en place :

- la première est située sur l'interface Ethernet du serveur. Elle permet d'enregistrer tout le trafic réseau du serveur.
- la seconde enregistre tout le trafic radio à partir de la carte WiFi du serveur configuré en mode "espion". Les sources et destinations des paquets pourront être identifiées à partir des adresses lues dans les paquets.

	WMM désactivé	WMM activé
RTP seul	1	4
RTP et HTTP	2	5
RTP et 2 HTTP	3	6

TABLE 1.1 – Listes des expérimentations

Lors des tests 1, 2 et 3, l'option WMM est désactivée. Cette option active la compatibilité avec la norme de qualité de service 802.11e. Cette option est activée lors des tests 4, 5 et 6.

Lors de toutes les expérimentations, le client SIP établit une connexion RTP. Pour les tests 1 et 4, seule cette connexion RTP est établie. Pour les tests 2 et 5, un client HTTP télécharge un fichier sur le serveur. Pour les tests 3 et 6, deux clients HTTP téléchargent un fichier sur le serveur.

Pour chacun de ces tests, une mesure de ping est mise en place afin de mesurer le temps aller/retour à partir du serveur jusqu'au client SIP.

De même, la connexion RTP transitant entre le nœud 3 et le client SIP est capturée par l'interface "espion" du serveur. A partir de cette capture, les graphes de gigue sont présentés pour chaque test.

### 1.5.3 Résultats

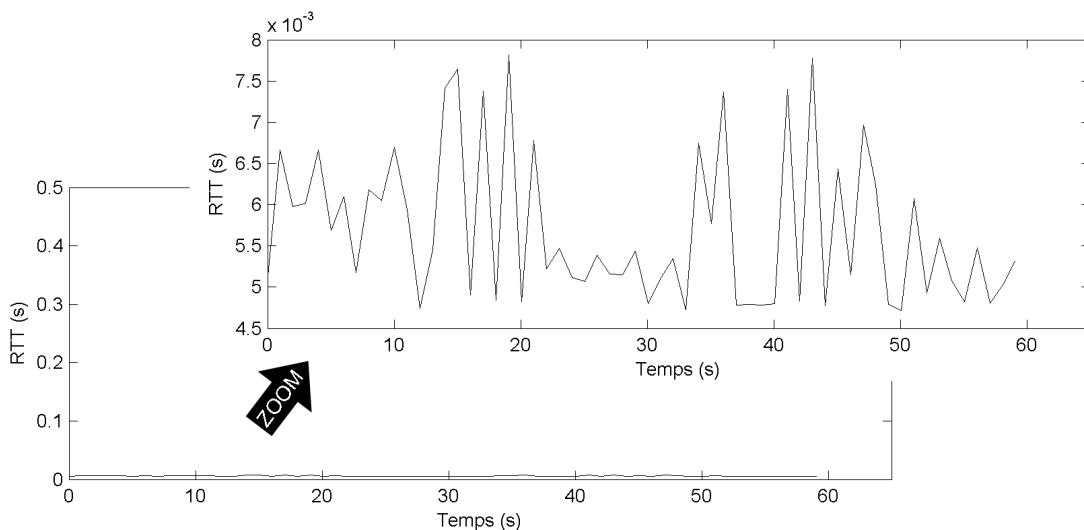


FIGURE 1.12 – RTT sans WMM avec 1 RTP

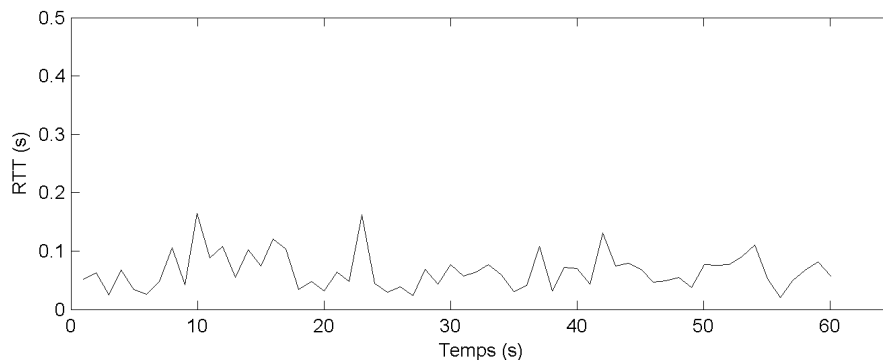


FIGURE 1.13 – RTT sans WMM avec 1 RTP et 1 HTTP

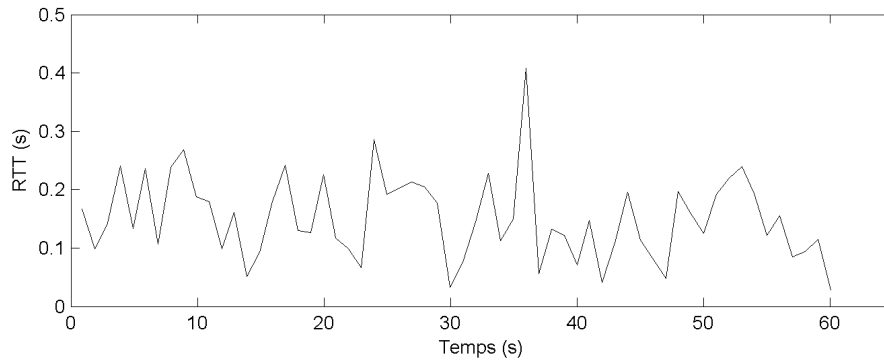


FIGURE 1.14 – RTT sans WMM avec 1 RTP et 2 HTTP

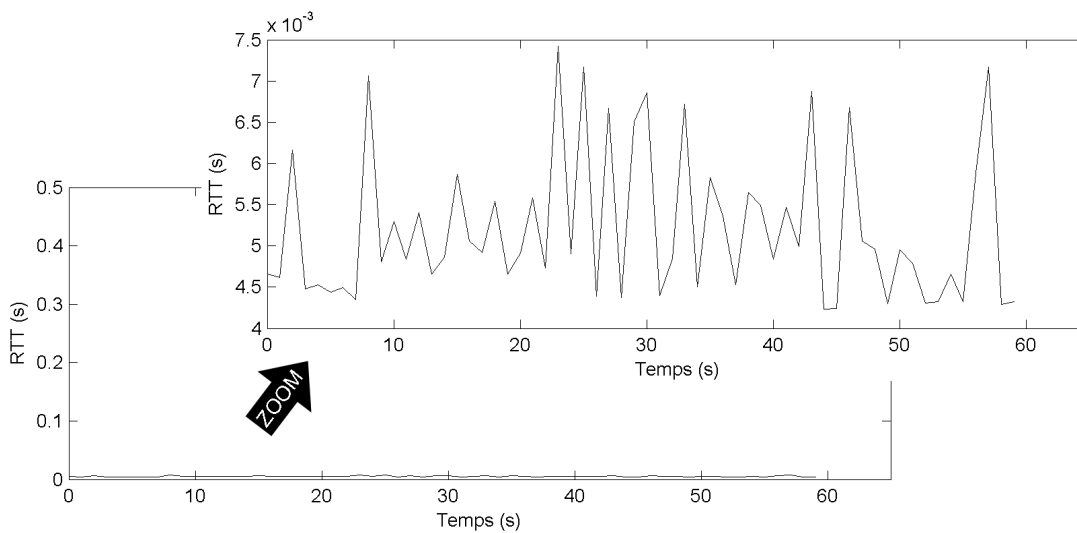


FIGURE 1.15 – RTT avec WMM avec 1 RTP

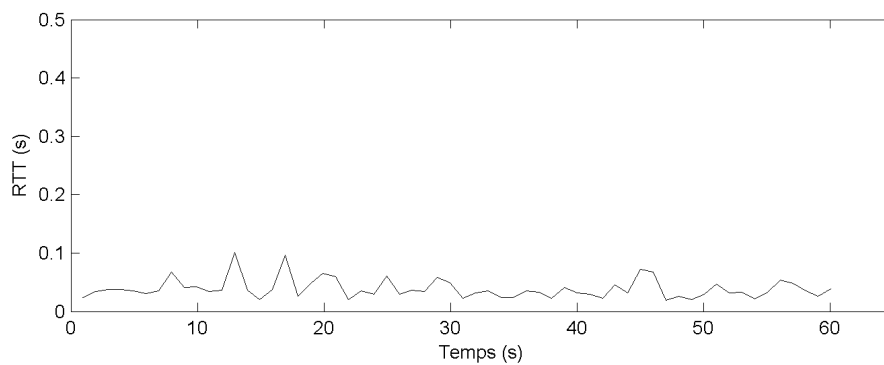


FIGURE 1.16 – RTT avec WMM avec 1 RTP et 1 HTTP

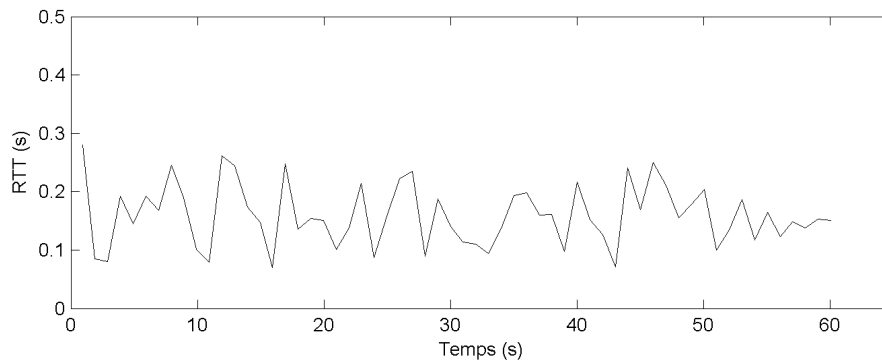


FIGURE 1.17 – RTT avec WMM avec 1 RTP et 2 HTTP

	Min	Max	Moy	Var
1	4,71 ms	7,81 ms	5,69 ms	0,001 ms
2	20,19 ms	164,23 ms	65,79 ms	0,92 ms
3	28,74 ms	407,98 ms	151,40 ms	4,83 ms
4	4,23 ms	7,43 ms	5,20 ms	0,001 ms
5	6,66 ms	100,87 ms	38,80 ms	0,27 ms
6	69,43 ms	280,58 ms	159,45 ms	2,45 ms

TABLE 1.2 – Liste des RTT Mesurés

Les figures 1.12, 1.13 et 1.14 montrent que le délai de traversée du réseau augmente en fonction de l'utilisation du canal lorsque WMM n'est pas activé. En effet, en ajoutant une communication HTTP (figure 1.13, puis deux communications HTTP (figure 1.14), les mesures de RTT augmentent. De même lorsque le RTT est activé, les figures 1.15, 1.16 et 1.17 montrent le même phénomène.

L'impact de WMM est faible lorsque le canal est peu utilisé. En effet, les figures 1.12 et 1.15 se ressemblent. Ceci est expliqué car lorsque le canal est peu utilisé, les paquets sont relayés immédiatement avec WMM activé ou non.

Cependant, avec une communication HTTP, les figures 1.13 et 1.16 montrent une amélioration du RTT lorsque WMM est activé. En présence de deux communications HTTP, les figures 1.14 et 1.17 montrent que le RTT est très important avec ou sans WMM.



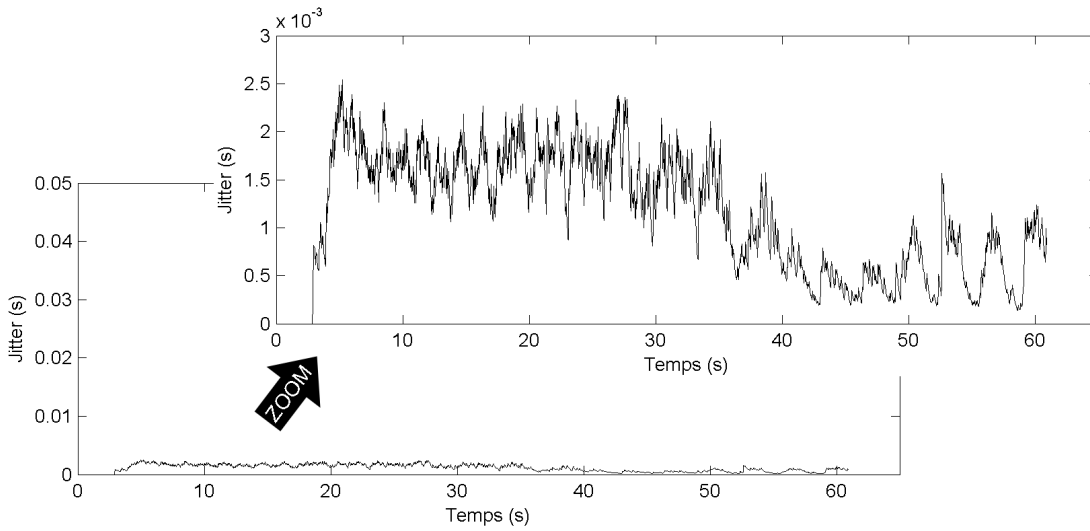


FIGURE 1.18 – Gigue sans WMM avec 1 RTP

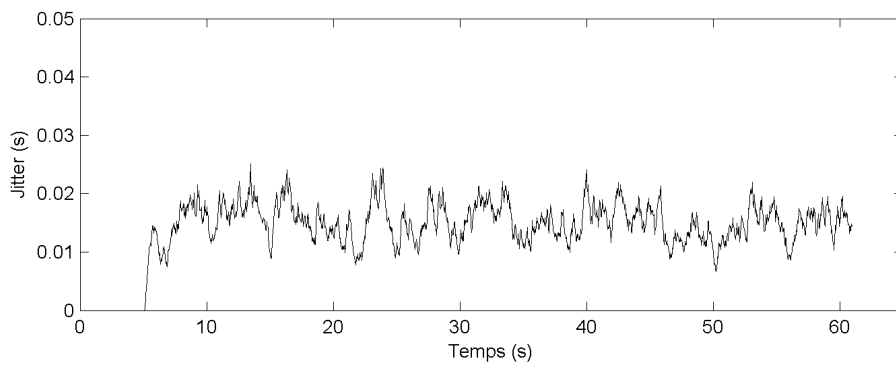


FIGURE 1.19 – Gigue sans WMM avec 1 RTP et 1 HTTP

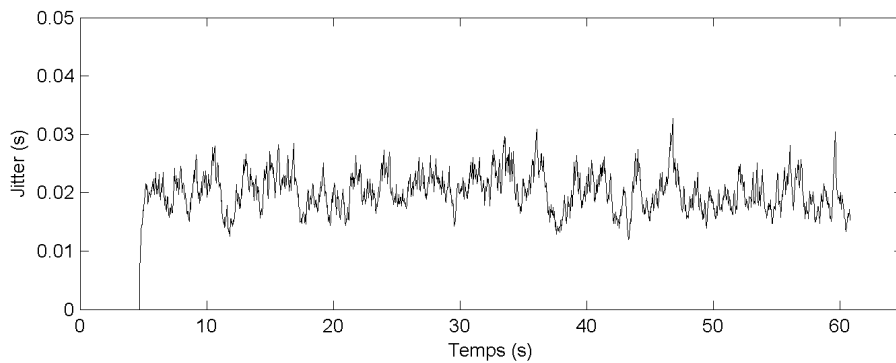


FIGURE 1.20 – Gigue sans WMM avec 1 RTP et 2 HTTP

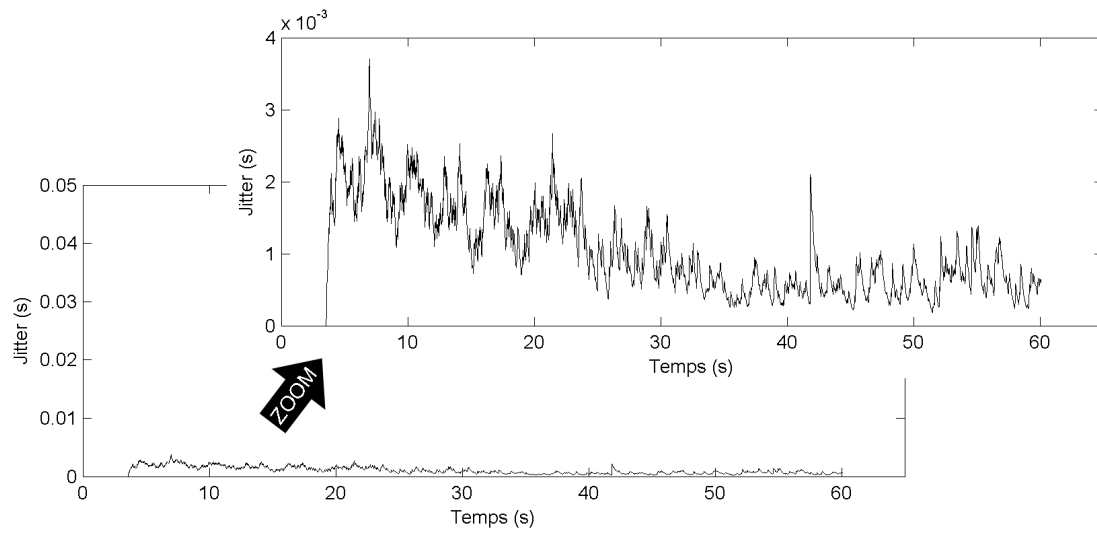


FIGURE 1.21 – Gigue avec WMM avec 1 RTP

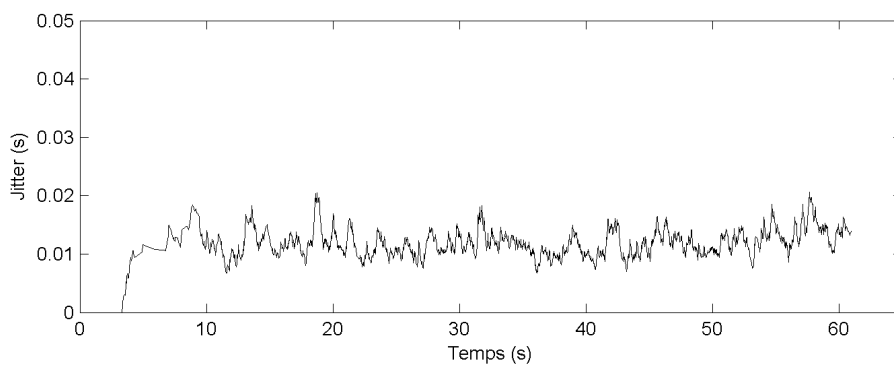


FIGURE 1.22 – Gigue avec WMM avec 1 RTP et 1 HTTP

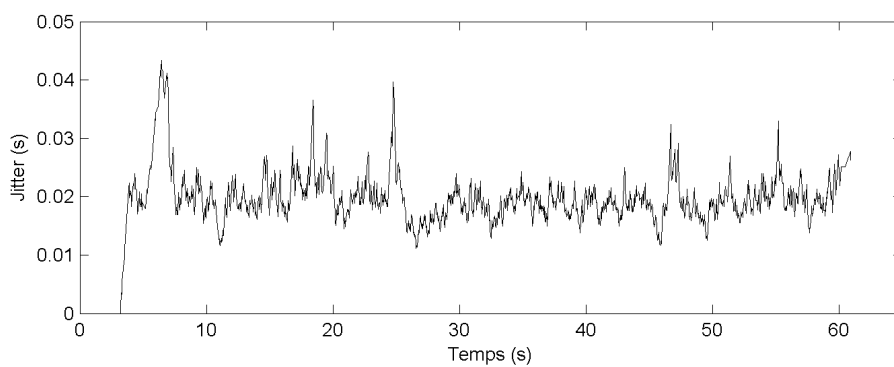


FIGURE 1.23 – Gigue avec WMM avec 1 RTP et 2 HTTP

	Max	Moy	Var	Paquets	Réémissions	Pertes
1	2,54 ms	1,18 ms	0,00 ms	2969	65	0
2	25,08 ms	15,24 ms	0,01 ms	2798	74	7
3	32,71 ms	20,13 ms	0,01 ms	2771	48	21
4	3,70 ms	1,05 ms	0,00 ms	2863	72	3
5	20,66 ms	11,93 ms	0,01 ms	2567	43	267
6	43,35 ms	19,66 ms	0,01 ms	2760	79	144

TABLE 1.3 – Liste des Gignes Mesurés

De même que le RTT, les figures 1.18, 1.19 et 1.20 montrent que la gigue de la communication audio augmente lorsque le canal est utilisé avec WMM désactivé. Lorsque WMM est activé sur les figures 1.21, 1.22 et 1.23, la gigue augmente aussi quand le canal devient chargé. Sur les figures 1.18 et 1.21, la gigue est faible car le réseau est peu utilisé. Avec une communication HTTP, WMM améliore sensiblement la gigue sur les figures 1.19 et 1.22. Lorsque le canal est très chargé, les figures 1.20 et 1.23 montrent que la gigue est importante avec ou sans WMM.

Le nombre de paquets correctement transmis peut être mesuré. Dans ces conditions expérimentales, la taille d'un paquet correctement transmis ainsi que le débit physique auquel il a été transmis est disponible. Les temps intertrames peuvent être déduits en fonction de la nature du paquet. Cependant, les périodes de contention, les collisions et les erreurs de transmissions ne peuvent pas être détectées et donc comptabilisées dans la partie "canal utilisé".

	Paquets	Canal libre	Canal utilisé
1	39289	57,958 s (93,57 %)	2,982 s ( 4,82 %)
2	186843	24,708 s (39,84 %)	36,318 s (58,55 %)
3	202265	26,784 s (43,29 %)	34,081 s (55,09 %)
4	33414	58,623 s (94,60 %)	2,347 s ( 3,79 %)
5	246086	26,815 s (43,24 %)	34,192 s (55,14 %)
6	159018	27,055 s (43,63 %)	33,955 s (54,76 %)

TABLE 1.4 – Etat du canal

#### 1.5.4 Problèmes observés

Cette expérimentation montre que, dans ce petit réseau maillé, des transmissions vocales qui nécessitent de la qualité de service sont dégradées par des grosses transmissions de données concurrentes. En effet, l'écoute des communications audio était dégradée lorsque la mesure de la gigue était élevée pendant le test.

De plus, l'activation de la norme de qualité de service 802.11e n'améliore que faiblement la qualité audio lorsque le réseau est chargé.

### 1.5.5 Conclusion

Cette expérimentation permet de voir que le RTT reste correct sur un réseau WiFi maillé lorsqu'il y a peu de communication. De plus, l'activation de WMM améliore le RTT et la gigue d'une communication audio en présence d'un trafic moyen. Avec un trafic très important, nous pouvons constater de nombreuses pertes et réémissions, avec WMM activé ou non.

L'enregistrement de tout ce qu'il se passe sur la fréquence radio utilisée par le WiFi permettrait de comprendre précisément les problèmes de communication. Il serait nécessaire d'utiliser plusieurs analyseurs vectoriels synchronisés temporellement, et positionnés judicieusement pour avoir toutes les informations, et tous les événements d'une transmission. Cette méthode nécessitant plusieurs matériels très performants, elle n'a pas pu être mise en œuvre.

De plus, cette expérimentation utilise des nœuds positionnés à proximité, ce qui signifie que chacun reçoit le signal de tous les autres. Afin d'expérimenter des communications avec certains nœuds qui ne se captent pas, de grandes surfaces sont nécessaires. Cependant, l'expérimentation reste une bonne méthode pour mesurer réellement l'impact des améliorations sur les leviers identifiés.

Dans cette thèse, nous cherchons à améliorer la qualité de service. L'expérimentation précédente montre que celle-ci n'est pas optimale pour du transport de communications audio ou vidéo. Cependant, l'utilisation de réseau maillé étant en plein essor, des travaux ont déjà été effectués dans ce domaine. C'est pourquoi nous pouvons poursuivre cette étude par un état de l'art sur l'amélioration de la qualité de service du WiFi maillé.



## Chapitre 2

# Etat de l'art

Le WiFi maillé est une nouvelle méthode qui permet le développement d'un réseau étendu. Ce domaine intéresse de nombreux chercheurs car couvrir de grandes surfaces tout en conservant une qualité de service acceptable pose de nombreux problèmes. C'est pourquoi nous commencerons par dresser un état de l'art des recherches en cours. Celles-ci vont présenter les problèmes rencontrés, ainsi que des solutions qui ont été proposées.

### 2.1 Solutions orientées couche physique

#### 2.1.1 Le WiFi maillé sur plusieurs canaux

Afin d'augmenter le nombre de transmissions radio, une idée est d'utiliser plusieurs canaux. C'est ce qui est proposé dans [KJCV06]. Si le nœud n'a qu'une interface radio, les auteurs envisagent que les interfaces radios changent de canaux. Pour ceci, ils présentent plusieurs possibilités. Tout d'abord, ils proposent de fixer des rendez-vous sur un canal donné pour un couple de nœud. Ensuite, une autre possibilité est de changer de canal en suivant une séquence connue de tous. Si le nœud a plusieurs interfaces radios, une interface peut être sur un canal fixe, et les autres peuvent changer de canal afin de communiquer avec ses voisins. Le broadcast doit être géré de manière particulière vu que la réémission ne pourra pas être immédiate et se devra se faire sur des canaux différents. Les auteurs ont présenté plusieurs protocoles de gestions de transmission multi canal : MMAC (Multichannel MAC) et HCMP (Hybrid MultiChannel MAC Protocol) avec le simulateur réseau NS-2 [NS-11a]. MMAC est conçu pour des nœuds avec une seule interface radio alors que HCMP est optimisé pour des réseaux composés de nœuds à deux interfaces. Cependant, changer de canal de manière très fréquente n'est pas implémenté dans les pilotes des cartes WiFi existante, ni dans les système d'exploitation.

Un autre moyen d'utiliser plusieurs canaux est présenté dans [JR05]. Les auteurs utilisent des points d'accès ayant deux interfaces radio. Ils mettent en place des groupes de points d'accès avec la première interface radio sur un même canal. Afin de permettre des communications d'un groupe à l'autre, certains points d'accès utilisent une seconde interface radio afin de communiquer avec le groupe voisin. Cette approche nommée Clustered Multi-channel Two-radio (CMT) permet de rester compatible avec du matériel WiFi standard. Sur les nœuds n'utilisant qu'une interface radio, la seconde permet de

contrôler l'utilisation des canaux et de tester si un changement de canal ne permettrait pas d'optimiser les transmissions. Le point d'accès qui a le plus de liens avec les autres membres du cluster est élu contrôleur de cluster (Cluster Head). Les auteurs ont mesuré les performances avec le simulateur réseau OPNET [OPN11]. Les résultats apportés par cette méthode montrent une augmentation du débit atteignable sur ce réseau. En effet, cette méthode permet notamment de choisir les canaux radios les moins bruités par l'environnement.

Une autre utilisation de plusieurs interfaces sans fil par nœud est présentée dans [NNCA06]. Une interface est utilisée comme point d'accès permettant à des stations de se connecter, et l'autre interface permet le maillage avec les nœuds voisins 2.1.

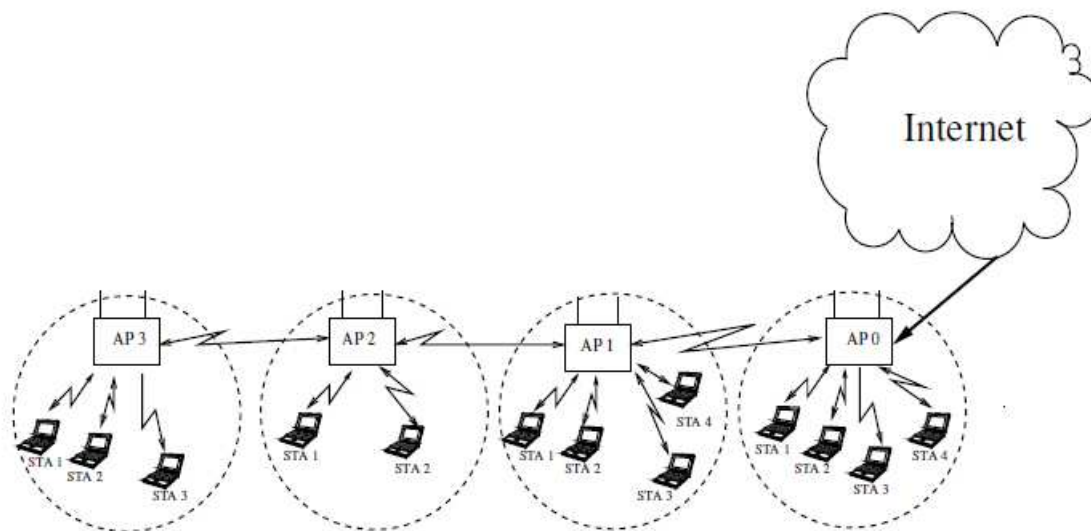


FIGURE 2.1 – Topologie présentant quatre nœuds alignés

Les simulations montrent des débits corrects pour les stations du point d'accès connecté à Internet, mais des débits progressivement plus faibles pour les nœuds suivants (AP1 :  $566 \text{ Kbit.s}^{-1}$ , AP2 :  $61 \text{ Kbit.s}^{-1}$ , AP3 :  $9 \text{ Kbit.s}^{-1}$ ). Les auteurs [BCG05] indiquent que ce phénomène est dû au problème du terminal caché. En effet, la simulation montre un remplissage des files d'attente d'émission du nœud AP2. Lorsqu'une file est pleine, les nouveaux paquets à empiler sont perdus. De nombreux paquets de AP3 sont donc perdus. Le nœud AP2 a moins de chance de gagner la période de contention d'une part parce que ce nœud reçoit à la fois AP1 et AP3, et d'autre part car lorsqu'il détecte un paquet mal reçu, il attend un délai intertrame plus important nommé EIFS, et il détecte beaucoup de collisions entre AP1 et AP3. Les auteurs proposent donc une nouvelle gestion de la file d'attente nommée QMMN (Queue Management in Multi-hop Networks). L'idée principale de cette proposition est premièrement de limiter les flux provenant des nœuds proches car ils vont rapidement remplir les files d'attente, et deuxièmement de prioriser les flux qui arrivent plus lentement parce qu'ils ont traversé plusieurs nœuds. Chaque nœud va donc mémoriser, pour chaque source, le nombre de paquets en queue, et le délai moyen entre chaque arrivée de paquet. Des simulations avec le logiciel NS-2 [NS-11a], montrent que cette gestion de la file d'attente améliore

le débit des transmissions issues des nœuds ayant effectués plus de sauts.

### 2.1.2 Changement de modulation basé sur des chaînes de Markov

Une approche de modélisation analytique puis de simulation a été réalisée dans [NHF07]. En effet, les auteurs proposent une modélisation d'un réseau de capteurs sans fil maillé. Les nœuds n'ont qu'une interface radio et partagent donc le même canal. La méthode d'accès est du TDMA (Time Division Multiple Access). Le canal radio est modélisé par le fading de Nakagami-m, qui est représenté par un FSMC (Finite State Markov Chain). Chaque état de la chaîne de Markov correspond à une modulation utilisée sur le canal radio qui est de type  $2^n - QAM$ . Le rapport signal sur bruit (SNR, Signal to Noise Ratio) reçu est partitionné en  $N+1$  intervalles, qui commande l'état de la chaîne de Markov. En supposant que le canal varie lentement, ce qui est défini dans Nakagami-m, les variations de modulation ne se feront que pour des états adjacents. Ainsi, une probabilité de changement de modulation, et donc de débit peut être spécifiée. A partir de ce modèle, les auteurs présentent une couche radio avec une probabilité d'erreur et de changement de modulation. Cependant, pour rester compatibles avec des stations respectant la norme 802.11, nous n'orienterons pas nos travaux sur la couche radio. Nous nous limiterons à appliquer un débit dépendant de la qualité des liens et d'appliquer un taux d'erreur manuellement.

## 2.2 Solutions protocolaires pour le WiFi maillé

### 2.2.1 Approches visant l'équilibrage de charge par les point d'accès

Afin d'améliorer la qualité de service, l'auteur de [JAB08] a travaillé sur l'équilibrage de charge entre points d'accès. Dans le réseau présenté, où tous les points d'accès sont reliés en filaire, les stations spécifient la qualité de service requise, et un serveur indique quel point d'accès est le plus approprié à cette demande. Ainsi, le serveur intègre cette nouvelle station au réseau, et peut redistribuer les charges entre les points d'accès. Cependant, ceux-ci ne communiquent pas entre eux sans fil. En effet, chaque groupe composé d'un point d'accès avec ses stations associées, peut utiliser un canal différent. De plus, cette approche implique que les stations implémentent un protocole spécifique pour interroger le serveur. Ceci n'est pas compatible avec le service que nous voulons proposer où les stations appartiennent à des clients, et où nous ne voulons pas installer de logiciel spécifique. Cependant, cette approche est envisageable si la topologie du terrain permet aux stations d'avoir le choix entre plusieurs points d'accès, et que ce soit les points d'accès qui acceptent où refusent les stations en fonction de la charge réseau.

### 2.2.2 Approche traitant du problème du nœud caché

Un nœud caché est un élément WiFi qui peut communiquer avec son voisin, mais qui ne peut pas savoir que le voisin du voisin est en train d'émettre. Si les deux nœuds qui ne se voient pas, décident d'émettre en même temps, le nœud central reçoit une collision. C'est ce qui est présenté dans [LKC05]. Les auteurs analysent le débit d'une connexion TCP sur un réseau en chaîne à base de lien ad hoc.



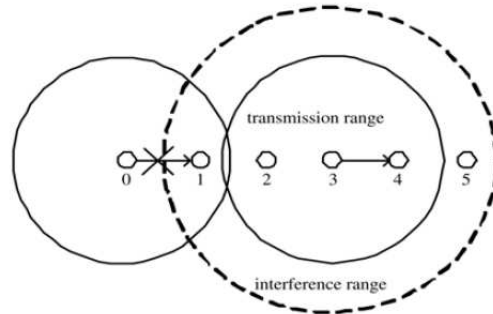


FIGURE 2.2 – Topologie et zone d'influence

Ils présentent un exemple de topologie en chaîne, figure 2.2, avec les nœuds 0, 1, 2, 3, 4, 5, 6 où un nœud central 3 est reçu de ses voisins 2 et 4, et perturbe les nœuds 1 et 5. Les nœuds 0 et 6 ne sont pas affectés par une émission du nœud 3. Leurs modèles analytiques ainsi que leurs simulations avec le simulateur réseau NS-2 mettent en évidence que le problème du terminal caché joue fortement sur les réémissions TCP et donc le débit de bout en bout.

Ce même problème est présenté dans [XG06]. Pour ceci, les auteurs proposent une méthode permettant d'améliorer la qualité de service de flux vidéos au travers d'un réseau maillé WiFi. Ils présentent tout d'abord un moniteur de liens au niveau de la couche MAC permettant de mesurer la capacité du lien ainsi que les débits de chaque flux vidéo transitant par ce nœud. La bande passante disponible peut alors être calculée. Ce moniteur estime aussi la congestion sur un nœud en calculant le délai moyen entre la réception d'un paquet et sa réémission. A partir de ces données, les débits des flux des caméras sont diminués ou augmentés afin d'optimiser la congestion sur les nœuds.

La publication [TCCYMC06] présente une amélioration de la qualité de service sur un réseau maillé WiFi, à base de jeton. Leur proposition est basée sur une topologie en chaîne uniquement, avec un nombre de nœuds connu comme présenté sur la figure 2.3.

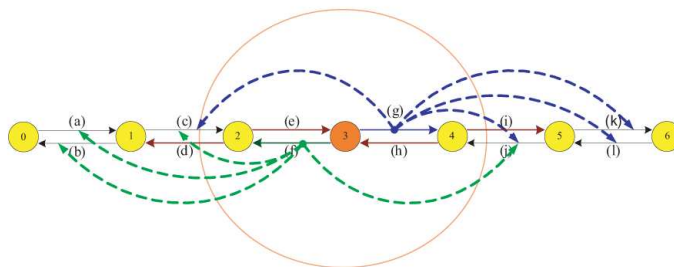


FIGURE 2.3 – Topologie et influence des nœuds de cette étude

La problématique suivante est présentée figure 2.4 : pour que la communication entre les nœuds 3 et 4 réussisse, les nœuds 2 et 5 ne doivent pas émettre. De plus, si un paquet est émis par le nœud 1, le nœud 2 sera occupé à répondre par un accusé de réception et de même pour les nœuds 5 et 6. Enfin, si le nœud 3 répond au nœud 2, il peut y avoir collision avec le nœud 1 qui répond au nœud 0. En résumé, lorsque le nœud 3 émet, les

nœuds 0 à 6, qui ont jusqu'à trois sauts d'écart, ne doivent pas émettre pour éviter des collisions.

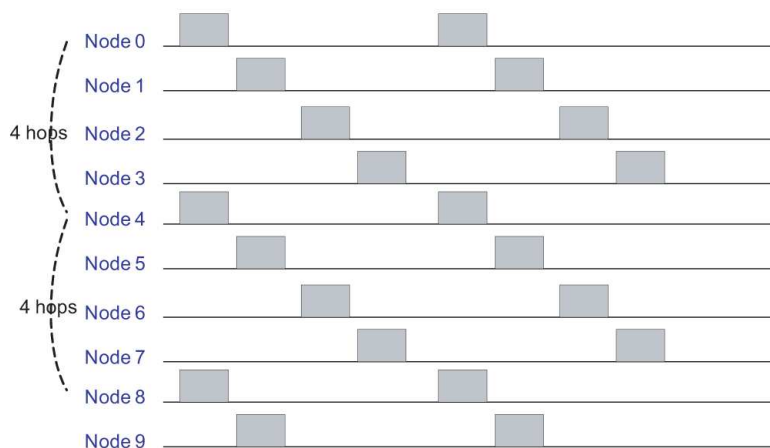


FIGURE 2.4 – Planification des émissions

A partir de ces limitations, les auteurs proposent une méthode d'accès basée sur le TDMA (Time Division Multiple Access) nommé TDS (Token Based Scheduling). Les groupes de nœuds pouvant émettre au même moment sans risque de collision pourront successivement émettre. Afin de valider leurs travaux, les auteurs utilisent le simulateur Glomosin. Les résultats montrent une diminution des délais moyens d'arrivée des paquets, ainsi qu'une amélioration des débits lorsque le réseau comporte plus de 7 nœuds.

L'activation des paquets de contrôle RTS/CTS peut aider à résoudre ce problème du nœud caché. Cependant, dans [JYM07], les auteurs identifient un problème avec l'utilisation ces paquets de contrôle, qu'ils nomment PPC (Persistent Pseudo-Clearance). Ce problème apparaît essentiellement en présence de deux paires de nœuds suivant la topologie ci-dessous.

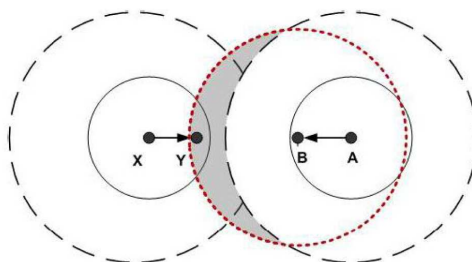


FIGURE 2.5 – Topologie des nœuds X Y B A

Dans cette topologie figure 2.5, les nœuds ne voient que leurs voisins. X reçoit Y. Y reçoit X et B. B reçoit Y et A, et A reçoit B. Le scénario posant problème est le suivant. A émet un RTS à B. Celui-ci répond CTS. Y ne reçoit pas ce CTS suite à une interférence. A émet alors son paquet. Si à ce moment là X émet un RTS à Y, les paquets sont perdus car les CTS qu'ils émettent perturbent la transmission du paquet

(figure 2.6).

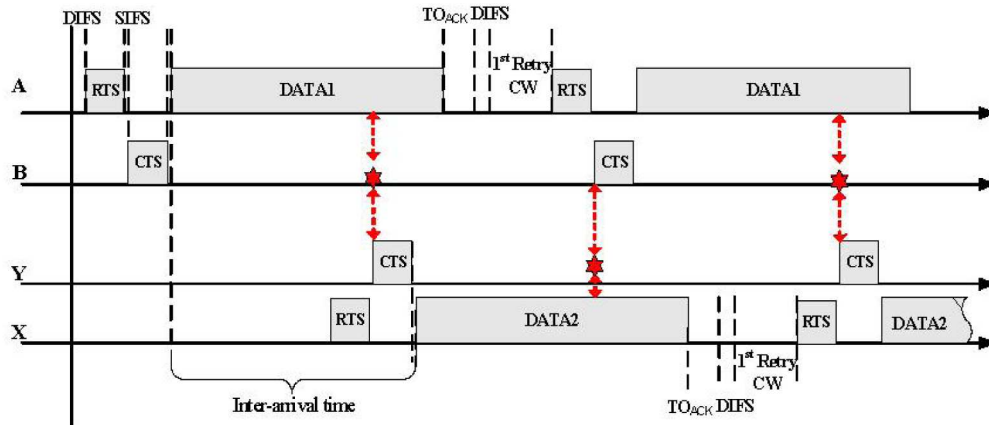


FIGURE 2.6 – Description du problème

Les auteurs proposent une solution en utilisant des paquets MRTS (Multicast RTS). Ces paquets contiennent la liste de nœuds voisins. Chaque nœud voisin répond un CTS dans l'ordre de la liste. Ainsi, un nœud voisin du voisin ne tentera pas d'émettre un RTS, et le CTS qui aura été répondu ne perturbera pas le paquet voulant être émis par le nœud d'origine. Le simulateur NS-2 est utilisé pour valider les améliorations proposées par ce mécanisme. Les résultats montrent une diminution du nombre de paquets perdus.

De nombreuses topologies mettent en évidence ce problème du nœud caché. Les auteurs de [RAGK08] proposent une modélisation des interactions entre deux flux dans un réseau WiFi maillé. Le modèle de propagation est basé sur SINR (Signal to Interference Noise Ratio). Dans ce modèle, un paquet ne peut être reçu correctement que si le rapport entre le signal et la somme des bruits est plus grand qu'un seuil nommé Capture Threshold. Le premier flux transite du nœud source S1 au nœud destination S2, et le second va du nœud source S2 au nœud destination D2. Les interactions modélisées sont classées en 10 catégories (figure 2.7).

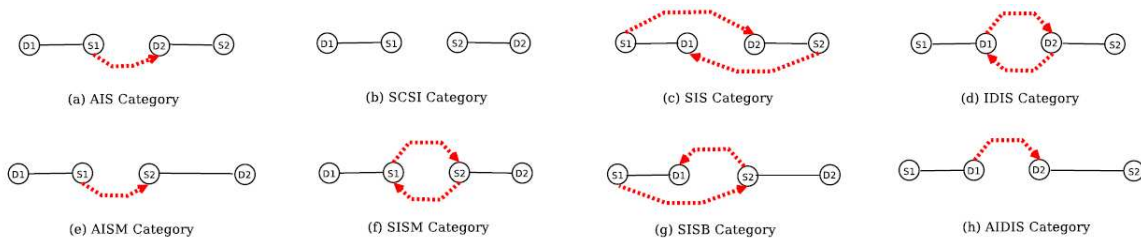


FIGURE 2.7 – Différents scénarios d'influence

Voici les 5 catégories principales.

- AIS (Asymmetric Incomplete State - Figure 2.7 a) : Cette catégorie représente le cas où un flux (émis par S1 dans l'exemple) interfère avec les paquets émis par

l'autre source car les nœuds sources ne se captent pas. Dans ce cas, D1 reçoit correctement les données alors que D2 reçoit des collisions.

- SCSI (Senders Connected Symmetric Interference) - Figure 2.7 b) : La catégorie représente le cas où les deux émetteurs se captent. La méthode d'accès CSMA/CA empêche un émetteur de décider d'émettre si l'autre est déjà en transmission. Les seules collisions inévitables, à faible probabilité, seraient dues à un tir de période de contention identique, ce qui pourrait entraîner une collision d'accusé de réception.
- SIS (Symmetric Incomplete State - Figure 2.7 c) : Cette catégorie représente deux nœuds sources qui ne se captent pas alors que les deux destinataires reçoivent les deux nœuds sources. De nombreuses collisions peuvent se produire car un émetteur ne sait pas si l'autre émet.
- IDIS (Interfering Destination Incomplete State - Figure 2.7 d) : Cette catégorie décrit la topologie pour laquelle il n'y a pas d'interférence entre une source et la destination de l'autre flux. Seules les deux destinations peuvent créer une collision entre les accusés de réception.
- SCAI (Senders Connected Asymmetric Interference) : Cette catégorie représente deux émetteurs qui sont à portée, et S1 qui détecte une transmission de D2 mais sans comprendre le paquet. Dans ce cas, S1 observe un temps intertrame EIFS plus long car il a détecté une erreur de transmission.

Ces différentes catégories montrent plusieurs topologies qui font apparaître le problème du nœud caché, qui a pour conséquence soit la perte de paquet, soit la perte d'accusé de réception.

Ce problème est aussi adressé dans [LPSY09]. En effet, les auteurs proposent une modélisation mathématique puis une expérimentation d'un réseau WiFi maillé de 6 nœuds en extérieur. Afin de moins être perturbées par l'environnement, les interfaces radios sont configurées sur des canaux 5 GHz suivant la norme 802.11a.

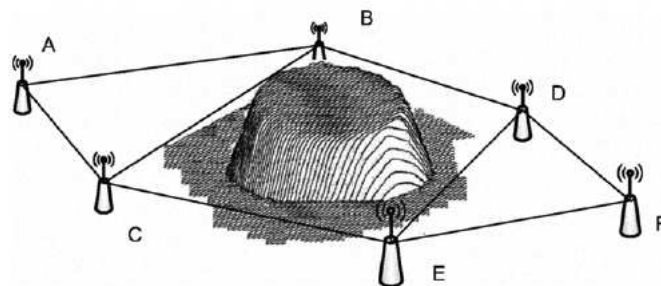


FIGURE 2.8 – Topologie présentant différents scénarios d'influence

La topologie utilisée est présentée dans la figure 2.8, et seuls les nœuds reliés peuvent communiquer entre eux. Les liens sont symétriques et le rapport signal sur bruit est de l'ordre de 9 à 22 dB, ce qui est relativement faible. Sur chaque paire de nœud AE, CD et BF, trois flux sont mis en place :

- un flux TCP à  $512 \text{ Kbit.s}^{-1}$  avec des paquets de 1500 octets en priorité BE (Best Effort),
- un flux UDP à  $28 \text{ Kbit.s}^{-1}$  avec des paquets de 200 octets en priorité VI (Video),
- un flux UDP à  $64 \text{ Kbit.s}^{-1}$  avec des paquets de 100 octets en priorité VO (Vo-IP).

Pour chaque communication, les débits minimum, moyen et maximum, le délai de bout en bout, le jitter et le taux de paquets perdus sont mesurés. Les résultats de l'expérimentation montrent que les flux vidéo et voix sur ip ont un taux de perte et un jitter acceptables. Cependant, les communications TCP ont un débit très faible. Ceci est dû d'une part à la classification 802.11e qui priorise les paquets VI et VO, et d'autre part au problème du terminal caché qui augmente la probabilité de collision. En conclusion, les auteurs suggèrent une nouvelle méthode d'accès qui serait normalisée dans la norme de WiFi maillé 802.11s. Celle-ci, nommée MCCA (Mesh Coordinated Channel Access) permettrait aux nœuds maillés de réserver des temps d'accès à la couche radio de manière périodique pour les applications nécessitant de la qualité de service.

### 2.2.3 Approches visant à améliorer la qualité de service de certains flux

Le WiFi maillé provoque des réémissions de paquet à chaque saut. C'est d'ailleurs le principe du WiFi maillé. Le canal WiFi est donc chargé, et la qualité de service se dégrade. Il peut donc être intéressant de chercher à améliorer la qualité de service des flux qui en ont besoin comme par exemple les communications audio ou vidéo.

Dans ce but, [SPK<sup>+</sup>07] propose une nouvelle méthode d'accès au médium, nommée Sticky CSMA/CA, ayant pour but d'améliorer la qualité de service des communications de type voix sur IP. Leurs expérimentations ont été faites avec du matériel compatible IEEE 802.11b seulement, avec et sans prendre en compte la norme de qualité de service 802.11e. Cette implémentation propose que chaque nœud mette à jour une table des flux périodiques dans le cas où un nœud détecte une transmission VoIP codée en G711. La taille des paquets ainsi que la période est fixée. Pour ajouter un nouveau flux périodique dans les tables de chaque nœud, le premier nœud émet un paquet nommé Real-time RTS (R-RTS). Celui-ci contient la durée du paquet périodique ainsi que l'identifiant du nouveau flux. Le destinataire vérifie sa table de flux afin d'accepter ou non ce flux, et répond par un paquet nommé Real-time CTS (R-CTS). Afin que ces flux soient prioritaires par rapport aux autres paquets, les auteurs modifient les paramètres de fenêtres de contentions  $CW_{min}$ ,  $CW_{max}$  et  $AIFS$  des paquets prioritaires. Les accusés de réception des paquets prioritaires ne sont pas envoyés car il n'est pas nécessaire de réémettre un paquet de voix dans un contexte temps réel. Périodiquement, un paquet de statistique sur chaque flux est transmis à la source du flux. Les auteurs proposent une solution au problème du terminal caché. Ils équipent leurs nœuds d'antennes sensibles et diminuent le seuil de détection de transmission afin d'éviter des transmissions simultanées.

Le WiFi maillé peut aussi transporter des paquets de vidéo sur ip. En effet, ce moyen de transport permet d'éviter l'installation d'un réseau filaire entre les caméras et le serveur vidéo. Ceci est présenté dans [KRR<sup>+</sup>09]. En effet, les auteurs ont développé la solution OmniEye, où chaque caméra est haute résolution et motorisée. Deux classes de flux sont donc présentes : le flux vidéo, et le flux de contrôle de la rotation et du zoom. Afin de garantir une qualité de service suffisante pour la vidéo, les auteurs ont mis en place le protocole TSAM (Time-Synchronised Application-level MAC protocol). Celui-ci est basé sur du TDMA, et permet d'allouer des temps de parole à chaque nœud du réseau. En effet, pour une caméra, tous les paquets vidéos ont la même taille. Il est donc prévisible de connaître le temps de parole qu'un nœud va nécessiter. Le problème

du nœud caché n'apparaît pas car tous les nœuds connaissent à l'avance le moment où ils ont le droit de parler. Les performances de cette solution sont meilleures qu'avec du WiFi standard, cependant tous les nœuds du réseau doivent implémenter ce protocole pour préserver cette qualité de service.

## 2.3 Positionnement

Cette première partie de l'état de l'art nous a permis d'identifier les problèmes connus sur un réseau WiFi maillé et quelques propositions de solutions soit radio soit protocolaires. Nous pouvons résumer ces problèmes dans une première partie. Malheureusement, les améliorations proposées précédemment ne sont pas compatibles avec notre approche. En effet, une contrainte imposée initialement est de rester compatible avec des stations qui respectent la norme et pourraient se connecter sur ces nœuds. Ainsi, une nouvelle méthode d'accès comme TDS présenté dans [TCCYMC06], ainsi que de nouveaux paquets de contrôle comme M-RTS présentée dans [JYM07] et R-RTS présentée dans [SPK<sup>+</sup>07] ne serait pas efficaces avec des stations ne respectant pas améliorations. De plus, une autre contrainte initiale est de n'utiliser qu'une interface radio dans un nœud, ce qui nous limite dans l'utilisation de plusieurs canaux. Cependant, nous avons pu identifier des leviers sur lesquels nous pouvons agir. Ceci va être présenté dans une seconde partie.

### 2.3.1 Identification des problèmes

Les problèmes présentés lors de transmission sur un réseau maillé peuvent être classés en plusieurs catégories, que nous rappelons dans les paragraphes suivants en se positionnant pour chaque cas.

#### 2.3.1.1 Partage du canal radio

Dans un réseau WiFi maillé utilisant un seul canal radio, tous les nœuds communiquent sur ce même canal. Cependant, des nœuds éloignés peuvent l'utiliser en même temps. Afin d'augmenter les performances du réseau, [KJCV06] et [JR05] proposent d'ajouter des interfaces physiques radios afin d'utiliser plusieurs canaux en même temps. Cependant, pour des raisons économiques, le produit que nous allons utiliser, n'a qu'une carte WiFi. C'est pourquoi nous allons contraindre notre étude à l'utilisation d'un seul canal radio.

Toutefois, diminuer le nombre de collisions et de pertes de paquets ainsi que optimiser les temps intertrames sont des pistes qui permettraient de laisser transiter plus de paquets sur le médium.

#### 2.3.1.2 Nœuds cachés

Chaque nœud ayant une portée limitée, il est probable que deux nœuds éloignés ne se captent pas et décident d'émettre en même temps. Si un des récepteurs est à portée de l'autre émetteur, il peut y avoir collision. De même, un accusé de réception peut aussi

être perdu, ce qui entraînera une nouvelle émission du paquet. [RAGK08] présente les topologies où il peut y avoir des collisions entre des nœuds cachés.

Les paquets de contrôle RTS/CTS prévus dans la norme 802.11 ont pour objectif de régler ce problème. Cependant, dans un réseau WiFi maillé, un nœud central qui reçoit tous les autres nœuds n'existe pas forcément. [JYM07] montre qu'une mauvaise réception d'un paquet de contrôle peut entraîner une succession de collisions. De plus, ce mécanisme induit des délais de contrôle supplémentaires.

Lors de notre étude, il est important de considérer une topologie où ce problème apparaît, car cette situation peut entraîner des pertes de paquets. De plus, l'activation de RTS/CTS est un levier qui peut améliorer les transmissions dans certains cas.

### 2.3.1.3 Baisse du débit utile

Les expérimentations ainsi que certaines publications [NNCA06] montrent que le débit utile diminue à chaque répétition de paquet par les différents nœuds traversés sur le réseau WiFi maillé. En effet, chaque nœud utilisant le même canal radio, celui-ci limite le débit utile.

Ce débit doit être partagé entre les différentes applications. Cependant, prioriser certains flux peut permettre d'obtenir un débit supérieur voir garanti pour ces applications.

### 2.3.1.4 Qualité de service

La norme de qualité de service 802.11e a apporté une classification de trafic, ce qui a amélioré la qualité de service des transmissions de voix sur IP et de vidéo. Cependant, dans le cadre du WiFi maillé, lorsque le canal radio est chargé, ceci n'est pas suffisant comme le montre [SPK<sup>+</sup>07].

Des mécanismes supplémentaires limitant l'engorgement, en priorisant les paquets nécessitant de qualité de service, ou en retardant les autres paquets sont des pistes qui permettent d'améliorer la qualité de service de bout en bout.

### 2.3.1.5 Influence des pertes de paquet WiFi sur une connexion TCP

La perte, suite aux multiples tentatives de réémission, d'un paquet TCP transmis sur un lien maillé WiFi, impacte l'ensemble d'un réseau maillé. Ceci est présenté dans [LKC05]. En effet, si un paquet n'est pas reçu par son destinataire, un accusé de réception TCP ne sera pas envoyé, le paquet sera retransmis à partir de sa source, et le réseau maillé sera à nouveau sollicité. Ainsi, de nombreuses pertes de paquet TCP vont entraîner de nombreuses retransmissions, et donc un trafic plus important. Nous pouvons donc nous soucier de la même façon des pertes de paquets sur un flux UDP qui dégrade la qualité de service, et des pertes de paquet sur un flux TCP qui entraîne des retransmissions de la couche transport.

## 2.3.2 Identification des leviers

Afin d'améliorer les transmissions sur un réseau WiFi, il est important d'identifier les leviers sur lesquels nous pouvons agir. Ceux-ci devront pouvoir être modifiables sur

du matériel existant, et doivent rester compatibles avec des stations quelconques.

Le matériel que nous utilisons permet de modifier les durées suivantes. Les durées AIFS pour chaque classe de qualité de service, ainsi que la durée DIFS lorsque la norme de qualité de service n'est pas utilisée, sont configurables. De plus, les entiers  $CW_{min}$  et  $CW_{max}$  pour chaque classe de qualité de service sont aussi configurables.

Nous pouvons également configurer l'activation ou non des paquets de contrôle RTS/CTS, ainsi que la taille minimale des paquets pour laquelle ce mécanisme est utilisé. En effet, il peut être judicieux de l'activer seulement pour les gros paquets, et de prendre des risques de pertes pour les petits paquets.

Le nombre de retransmission en cas d'échec est aussi configurable pour un paquet sans les mécanismes de contrôle RTS/CTS, mais aussi avec une valeur différente dans le cas de l'utilisation de ces mécanismes.

Enfin, la valeur du seuil de fragmentation des paquets est aussi configurable.

### 2.3.3 Problème de l'expérimentation

L'évaluation de l'impact de ces leviers sur le système pourrait être réalisée par expérimentation. En effet, le plus grand avantage serait que les valeurs mesurées seraient des valeurs réelles.

Cependant, cette approche présente de nombreux inconvénients. Le problème du nœud caché n'apparaît que si les nœuds sont éloignés, ce qui nécessite de grands espaces car la portée des nœuds est de l'ordre de la centaine de mètres. Sur de telles surfaces, l'environnement est difficilement maîtrisable et des perturbations hyperfréquences peuvent altérer l'expérimentation. De plus, dater le départ et l'arrivée de chaque paquet dans le but de mesurer le délai de parcours au travers du réseau maillé implique que les nœuds soient précisément à la même heure. Cette précision devant atteindre la centaine de microsecondes, il n'est pas possible de synchroniser les nœuds à cette échelle. Enfin, les interfaces en mode "espion" ne capturent que les paquets que leurs antennes peuvent recevoir et qui sont correctement transmis. Ceci signifie que, sur une grande surface, plusieurs nœuds devront enregistrer ce qu'ils captent puis il sera nécessaire de synchroniser toutes ces données. Pour finir, ceci limite l'analyse macroscopique de l'interaction de plusieurs paquets provenant de nœuds différents à un même moment.

Cependant, elle reste une bonne méthode pour mesurer réellement l'impact des améliorations à travers les leviers identifiés.

### 2.3.4 Simulation

L'approche que nous allons retenir est la simulation. Celle-ci permettra d'évaluer l'action des leviers identifiés sur de nombreuses topologies, tout en maîtrisant les interférences dues à l'environnement. De plus, les scénarii seront enregistrés, et une vue macroscopique des événements pourra être analysée.

Notons que le recours à la simulation est à double objectif :

- avant déploiement, chercher la position de chaque nœud, les liens à utiliser entre eux, et le paramétrage offrant la meilleure qualité de service,
- en post-déploiement, chercher à améliorer la qualité de service en actionnant les leviers à partir d'une topologie existante et des mesures réalisées sur le terrain.



Présentons maintenant les différents simulateurs existants.

## 2.4 Simulateurs existants

### 2.4.1 Simulateurs réseaux classiques

De nombreux simulateurs réseaux existent. Nous allons présenter les principaux, à savoir NS-2, NS-3, OPNET Modeler, OMNET++ et QualNet.

#### 2.4.1.1 NS-2

Le logiciel NS "Network Simulator" [NS-11a] est gratuit et open source. Il a été conçu pour simuler des réseaux à partir d'évènements discrets. L'utilisateur peut définir des modules en C++. L'interaction entre les différents éléments, la configuration du simulateur et la programmation temporelle d'évènements sont décrits en TCL. Lors d'une simulation, ce logiciel fournit un fichier animation exploitable avec l'utilitaire NAM qui va afficher la topologie, les flux, ainsi que les évènements qui se sont produits lors de la simulation. Une analyse en détail de cette animation peut aider à la validation du réseau modélisé sous NS. Ce logiciel fournit aussi des fichiers exploitables par XGraph permettant de tracer des courbes représentant les performances à analyser. Enfin, il fournit aussi des fichiers de log, à partir desquels une application développée par l'utilisateur peut extraire d'autres informations.

De nombreux protocoles et modules ont déjà été développés et mis à disposition par la communauté. Cependant, l'objectif initial de ce logiciel est la simulation des couches hautes du modèle OSI sur des réseaux filaires. Des couches physiques sans fil ont été développées par différents contributeurs, avec des manières différentes de programmer et ne peuvent pas toujours être personnalisées sans en modifier les sources [MSK<sup>+</sup>10]. De plus, la modélisation d'une couche physique radio correspondant au WiFi maillé est complexe. En effet, l'émission d'un nœud ne doit être reçue que par certains nœuds qui sont à portée et plusieurs utilisations du canal radio sont possibles au même instant sur des nœuds éloignés. Ce simulateur a été créé initialement pour simuler des réseaux filaires où les problèmes de propagation et de collision sont différents. L'utilisation de cet outil nécessiterait de revoir la couche radio ainsi que la répétition des paquets par les nœuds relais, c'est pourquoi nous ne l'utiliserons pas.

Cependant, aujourd'hui, de nombreux contributeurs ont permis à ce simulateur d'évoluer, ce qui nous permettrait de l'utiliser. Par exemple, [BMG<sup>+</sup>10] améliore la couche physique pour permettre de simuler plus des réseaux maillés, avec différentes technologies dans la même simulation.

#### 2.4.1.2 NS-3

Le successeur du NS-2 est NS-3 "Network Simulateur 3" [NS-11b]. Ce logiciel est aussi gratuit et open source. Contrairement à son prédécesseur, il est écrit uniquement en C++, ce qui simplifie le développement de nouveaux modules. De plus, il utilise des logiciels automatisant la documentation, ce qui permet une prise en main plus facile

du logiciel. La simulation de réseaux filaires et non filaires a été prévue dès l'origine du projet. C'est pourquoi le code est plus intuitif et plus évolutif.

Des études ont utilisé ce simulateur dans le cadre des réseaux maillés comme [YJSHYB<sup>+</sup>11] qui propose un algorithme de routage orienté session, et [GGFK10] qui propose un protocole de multicast. Cependant, ce simulateur est récent et ne permettait pas notre étude au moment où nous cherchions un simulateur.

#### 2.4.1.3 OPNET Modeler

Le logiciel OPNET Modeler [OPN11] a été développé et commercialisé par OPNET Technologies Inc. C'est un simulateur réseau payant. Sa bibliothèque comporte de nombreux modèles de matériel existant, ainsi que de nombreux protocoles. L'utilisation des modules existants se fait de manière graphique, mais il est possible de développer ses propres modèles en C++. Il est tout d'abord nécessaire de définir la topologie. Ensuite, chaque nœud peut être configuré. Enfin, chaque flux et protocole sont définis. Ce simulateur réseau est orienté transmission radio et devrait permettre la modélisation de notre couche radio.

Cependant, la fonctionnalité de répéteur, qui est fondamentale dans le WiFi maillé n'était pas implémentée au moment où nous avons choisi l'outil de simulation. De même, certaines fonctionnalités du protocole 802.11e n'étaient pas implémentées. Enfin, la description de la topologie et des flux à injecter est principalement graphique, ce qui nécessite une forte reconfiguration pour analyser une amélioration sur des topologies différentes. C'est pourquoi nous n'avons pas choisi ce logiciel. Aujourd'hui, de nombreuses améliorations y ont été apportées, notamment à propos des réseaux maillés sans fil, car cette technologie est de plus en plus utilisée.

#### 2.4.1.4 OMNET++

OMNET++ [Var11] est un simulateur à événements discrets. Il a tout d'abord été conçu pour simuler les communications sur un réseau [VH08], mais son architecture permet la simulation de systèmes plus complexes [Var01]. Ce logiciel est libre pour les universités et applications à but non lucratif, il est soumis à licence pour les entreprises. Son environnement de développement est basé sur la plateforme du logiciel Eclipse. Ses modules sont développés en C++. Cependant, ce logiciel n'implémente que la couche physique 802.11b, et non des débits plus élevés proposés par la norme 802.11a ou 802.11g. De plus, il n'implémente pas de réseau sans fil maillé, ni de réseau ad-hoc qui aurait pu servir de base pour développer un réseau maillé. C'est pourquoi nous ne retiendrons pas cette solution.

#### 2.4.1.5 QualNet

La société Scalable Network Technologies commercialise le logiciel QualNet [nt11]. Il permet de simuler des réseaux filaires et sans fil. Son architecture est divisée en trois couches. Tout d'abord, la première couche est le noyau qui est un ordonnanceur à événement discret. Ensuite, des bibliothèques de modèles sont disponibles. Enfin, la troisième couche est l'interface utilisateur. Celle-ci est composée de différents outils permettant de décrire les scénarii, d'obtenir les animations ou des analyses.

Cependant, ce simulateur n'implémentait pas la norme de qualité de service sur réseau WiFi 802.11e au moment où nous cherchions un simulateur. C'est pourquoi nous n'avons pas retenu cet outil.

#### 2.4.1.6 Bilan

Au moment où nous cherchions un simulateur réseau permettant l'étude de réseau maillé sans fil, les simulateurs existants n'implémentaient pas ce mode de fonctionnement qui consiste à acheminer un paquet de proche en proche, ainsi que de partager un même canal radio pour des nœuds qui ne sont pas tous à portée les uns des autres.

De plus, la norme de qualité de service sur réseau maillé WiFi 802.11e est récente et n'était pas non plus complètement implémentée. Elle fournit des paramètres de configuration sur lesquels des actions nous permettraient de jouer sur la qualité de service. Ces paramètres doivent donc être configurables.

Cette absence d'outil nous amène à développer notre propre simulateur.

#### 2.4.2 Choix du formalisme de modélisation

Le formalisme de modélisation doit permettre de modéliser un réseau maillé WiFi. De plus, il doit donner des résultats qualifiant la qualité de service. Enfin, il doit permettre de détailler toutes les étapes du scénario simulé afin de mettre en avant les phénomènes significatifs pour l'amélioration de la QoS.

Nous avons choisi le réseau de Petri coloré, temporisé, stochastique et hiérarchique comme formalisme de modélisation car il est adapté pour la représentation du fonctionnement d'un réseau maillé WiFi. En effet, le réseau de Petri nous permet de représenter les choix selon des conditions, la sérialisation et le parallélisme. L'aspect coloré du réseau de Petri permet de singulariser des jetons afin de représenter des paquets différents, des nœuds différents, des classes de qualité de service différentes sur un seul réseau de Petri. Ceci permet de modifier la topologie et la configuration sans changer le modèle comme décrit dans la section suivante. Le réseau de Petri temporisé nous permet de faire intervenir le temps dans la modélisation. La norme WiFi met en œuvre des délais d'attente, et la transmission de paquets a une durée définie. Enfin, le réseau de Petri stochastique fait intervenir des valeurs aléatoires dans le modèle, comme la durée des périodes de contention avant d'émettre un paquet, ou des probabilités d'altération de paquets lorsque le canal WiFi subit des interférences. Pour finir, le réseau de Petri hiérarchisé permet une construction incrémentale du modèle. Les parties sur lesquelles nous avons besoin de détails pourront être raffinées de manière à enrichir le modèle. De plus, modularité et raffinement permettent de réduire la taille des réseaux, d'en simplifier tant la construction par composition que la lecture.

Le but de cette modélisation est de construire un modèle permettant de simuler différents scénarii. Cette simulation doit, à terme, fournir des informations permettant de qualifier ou de quantifier les améliorations du dispositif (portées sur le modèle) afin de les valider. Ainsi, l'animation pourra mettre en avant des événements élémentaires qui affectent la qualité de service. Une fois ces événements connus, nous pourrons mettre en place de nouveaux mécanismes pour améliorer le système.

### 2.4.3 Joueurs de réseaux de Petri

Afin de comparer les actions des leviers configurés dans le modèle, un joueur de réseau de Petri est nécessaire. Ce joueur doit permettre de valider la modélisation, et de simuler des transferts de données dans le but de comparer les performances et la qualité de service en fonction de différentes actions sur les leviers. Voici donc les joueurs de réseaux de Petri que nous avons analysés.

#### 2.4.3.1 Möbius

Le projet Möbius [SKGR11] est hébergé par l'Université de l'Illinois aux Etats-Unis. Il a été développé à partir de 2001, et est devenu le successeur de UltraSAN [OQDS06]. C'est un joueur de réseau de Petri à objet, temporisé, stochastique et hiérarchisé. Ce simulateur permet de modéliser des systèmes complexes à événements discrets. Les éléments de bases sont les places, les transitions, et les portes d'entrée et de sortie des transitions. Il existe deux types de places. Les places "standard" ne peuvent contenir que des jetons banalisés alors que les places "extended" peuvent contenir des objets. Chaque jeton est soit banalisé soit un objet. Les portes d'entrée des transitions exécutent des tests sur ces objets, et des méthodes sur ces objets sont définies dans les portes de sortie des transitions. L'utilisation de ce logiciel est donc lourde lorsque l'utilisation d'objet n'est pas nécessaire. Pour chaque modèle, un code en C++ est généré puis exécuté. Les primitives "Rep" et "Join" permettent de créer des instances de modèles et de dupliquer ces instances.

[MPV06] a utilisé cet outil pour modéliser un réseau WiFi dans le but d'analyser la qualité de service apportée par la norme 802.11e. Le modèle ainsi créé est simple sur un aspect réseau de Petri. Mais, la définition des objets, ainsi que leurs utilisations dans les portes d'entrée et de sorties sont complexes. La modélisation est donc difficile à lire et à concevoir. Nous avons utilisé cet outil dans un premier temps, mais l'évolutivité d'une telle modélisation est limitée.

#### 2.4.3.2 GreatSPN

Le département d'informatique de l'Université de Turin en Italie a créé le joueur de réseau de Petri GreatSPN [Gre11] qui signifie "GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets". Il permet de modéliser, valider, et effectuer de l'évaluation de performance de systèmes distribués à partir de réseau de Petri Stochastique avec l'extension colorée et temporisée. La structure du logiciel est modulaire, et permet de créer des extensions programmées en langage C. L'interface graphique fonctionne sous OSF, et permet une édition du modèle, une animation interactive de la simulation, des menus interactifs avec des outils d'analyse. Cependant, certaines de ces fonctionnalités ne sont pas disponibles pour des réseaux de Petri colorés et cet outil ne fonctionne que sous SunOS, Mac OS X, ainsi que sous Linux RedHat, Slackware, Mandrake et SuSe.

### 2.4.3.3 CPN-Tools

Le logiciel CPN-Tools [WV11], créé par le département Daimi de l'Université d'Aarhus au Danemark permet de simuler les réseaux de Petri colorés temporisés stochastiques hiérarchiques. De plus, il permet d'enregistrer les événements qui se sont produits au cours de la simulation afin, d'une part de valider que le modèle fonctionne conformément aux spécifications (dont la norme 802.11), et d'autre part de faire apparaître des situations qui entraînent une dégradation ou une amélioration de la qualité de service (en réponse à nos actions sur les leviers).

## 2.5 Modélisation et Réseaux de Petri

Une modélisation par réseau de Petri de la communication en WiFi a été réalisée dans [MPV06]. En effet, les auteurs proposent un modèle permettant l'évaluation de la norme de qualité de service 802.11e. L'utilisation de SPN (Stochastic Petri Nets) est fondée sur les raisons suivantes : tout d'abord, la description est graphique, ce qui la rend intuitive ; ensuite, la représentation d'un système complexe peut être compacte ; enfin, le formalisme est simple, et peut représenter des équations algébriques complexes. Le formalisme est indépendant du logiciel de simulation, et il existe de nombreux joueurs de réseaux de Petri. Le code C++ étant un peu lourd, les auteurs se limitent à ne présenter que la représentation graphique du réseau de Petri.

Le modèle est divisé en trois sous-parties : file d'attente, traitement, et séquence de transmission. Une classification de paquet en quatre files d'attente correspondantes aux types de paquets (AC) se fait en entrée du premier sous modèle. L'utilisateur peut configurer la distribution aléatoire représentant les dates de génération de paquets. Ensuite, la sous-partie traitement applique les délais d'attente en fonction de l'état du canal radio, en suivant la norme 802.11e. Enfin, la sous partie-séquence de transmission présente l'émission et l'attente d'accusé de réception. Ces sous-parties sont regroupées pour modéliser un nœud. Celui-ci est dupliqué pour représenter plusieurs nœuds. Chaque nœud est lié à un modèle décrivant le canal de propagation. Celui-ci est unique et modélise un canal radio commun à tous les nœuds. En effet, les auteurs considèrent que tous les nœuds sont à portée.

La simulation de ce modèle permet aux auteurs de faire des mesures de performance car le débit sur un tel réseau peut être calculé en comptant le nombre de paquets transmis en un temps donné. De plus, les nombres de collisions, perturbations ou pertes au niveau radio peuvent être comptés dès lors que ces événements correspondent à un tir de transition.

Ce modèle basé sur les réseaux de Petri est trop simplifié et la complexité est intégrée en C++ dans les transitions. Il est donc difficilement évolutif. De plus, la couche radio est commune à tous les nœuds et ne permet pas de communications simultanées pour des nœuds éloignés, ce qui est nécessaire pour la modélisation d'un réseau WiFi maillé. Cependant, ce simulateur fonctionne et le modèle a été réalisé à base de réseaux de Petri.

## 2.6 Conclusion

Ce chapitre nous a permis dans un premier temps de mettre en avant les problèmes traités dans la littérature à propos des réseaux WiFi maillés. Ces problèmes sont dus au partage du canal radio entre les différents nœuds du réseau qui ne sont pas tous à portée. De plus, en fonction de la topologie, le problème du nœud caché peut apparaître. La qualité de service et le débit utile sont affectés, ce qui impacte les communications audio et vidéo sur un tel réseau.

A partir de ces problèmes et des difficultés à utiliser l'approche expérimentale, nous optons pour la simulation. C'est pourquoi ce chapitre a décrit les principaux simulateurs utilisés en réseau. Cependant, ils ne sont pas adaptés à notre problématique. Ainsi, nous avons décidé d'utiliser le formalisme des réseaux de Petri, exécuté par l'intermédiaire d'un joueur. Il permet la modélisation d'un réseau WiFi maillé, mais permet aussi une forte évolutivité. Nous pouvons alors raffiner à mesure à la vue du détail nécessaire pour observer des problèmes et tester des améliorations à apporter.

Nous avons ensuite exposé et comparé différents joueurs de réseaux de Petri permettant de simuler un ensemble de flux, sur une topologie donnée, au travers du modèle représentant le réseau WiFi maillé.

Le chapitre suivant est consacré à la description de ce modèle de réseau WiFi maillé à partir de réseaux de Petri, ainsi qu'à sa validation.



## Chapitre 3

# De la modélisation à la validation du simulateur

### 3.1 Introduction

Pour améliorer la qualité de service des flux sur un réseau maillé WiFi, il est nécessaire de comprendre comment fonctionnent les mécanismes d'accès de la norme 802.11 et d'analyser ce qu'il se passe lorsque plusieurs points d'accès utilisent ce même canal de communication. C'est pourquoi une modélisation, puis une simulation du modèle peut permettre d'identifier les situations de dégradation de la QoS et de mettre en avant des mécanismes à améliorer.

Ensuite, ce simulateur permet d'évaluer les actions portées sur des leviers qui peuvent modifier la qualité de service. Ces actions peuvent être portées sur les leviers de certains nœuds, et de manière différente en fonction de la topologie. Cependant, ces leviers devront être modifiables sur le matériel utilisé afin de pouvoir apporter les améliorations sur le terrain. De plus, les améliorations apportées devront rester compatibles avec les transmissions d'autres nœuds présents dans la même installation et que nous ne pouvons pas modifier.

Tout d'abord, nous allons présenter l'objectif de la modélisation et le formalisme utilisé qui est le réseau de Petri. Ensuite, nous allons spécifier la granularité du modèle requise par notre objectif et définir les limites d'utilisation de celui-ci. Enfin, le modèle sera présenté, module par module. Cependant, avant d'exploiter le simulateur, il est nécessaire de le valider, afin de s'assurer de sa conformité aux phénomènes réels et ainsi, obtenir des résultats qualitativement cohérents. La pertinence des conclusions tirées de la simulation en dépend.

Le formalisme de modélisation doit permettre de modéliser un réseau maillé WiFi. Au regard de notre objectif, il doit donner des résultats qualifiant la qualité de service. De plus, il doit permettre de détailler toutes les étapes du scénario simulé afin de mettre en avant les phénomènes significatifs pour l'amélioration de la QoS.



### 3.1.1 Objectif de la modélisation

La modélisation dépend de l'objectif de la simulation. Notre objectif est de mettre au point un simulateur permettant de configurer :

- le nombre, le type et les paramètres des leviers de chaque nœud,
- la topologie, en spécifiant les qualités de transmission, les débits physiques, l'atteignabilité entre les nœuds,
- les chemins que les paquets doivent emprunter pour traverser le réseau maillé,
- le type, la taille, la quantité, la périodicité, la source et la destination des flux à simuler.

Ce simulateur doit respecter la norme 802.11, avec :

- le mécanisme de classification de paquet,
- le respect des délais interframes,
- le calcul aléatoire de période de contention,
- l'activation, en fonction de la taille des paquets et du seuil configuré, des mécanismes de réservation du médium RTS/CTS,
- l'émission d'accusé de réception suite à la réception d'un paquet,
- la réémission en cas de non réception d'accusé de réception,
- la mémorisation des numéros de paquet afin d'ignorer ceux reçus en double.

La couche radio doit correspondre à celle du WiFi maillé, ce qui signifie que :

- chaque nœud ne reçoit que les transmissions des nœuds qui sont à portée,
- plusieurs transmissions éloignées peuvent réussir en même temps.

De plus, ce simulateur doit permettre aux nœuds de :

- relayer des paquets en suivant le principe du WiFi maillé,
- ignorer les paquets à relayer si la file d'attente d'émission est pleine.

Enfin, le simulateur doit fournir des informations permettant de qualifier ou de quantifier les améliorations du dispositif (portées sur le modèle) afin de les valider. De fait, il est essentiel de pouvoir mettre en place des indicateurs (traces), voire même de les modifier selon les scénarii ou les points précisément étudiés. Naturellement, la simulation doit mettre en avant les phénomènes qui influencent la qualité de service. Une fois ces phénomènes connus, nous pourrions mettre en place de nouveaux mécanismes pour améliorer le système.

La simulation doit donc fournir :

- une trace temporelle de tous les événements afin de pouvoir reconstituer le scénario,
- les délais de parcours du réseau de chaque paquet,
- le nombre de collisions, de pertes et de paquets correctement transmis,
- l'évolution du nombre de paquets en file d'attente,
- l'évolution du temps que passe un paquet en file d'attente sur chaque nœud.

### 3.1.2 Présentation du formalisme de modélisation

Les différentes couleurs sont définies au cours de la présentation du modèle.

### 3.1.2.1 Unité de temps

Afin de compter le temps avec des valeurs entières, ce qui est imposé par l'outil utilisé, une unité de temps correspondra à une microseconde. La plus petite durée utilisée dans la norme 802.11 est *aSIFSTime* qui a une durée de 10  $\mu$ s.

### 3.1.2.2 Place

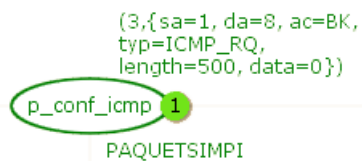


FIGURE 3.1 – Exemple de place

Une place est définie par un nom (*p\_conf\_icmp* dans l'exemple de la figure 3.1) et la couleur des jetons qu'elle accepte (*PAQUETSIMPI*). Elle peut aussi avoir un marquage initial (*3,sa=1, da=8, ac=BK, typ=ICMP\_RQ, length=500, data=0*) et le nombre de jetons qu'elle contient à cet instant de la simulation (1). Il s'agit en l'occurrence de la description de l'initialisation des paquets ICMP à travers le marquage initial du modèle.

### 3.1.2.3 Transition



FIGURE 3.2 – Exemple de transition

Une transition est présentée sur la figure 3.2. Elle a au minimum un nom (*t\_emission*). Elle peut aussi avoir une condition qui porte sur les jetons entrants, et une temporisation (*@+1*). Cette temporisation signifie que le jeton sera disponible dans la place suivante au temps courant plus 1 unité de temps. Ceci correspond au modèle de réseau de Petri T-Temporisé.

### 3.1.2.4 Arc



FIGURE 3.3 – Exemple d'arc

Un arc est présenté figure 3.3. Un arc entrant relie une place à une transition. Sa fonction de filtrage associée permet d'affecter à des variables les valeurs du jeton candidat au tir. Si le texte contient des constantes, elle permet de filtrer seulement

les jetons qui contiennent une valeur égale à la constante. Un arc sortant relie une transition à une place. Sa fonction associée peut permettre de générer et/ou modifier un ou plusieurs jetons déposés dans la place cible à laquelle l'arc est relié.

### 3.1.2.5 Macro transition

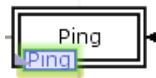


FIGURE 3.4 – Exemple de macro transition

Les transitions ayant un double contour, comme présenté figure 3.4, enferment un sous-réseau de Petri du nom de la macro transition. Les macro transitions permettent une description hiérarchisée. Les places liées par des arcs à ce type de transitions seront présentes à la fois dans ce modèle et dans le sous modèle.

### 3.1.2.6 Place d'entrée dans une macro transition

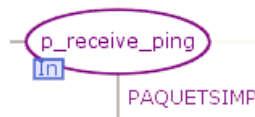


FIGURE 3.5 – Exemple de place d'entrée

Une place marquée *In* dans un sous réseau est une place d'entrée. Un exemple est présenté figure 3.5. Cela signifie que cette place existe dans le modèle parent, et que les jetons ne font que rentrer dans cette place. Lorsque la place à laquelle elle est reliée dans le modèle parent est marquée, celle-ci le sera aussi.

### 3.1.2.7 Place de sortie dans une macro transition



FIGURE 3.6 – Exemple de place de sortie

Une place marquée *Out* dans un sous réseau est une place de sortie (figure 3.6). Les jetons qui y seront placés seront disponibles dans la place du réseau parent avec laquelle elle est liée.

### 3.1.2.8 Place d'entrée/sortie dans une macro transition



FIGURE 3.7 – Exemple de place d'entrée sortie

Une place marquée *I/O* dans un sous réseau est une place d'entrée/sortie (figure 3.7). Les jetons peuvent entrer ou sortir de cette place qui est commune avec une place du modèle parent.

## 3.2 Structuration du modèle

### 3.2.1 Une approche modulaire

Face à la complexité du système considéré, celui-ci va être découpé en plusieurs parties afin de faciliter la construction du modèle, ainsi que son évolution.

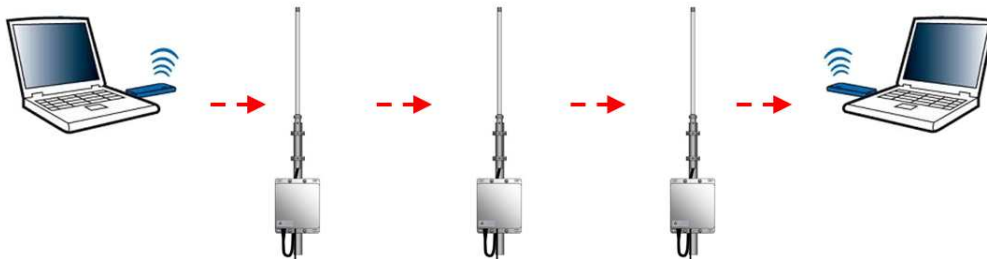


FIGURE 3.8 – Exemple de transmission sur un réseau maillé

La transmission, sur un réseau WiFi maillé, peut être exprimée sur un plan topologique ; chaque nœud a une position déterminée et ne peut communiquer qu'avec certains autres nœuds (figure 3.8). L'aspect coloré du formalisme de réseau de Petri que nous avons choisi permet de modéliser le comportement d'un nœud à partir de la structure du modèle et de représenter plusieurs nœuds à partir du marquage du modèle. De cette manière, le nombre de nœuds est donc facilement configurable. Chacun d'eux peut être une station ou un point d'accès. Ceci est aussi configurable.

La topologie est donc décrite par des jetons colorés, ce qui facilite la mise en place de différentes simulations. Le modèle est donc évolutif au sens de la topologie réseau qu'il représente. Ceci est important dans notre cas car le simulateur doit être exploitable sur différentes installations déployées ou à déployer par l'entreprise.

La transmission peut aussi être décomposée en fonction de la pile protocolaire, en suivant les couches du modèle TCP/IP. Pour une transmission qui doit aller d'un nœud 1 à un nœud 5 (figure 3.9) et devant être relayée par les nœuds 2, 3 et 4, les paquets sont encapsulés à chaque traversée de couche sur le nœud 1, chaque nœud intermédiaire

traite la retransmission au niveau de la couche Liaison, et enfin le nœud 5 désencapsule les données pour les délivrer à la couche Application.

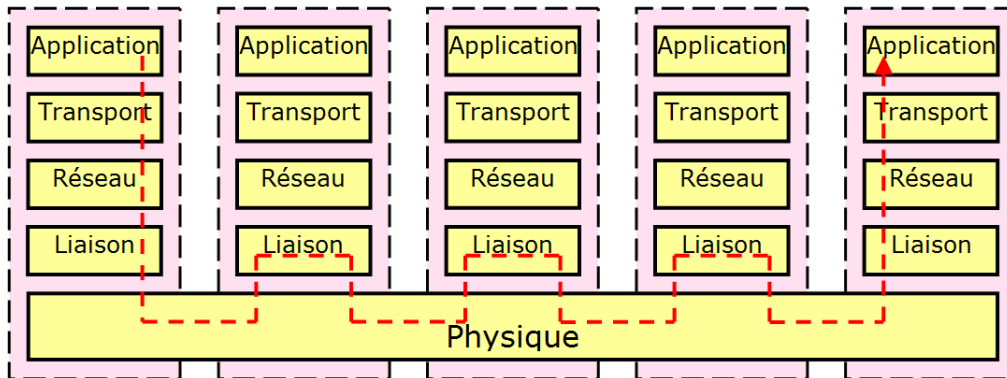


FIGURE 3.9 – Exemple de transmission sur le modèle Internet

Le modèle est donc explicitement structuré selon ces couches. Afin de détailler davantage la couche liaison de données, elle est divisée en sous couche MAC et LLC. La sous couche MAC modélise les mécanismes propres à la norme WiFi, avec les mécanismes d'accès au médium, alors que la sous couche LLC est utilisée pour la répétition de paquet, et le choix des chemins.

Les couches Réseau et Transport ne sont pas modélisées en détail car un paquet aura un nœud source et un nœud destination identifié au niveau MAC. Les notions d'adresses IP et de routage ne sont pas utiles dans le cadre de notre étude, néanmoins le modèle pourrait être enrichi si besoin est.

Enfin, deux applications permettant de générer des données ICMP et UDP seront implémentées. Un premier module nommé ICMP permet de générer des requêtes périodiquement. Quand une requête atteint sa destination, une réponse est émise vers la source de la requête. Ainsi, le RTT peut être mesuré au travers du réseau maillé. Ce protocole permet tout d'abord de valider le fonctionnement du modèle, car un paquet simple est facile à suivre. De plus, ICMP permet de mesurer le RTT en même temps que d'autres transmissions transitent sur le réseau maillé.

Un second module générant des paquets est inscrit dans l'architecture du simulateur, il s'agit du module UDP. Comme le protocole du même nom, celui-ci permet de générer des paquets en spécifiant :

- le nombre,
- la date de mise en file d'attente d'émission,
- la classe de qualité de service,
- la taille,
- les nœuds sources et destination.

Ainsi, différents types de trafic peuvent être générés avec ce module, qu'il soit périodique ou sporadique, représentant de la voix sur ip ou du téléchargement.

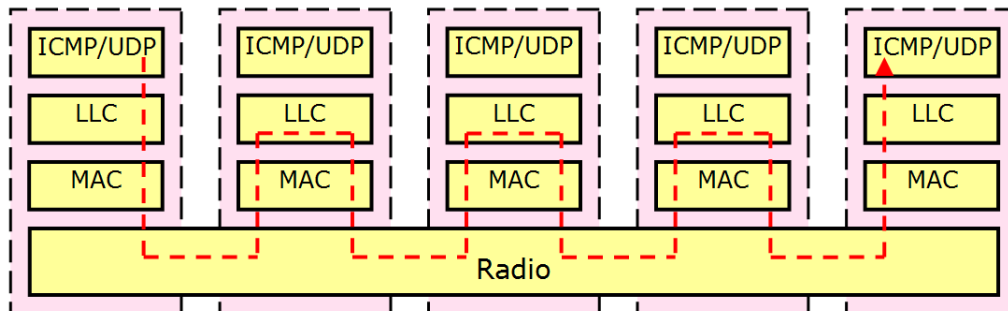


FIGURE 3.10 – Transmission sur chaque nœud

Chaque nœud étant singularisé par une couleur différente, il n'est nécessaire de représenter qu'un seul modèle comportemental pour chaque couche, et celui-ci sera utilisé pour tous les nœuds grâce au principe du repliage et de la coloration. Ainsi, à partir de la figure 3.10, nous obtenons le modèle replié présenté figure 3.11.

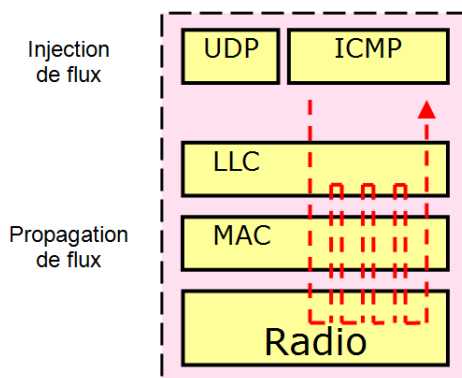


FIGURE 3.11 – Repliage du modèle de la transmission

### 3.2.2 Présentation des modules

Le modèle général est nommé *Network*. Il a été raffiné au fur et à mesure de sa construction comme le présente la figure 3.12. Il est décomposé en plusieurs modules :

- Le modèle *ICMP* modélise une émission de requête périodique, comme le fait le programme ping. Le principe requête/réponse de ce protocole nous permet de mesurer des temps aller-retour (RTT). Cette fonctionnalité simple nous permet dans un premier temps de valider les principes de base du modèle : génération de paquet, transmission sur la couche radio, délais intertrames, période de contention, émission d'accusé de réception...
- Le modèle *UDP* décrit des émissions ponctuelles, avec différentes caractéristiques de paquet (qualité de service, temps nécessaire pour l'émettre...) de manière à injecter des trafics particuliers dans le modèle. Ceci permet de simuler d'une part des émissions périodiques de petits paquets comme le ferait un flux audio, et

- d'autre part des transmissions de nombreux gros paquets comme le ferait un transfert de fichier.
- Le modèle *LLC* représente la partie haute de la couche liaison. C'est au sein de cette couche qu'est effectué le choix du chemin pour chaque paquet ainsi que la répétition de paquet.
  - Le modèle *MAC* est spécifique au WiFi et implémente la partie basse de la couche liaison, avec la méthode d'accès, la gestion d'accusé de réception et l'émission périodique de beacon.
  - Le modèle *Radio* décrit la couche physique.

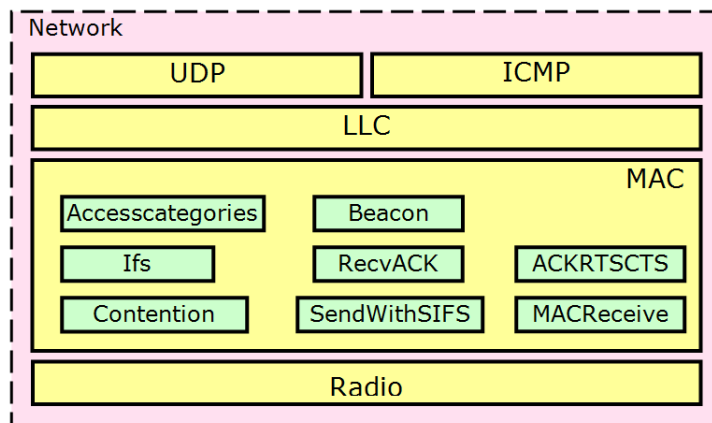


FIGURE 3.12 – Présentation des éléments du modèle

Au regard de l'influence du protocole MAC sur les échanges entre nœuds, le modèle *MAC* est le plus complexe. Il est aussi divisé en plusieurs modules :

- *Beacon* permet la diffusion périodique de beacon pour les nœuds de type point d'accès.
- *AccessCategories* modélise les quatre files d'attente de la norme 802.11e : VO (Voix), VI (Vidéo), BE (Best Effort), et BK (Background).
- *Ifs* impose aux paquets les délais intertrames correspondants.
- *Contention* décrit le tir ou non de période de contention ainsi que sa durée puis temporise le paquet en conséquence. De plus, il surveille l'utilisation du canal radio et réinitialise l'émission si une transmission est en cours comme le présente le mécanisme CSMA/CA.
- *RecvAck* gère les réceptions d'accusé de réception et décide de réémettre un paquet qui n'a pas été acquitté.
- *MACReceive* décrit la réception de paquet. En effet, ceci impacte le prochain délai intertrame. Ensuite, il ignore les paquets qui ne sont pas destinés à ce nœud. Puis il achemine les RTS, CTS et ACK au module ACKRTSCTS. Enfin, il propose les paquets de données à la couche LLC.
- *SendWithSIFS* permet d'émettre des paquets de contrôle à la suite d'un temps intertrame SIFS.
- *ACKRTSCTS* gère l'utilisation des paquets de contrôle RTS/CTS s'ils sont activés, ainsi que l'émission d'accusé de réception.

### 3.3 Granularité

La granularité d'un modèle dans le temps et dans l'espace est fonction de ce que l'on souhaite observer et valider. Cette granularité a des conséquences sur la pertinence et sur la confiance que nous pouvons avoir sur les résultats ainsi que sur leur exploitation dans la mesure où la granularité retenue définit le domaine de validité du simulateur ainsi que les niveaux d'observation et de contrôle. Par exemple, un modèle à gros grain pourrait masquer certains phénomènes ou ne pas offrir les leviers sur lesquels une amélioration potentielle pourrait être exploitée. Cette granularité a été étudiée sur les différentes couches protocolaires qu'encapsule le simulateur.

#### 3.3.1 Granularité au niveau de la couche radio

La particularité du WiFi maillé dans notre cas par rapport au WiFi standard est que les différents acteurs du réseau n'utilisent qu'un seul et même canal radio. Cependant, tous les nœuds ne se captent pas entre eux. Le modèle doit donc représenter l'utilisation du canal radio pour chaque récepteur. De cette manière, une communication est possible entre deux nœuds au même moment où deux autres nœuds éloignés communiquent aussi ensemble. De plus, le modèle doit faire apparaître des collisions si deux nœuds émettent en même temps et qu'un autre nœud reçoit les deux émissions en même temps à des niveaux de réception proches. Une probabilité d'échec de transmission sur le canal est configurée car dans des cas réels, le canal radio n'est pas absolument fiable. En effet, la bande de fréquence 2,4 GHz est ouverte à d'autres types de normes que le WiFi (Bluetooth, caméra sans fil, transmetteurs vidéos, alarme...). D'autres types de matériels peuvent donc entraîner des perturbations qui rendent impossible la bonne réception de paquets à un moment précis au niveau de certains nœuds. De plus des phénomènes de réflexion et de fading dans certains milieux entraînent aussi des altérations de paquets.

Notre modèle se limite à considérer un canal libre ou occupé par une ou plusieurs transmissions, en distinguant les zones de couverture, et à appliquer une probabilité d'altération de paquets aléatoirement, car les phénomènes hyperfréquences ne sont pas modélisés. Cependant, la modularité de ce modèle permet de faire évoluer cette couche radio si nécessaire, sans modifier le reste du modèle.

#### 3.3.2 Granularité au niveau de la couche liaison de données

La modélisation de cette couche est décomposée en deux sous-couches : *MAC* et *LLC*. Nous allons nous intéresser particulièrement à la sous-couche *MAC* qui modélise la norme WiFi et notamment la méthode d'accès au canal radio. Celle-ci est décomposée en plusieurs parties par raffinements successifs. La méthode d'accès présentée est basée sur DCF et HCF. La méthode PCF n'est pas décrite car elle est très peu utilisée de nos jours et elle n'est pas adaptée au WiFi maillé. En effet, elle nécessite un coordinateur que tous les nœuds doivent recevoir, ce qui n'est pas toujours possible dans le cas du WiFi maillé. De même, les méthodes d'association et de désassociation d'une station à un point d'accès ainsi que les connexions initiales entre points d'accès ne sont pas implémentées. Celles-ci ne représentent qu'un très faible nombre de paquets qui ne contribuent pas à améliorer ou à altérer la qualité de service. Nous considérons que les



nœuds sont connectés entre eux dès le début de la simulation.

La sous-couche *LLC* gère la répétition de paquets ainsi que l'identification des receveurs successifs permettant l'acheminement du paquet au travers du réseau maillé. Nous limitons notre étude à des introductions de paquets dans la couche liaison ainsi qu'au relais de paquets en appliquant des chemins statiques définis au travers du réseau.

### 3.3.3 Granularité au niveau supérieur à la couche liaison de données

Les niveaux supérieurs de la pile protocolaire se limitent au sein du modèle à de l'injection de flux dans le but de générer du trafic, et à analyser et dater les paquets reçus afin d'évaluer la qualité de service.

Les requêtes ICMP sont modélisées succinctement afin de fournir régulièrement des paquets à la couche liaison, et de mettre en place un aller-retour de paquets sur le réseau. Ceci nous permet de mesurer le RTT pour une communication. L'introduction de paquets suivant différentes qualités de service est aussi modélisée par une transmission UDP. Ceci permet de programmer des générations de paquets selon différents scénarii. Le protocole TCP, les protocoles Application et le fonctionnement des logiciels ne sont pas exprimés car les flux nécessitant de la qualité de service sont à base d'UDP. Cependant, ceux-ci pourront être modélisés par la suite grâce à la modularité du modèle.

## 3.4 Description

### 3.4.1 Configuration de la topologie d'une installation WiFi maillé

La topologie du réseau maillé WiFi doit être une entrée du modèle. En effet, le nombre de nœuds, ainsi que les caractéristiques des liens entre chaque nœud vont jouer un rôle majeur dans la qualité d'une communication sur ce réseau. Cependant, dans un cas réel, nous devons adapter la topologie au terrain car il n'est pas possible d'implanter des points d'accès n'importe où, ou de couper des arbres qui affaiblissent la qualité de transmission.

Trois places permettent de configurer la topologie étudiée dans le simulateur.

#### 3.4.1.1 Identifiant type et configuration de chaque nœud

La place *p\_conf\_host* présente dans le modèle *Network* (figure 3.14) doit contenir, pour chaque nœud, une association d'un entier identifiant le nœud et d'une structure de configuration de couleur *CONFHOST* :

- *isap* de couleur ISAP est égal à la constante *AP* pour point d'accès ou *STA* pour station,
- *rts* est un entier indiquant la taille du paquet à partir de laquelle les paquets de contrôle RTS/CTS seront activés. Une valeur supérieure à la taille maximale des paquets injectés dans la simulation désactivera ce mécanisme, et 0 permettra de l'activer tout le temps.
- *buf* est un entier spécifiant la taille du tampon pour les 4 files d'attente du nœud,
- *src* et *lrc* pour Short Retry Count et Long Retry Count sont des ensembles de couleurs *QINT* présentés ci-dessous. Ils contiennent les nombres de réémission

maximum en cas de non réception de l'accusé de réception. Si la réservation de médium RTS/CTS est activée, la valeur *lrc* est utilisée, sinon *src* est utilisée.

- *AIFS*,  $CW_{min}$  et  $CW_{max}$  sont des ensembles de couleurs *QINT* présentés ci-dessous. Ils précisent les délais intertrames et les limites des délais de période de contention en fonction de la classe de qualité de service.

Un *QINT* est une structure spécifiant les valeurs des constantes pour chaque type de classe de paquet. Elle contient 4 entiers nommés : *vo* (Vo-ip), *vi* (Video), *be* (Best Effort) et *bk* (Background).

Ainsi, il est possible de fixer ces paramètres différemment sur chaque nœud, comme c'est possible sur le terrain. Les tableaux 3.1 et 3.2 présentent un marquage initial de cette place *p\_conf\_host* qui contient la description des nœuds de l'installation.

Identifiant	<i>isap</i>	<i>rts</i>	<i>buf</i>	<i>src</i>				<i>lrc</i>			
				vo	vi	be	bk	vo	vi	be	bk
1	<i>AP</i>	2000	100	4	4	4	4	7	7	7	7
2	<i>AP</i>	2000	100	4	4	4	4	7	7	7	7
3	<i>AP</i>	2000	100	4	4	4	4	7	7	7	7
4	<i>AP</i>	2000	100	4	4	4	4	7	7	7	7
5	<i>AP</i>	2000	100	4	4	4	4	7	7	7	7
6	<i>AP</i>	2000	100	4	4	4	4	7	7	7	7
7	<i>STA</i>	2000	100	4	4	4	4	7	7	7	7
8	<i>STA</i>	2000	100	4	4	4	4	7	7	7	7

TABLE 3.1 – Exemple de configuration des nœuds

Identifiant	<i>AIFS</i>				$CW_{min}$				$CW_{max}$			
	vo	vi	be	bk	vo	vi	be	bk	vo	vi	be	bk
1	2	2	2	2	2	3	3	4	3	4	10	10
2	2	2	2	2	2	3	3	4	3	4	10	10
3	2	2	2	2	2	3	3	4	3	4	10	10
4	2	2	2	2	2	3	3	4	3	4	10	10
5	2	2	2	2	2	3	3	4	3	4	10	10
6	2	2	2	2	2	3	3	4	3	4	10	10
7	7	7	7	7	2	3	3	4	3	4	10	10
8	7	7	7	7	2	3	3	4	3	4	10	10

TABLE 3.2 – Exemple de configuration des nœuds (suite)

Ceci déclare la présence de huit nœuds dans le réseau. Les nœuds 1 à 6 sont des points d'accès, et les nœuds 7 et 8 sont des stations. Tous les nœuds ont la valeur *rts* à 2000, ce qui indique que le RTS/CTS n'est activé que pour les paquets faisant plus de 2000 octets. Ensuite, la valeur *AIFS* contient 2 pour toutes les classes de qualité de service du paquet pour les points d'accès et 7 pour toutes les classes de trafic des stations. Enfin,  $CW_{min}$  et  $CW_{max}$  sont identiques pour tous les nœuds et ont des valeurs différentes en fonction de la classe de qualité de service du paquet.

Cette configuration est accessible et regroupée ici, car ce sont des leviers réels qui pourront être modifiés dans le pilote WiFi des prototypes lors d'expérimentations.

### 3.4.1.2 Matrice de rapport signal sur bruit

La possibilité de transmission d'un paquet dépend du nœud émetteur et du nœud récepteur. En effet, le positionnement des nœuds et la topologie du terrain rendent certains liens possibles alors que d'autres sont mauvais voire impossibles. Le rapport signal sur bruit est mesuré par l'interface radio quand elle reçoit un paquet. Celui-ci dépend de la distance entre les nœuds, du dénivellement du terrain, de l'orientation des antennes, mais aussi de perturbations extérieures telles que le bruit ambiant ou des transmetteurs radios fonctionnant sur la même fréquence.

Ces rapports signal sur bruit entre chaque nœud sont renseignés dans un jeton de couleur *LISTLISTINT* situé dans la place *p\_powers* (présente dans le modèle *Network*, figure 3.14). Cette couleur représente un tableau de tableau d'entiers, ce qui correspond à une matrice (tableau 3.3). L'ordonnée renseigne l'émetteur et l'abscisse indique le receveur. La valeur 0 indique que le receveur n'est pas à portée de l'émetteur. La diagonale indiquant un même émetteur et receveur n'ayant pas de sens, sera mise à 0.

Le tableau 3.3 représente un exemple de jeton définissant les rapports signal sur bruit en prenant exemple sur la topologie de la figure 3.13. La première ligne signifie que le nœud 1 reçoit le nœud 2 à un niveau de 33 dB, et que le nœud 3 reçoit le nœud 1 à 1dB. Le nœud 4 ne reçoit pas le nœud 1.

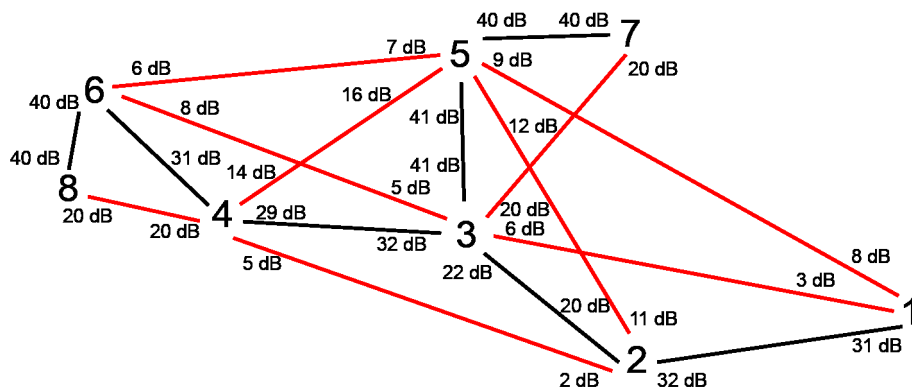


FIGURE 3.13 – Disposition des nœuds avec les rapports signal sur bruit

Cette matrice de rapports signal sur bruit permet de calculer les débits physiques utilisés entre chaque nœud, ainsi que la matrice d'atteignabilité au sens topologique.

En effet, la probabilité de succès de transmission entre deux nœuds dépend d'une part de la puissance de réception et d'autre part des perturbations de l'environnement. Cette matrice d'atteignabilité spécifiant les probabilités de succès de transmission entre chaque nœud synthétise ces paramètres car l'objet de cette étude ne porte pas sur la qualité des communications radio mais sur son impact en terme de transmission de paquets. Cette probabilité est appliquée à chaque transmission de paquet pour chaque récepteur afin de générer des pertes et des retransmissions.

		<i>Nœud récepteur</i>							
		1	2	3	4	5	6	7	8
<i>Nœud émetteur</i>	1	0	33	1	0	1	0	0	0
	2	35	0	16	6	18	0	0	0
	3	1	16	0	19	29	3	20	0
	4	0	6	18	0	9	34	20	20
	5	1	17	29	8	0	2	40	0
	6	0	0	3	35	4	0	0	40
	7	0	0	20	20	40	0	0	0
	8	0	0	0	20	10	40	0	0

TABLE 3.3 – Exemple de matrice de rapport signal sur bruit

Lorsqu'un nœud veut émettre, il vérifie qu'il ne capte aucun signal radio. Il vérifie donc qu'aucune station où la probabilité le concernant est strictement supérieure à 0 n'est en train d'émettre. De même si deux émetteurs envoient un paquet à un même récepteur, le paquet est marqué comme ayant subi une collision (le paquet ne sera pas lisible).

Le tableau 3.4 présente un exemple de marquage de  $p\_list\_atteign$  qui est calculé dans le modèle *Network* pour la topologie de la figure 3.13.

		<i>Nœud récepteur</i>							
		1	2	3	4	5	6	7	8
<i>Nœud émetteur</i>	1	0	95	58	0	58	0	0	0
	2	95	0	91	73	93	0	0	0
	3	58	91	0	94	95	64	95	0
	4	0	73	93	0	82	95	95	95
	5	58	92	95	79	0	61	95	0
	6	0	0	64	95	67	0	0	95
	7	0	0	95	95	95	0	0	0
	8	0	0	0	95	85	95	0	0

TABLE 3.4 – Exemple de matrice d'atteignabilité

La première ligne signifie que lorsque le nœud 1 émet, le nœud 2 a une probabilité de 95 % de recevoir correctement le paquet et que les nœuds 3 et 5 ont une probabilité de 58 % de recevoir correctement le paquet. Les nœuds 4, 6, 7 et 8 n'ont aucune chance de le recevoir. De plus, qu'il y ait collision ou non, et que le paquet soit transmis correctement ou non, les nœuds 2, 3 et 5 voient leur canal radio comme étant occupé.

### 3.4.1.3 Matrice de chemin

Dans ce modèle, les chemins qu'empruntent les paquets pour traverser le réseau sont statiques et définis initialement, à l'image des installations que nous déployons. Un jeton de couleur *LISTLISTINT* doit être placé dans la place  $p\_list\_chemin$  (modèle

*Network*, figure 3.14). Ce tableau à deux dimensions renseigne par quel nœud receveur il faut émettre ou relayer un paquet en fonction du destinataire.

Par exemple, dans la topologie de la figure 3.13, le marquage initial de la place *p\_list\_chemin* est la matrice présente sur la figure 3.5.

		<i>Nœud destinataire</i>							
		1	2	3	4	5	6	7	8
<i>Nœud source</i>	1	1	2	2	2	2	2	2	2
	2	1	2	3	3	3	3	3	3
	3	2	2	3	4	5	4	5	4
	4	3	3	3	4	3	6	3	6
	5	3	3	3	3	5	3	7	3
	6	4	4	4	4	4	6	4	8
	7	5	5	5	5	5	5	7	5
	8	6	6	6	6	6	6	6	8

TABLE 3.5 – Exemple de matrice de chemin

La deuxième ligne signifie que lorsque le nœud 2 doit émettre ou relayer un paquet au nœud 1, il lui envoie directement. Pour émettre ou relayer au nœud 3, il lui envoie aussi directement. Par contre, pour émettre ou relayer un paquet aux nœuds 4 à 8, il doit envoyer au nœud 3.

### 3.4.2 Modèle *Network*

Ce modèle présenté figure 3.14 est le modèle de plus haut niveau. Il permet d'assembler les autres modules *UDP*, *ICMP*, *LLC*, *MAC* et *Radio*. Ces sous modèles seront raffinés de manière à modéliser les mécanismes de la norme 802.11, et notamment les mécanismes liés à la qualité de service.

Nous pouvons voir sur la partie gauche la configuration de la topologie, sur la partie droite la configuration des transmissions, et sur la partie centrale les différents sous modèles impliqués.

La configuration de la topologie ne se fait que dans ce modèle. Elle est ensuite propagée à travers l'initialisation des sous modèles. De cette manière, ceux-ci sont initialisés à partir du modèle de plus haut niveau et il n'y a qu'un seul endroit où modifier cette configuration.

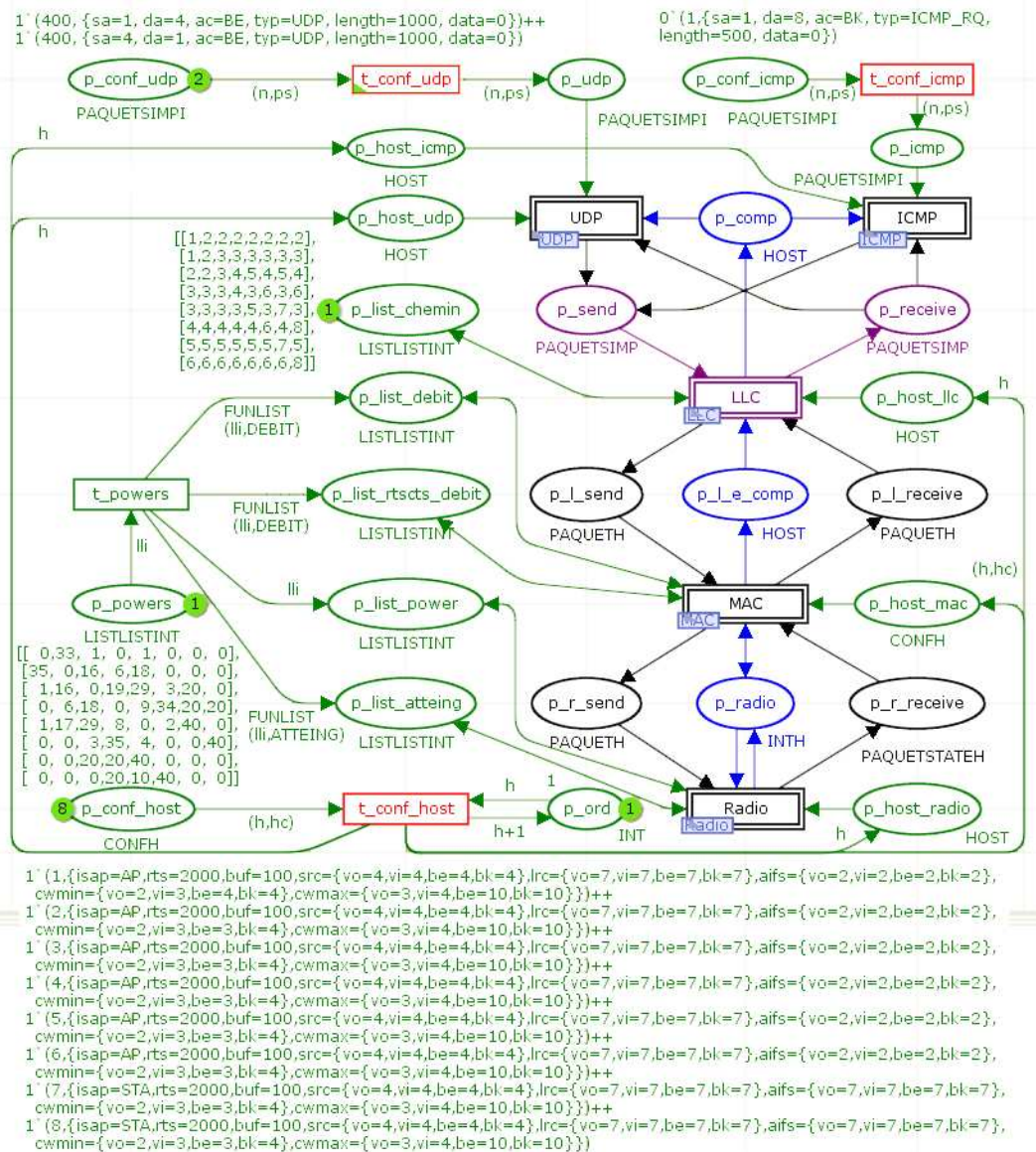


FIGURE 3.14 – Modèle Network

La configuration de la topologie se fait par l'intermédiaire des trois places : les chemins sont décrits dans *p\_list\_chemin*, les nœuds dans *p\_conf\_host* et les rapports signal sur bruit dans *t\_powers*.

Le marquage initial de *p\_list\_chemin* permet de configurer les différents chemins possibles entre chaque nœud. Le jeton issu du marquage initial sera à disposition du modèle *LLC* car c'est cette couche qui gère le relais de paquets et assigne par quel nœud va passer un paquet en fonction de sa destination.

Le marquage de *p\_conf\_host* indique le nombre et la configuration des nœuds. Pour chacun, le tir de *t\_conf\_host* duplique la configuration pour être transmise à

*UDP*, *ICMP*, *LLC*, *MAC*, et *Radio*. Dans ces sous modèles, il est nécessaire d'initialiser certaines places et de transmettre la configuration. Par exemple, les places complémentaires modélisant les files d'attente sont initialisées à ce moment là. De plus, le tir de  $t\_conf\_host$  incrémente un entier initialisé à 1 dans la place  $p\_ord$ . Ceci permet d'ordonner le tir des jetons correspondants aux nœuds pour faciliter le développement du modèle.

Enfin, le tir de  $t\_powers$  copie la configuration des rapports signal sur bruit, initialisés dans la place  $p\_powers$ , dans la place  $p\_list\_power$  pour qu'ils soient disponibles dans le sous modèle *Radio*. En effet, ce module transmet les paquets aux nœuds concernés par la transmission. De plus, cette matrice est transformée par la fonction *DEBIT* afin de créer un tableau de débit dans les places  $p\_list\_debit$  et  $p\_list\_rtscts\_debit$ . Celles-ci seront disponibles dans le sous modèle *MAC* qui définit le temps d'utilisation du canal radio en fonction de la taille du paquet, et implémente le mécanisme RTS/CTS. De même, la matrice de rapports signal sur bruit est transformée en matrice d'atteignabilité par la fonction *ATTEIGN* dans la place  $p\_list\_atteign$  qui sera disponible dans le sous modèle *Radio*.

Cette configuration est détaillée dans la partie Configuration de la topologie de la simulation.

La configuration du trafic à injecter sur le réseau se fait dans la place  $p\_conf\_icmp$  pour le modèle *ICMP* et dans la place  $p\_conf\_udp$  pour le modèle *UDP*. Le contenu de ces places sera détaillé dans les sous modèles correspondants.

Sur la partie centrale de ce modèle, nous pouvons voir les différentes macro transitions qui représentent les couches *Radio*, *MAC*, *LLC*, *ICMP* et *UDP*.

#### **Du module Radio au module MAC :**

La macro transition modélisant *Radio* est connectée à la macro transition *MAC* par l'intermédiaire de trois places :

- $p\_r\_send$  permet à *MAC* d'envoyer des paquets à *Radio*,
- $p\_r\_receive$  permet à *Radio* d'envoyer des paquets à *MAC*,
- $p\_radio$  permet à *MAC* de savoir, pour chaque récepteur, si le médium radio est occupé par un autre émetteur à chaque instant au niveau de chaque nœud.

En effet, la méthode d'accès CSMA/CA modélisée dans la couche *MAC* doit avoir connaissance de la présence d'un signal, pour savoir s'il peut démarrer une transmission.

#### **Du module MAC au module LLC :**

De même, trois places permettent de connecter le module *LLC* au module *MAC* :

- $p\_l\_send$  permet à *LLC* d'envoyer des paquets à la couche *MAC*,
- $p\_l\_receive$  permet à la couche *MAC* d'envoyer des paquets à la couche *LLC*,
- $p\_l\_e\_comp$  est une place complémentaire à la place  $p\_l\_send$  qui permet de limiter le nombre de paquets en attente d'émission dans la couche *MAC* car la taille des buffers d'émission n'est pas infinie.

#### **Du module LLC aux modules ICMP et UDP :**

Les deux places  $p\_send$  et  $p\_receive$  sont connectées à la macro transition *LLC* et permettent respectivement d'envoyer et de recevoir des paquets. La place  $p\_comp$  est une place complémentaire à la place  $p\_send$ .

Les modèles *ICMP* et *UDP* sont directement connectés à ces deux places  $p\_send$  et  $p\_receive$  de manière à émettre et recevoir des paquets. *ICMP* peut émettre des paquets *ICMP* de requête périodiquement comme le ferait le programme ping, et côté

récepteur, d'émettre des réponses dès la réception de la requête. De même, *UDP* peut générer des paquets ponctuellement, qui sont relayés d'un nœud à un autre, avec des caractéristiques de qualité de service, des tailles de paquets et des dates de départ des paquets qui peuvent être différents.

De nouveaux modules pourront être implémentés et connectés à ces places si d'autres manières de générer du trafic sont nécessaires selon les scénarii à étudier sur ces topologies.

### 3.4.3 Modèle *Radio*

Ce modèle décrit le comportement de la couche radio. En fonction de l'émetteur, chaque paquet peut donner lieu à un paquet pour chaque nœud pouvant recevoir la transmission. Cependant, la transmission peut être altérée pour certains récepteurs.

Pour le WiFi maillé, nous ne pouvons pas considérer un seul état du canal radio, car un nœud peut le voir occupé alors qu'un autre nœud distant peut le voir libre. De plus, pour détecter des collisions, si plusieurs paquets sont présents pour un même récepteur, nous allons comparer les niveaux de réception. Si les niveaux sont proches, la réception ne sera pas possible car il y aura eu collision.

Comme présenté dans le modèle parent *Network*, six places sont partagées entre ce modèle et son parent :

- La place d'entrée  $p\_init\_host$  va recevoir un jeton identifiant le nœud pour chaque nœud de manière à initialiser le modèle en fonction de la configuration.
- La place d'entrée  $p\_r\_send$  va recevoir les paquets devant être émis, et  $p\_r\_receive$  va transmettre au modèle parent les paquets qui vont être reçus.
- Les places d'entrée/sortie  $p\_list\_atteing$  et  $p\_list\_power$  contiennent respectivement un jeton configurant l'atteignabilité et un jeton configurant les niveaux de réception. Ceux-ci sont configurés dans le modèle *Network*.
- Enfin la place d'entrée/sortie  $p\_radio$  permet d'indiquer au modèle parent l'occupation radio pour chaque récepteur.

Nous pouvons décomposer le comportement de ce module en 6 étapes.

#### 3.4.3.1 1<sup>ère</sup> étape : initialisation

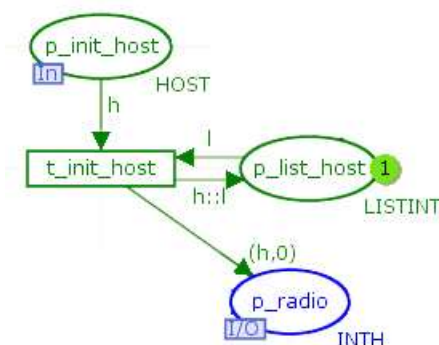


FIGURE 3.15 – Modèle Radio - 1<sup>ère</sup> étape - initialisation



Cette étape, présentée figure 3.15, permet la création d'une liste des identifiants de tous les nœuds dans la place  $p\_list\_host$ . Lors de l'initialisation, les jetons identifiant chaque nœud de la place  $p\_init\_host$ , fournis par le modèle *Network*, permettent des tirs de la transition  $t\_init\_host$ . La place  $p\_list\_host$  acceptant des jetons de couleur *LISTINT* est initialisée avec un jeton *nil*. Cette couleur correspond à une liste d'entiers et le jeton *nil* signifie une liste vide. A chaque tir de la transition  $t\_init\_host$ , l'identifiant du nœud est ajouté à la liste de la place  $p\_list\_host$ . De plus, un jeton  $(h, 0)$  est généré pour initialiser la place  $p\_radio$  qui accepte la couleur *INTH*. Cette couleur est une composition de l'identifiant du nœud avec un entier. Elle permet, pour chaque nœud de compter le nombre de fois où le récepteur radio reçoit un signal WiFi à un instant donné. Lors qu'il n'y a plus de jeton dans  $p\_init\_host$ , le jeton de la place  $p\_list\_host$  contient les identifiants de tous les nœuds, et  $p\_radio$  contient un jeton  $(h, 0)$  pour chaque nœud.

### 3.4.3.2 2<sup>ème</sup> étape : lorsqu'un paquet doit être émis, il est dupliqué pour chaque nœud pouvant le recevoir

Cette étape est présentée figure 3.16.

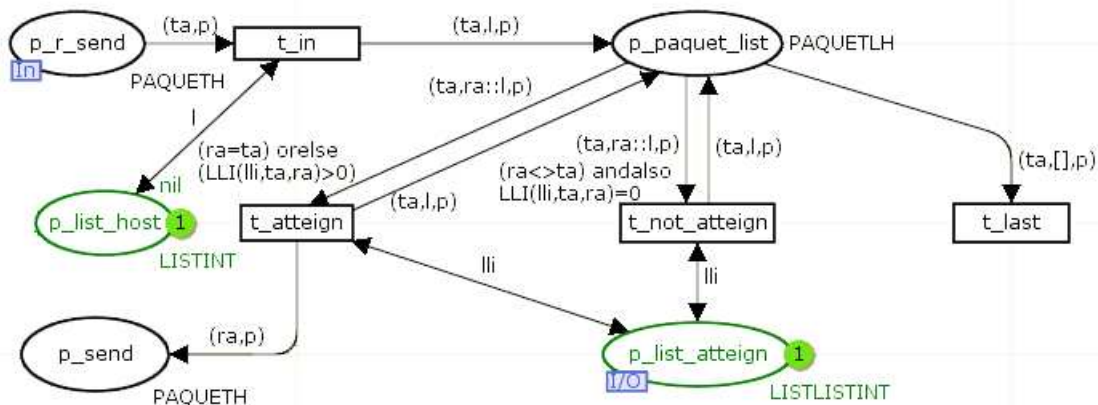


FIGURE 3.16 – Modèle Radio - 2<sup>ème</sup> étape - duplication

Lorsqu'un nœud émet un paquet, tous les nœuds qui sont à portée doivent recevoir ce paquet. Cette étape permet de dupliquer le paquet pour tous ces nœuds. La place d'entrée/sortie  $p\_list\_atteign$  est présente dans le modèle parent, et contient la matrice d'atteignabilité configurée initialement. La place d'entrée  $p\_r\_send$  accepte des jetons de couleur *PAQUETH*. Cette couleur est une composition de la couleur *HOST* qui identifie un nœud, et de la couleur *PAQUET*. Celle-ci est une structure qui contient tous les paramètres contenus dans le paquet :

- $sa$  de couleur *HOST* : l'identifiant du nœud source
- $ta$  de couleur *HOST* : l'identifiant du transmetteur
- $ra$  de couleur *HOST* : l'identifiant du receveur
- $da$  de couleur *HOST* : l'identifiant du destinataire
- $ac$  de couleur *AC* : spécifiant la classe de qualité de service du paquet

- *typ* de couleur *TYPE* : son type
- *airtime* entier précisant la durée nécessaire pour émettre ce paquet
- *length* : entier mémorisant la taille du paquet en octets
- *id* : entier identifiant du paquet, assigné dans une couche supérieure
- *data* : entier représentant les données fournies par la couche supérieure
- *ack* : booléen, faux si c'est un paquet ou vrai si c'est un accusé de réception.

Lorsqu'un paquet doit être émis, le modèle *Network* fournit un jeton de couleur *PAQUETH* dans la place *p\_r\_send*. La transition *t\_in* est tirée. Celle-ci consulte la liste des nœuds créés à l'étape précédente dans *p\_list\_host* afin de créer un jeton de couleur *PAQUETLH* dans la place *p\_paquet\_list*. Cette couleur est une composition des couleurs *HOST*, *LISTINT* et *PAQUET*. Ceci permet d'obtenir le paquet associé à une liste de tous les nœuds existants.

Cependant, nous ne voulons récupérer le paquet que pour les nœuds pouvant le recevoir. C'est pourquoi, pour chaque nœud de la liste, la transition *t\_atteign* ou la transition *t\_not\_atteign* va être tirée en fonction de la matrice d'atteignabilité contenue dans la place *p\_list\_atteign*. La transition *t\_atteign* ne va être tirée que pour les nœuds concernés par ce paquet, et un jeton de couleur *PAQUETH* est créé dans la place *p\_send* afin de procéder à la transmission. L'identifiant du nœud contenu dans ce jeton est maintenant l'identifiant de celui qui va recevoir le paquet. De plus, *t\_atteign* est aussi tirée pour le nœud émetteur. Ceci permettra de marquer le canal WiFi occupé aussi pour ce nœud. Lorsque la liste est vide, la transition *t\_last* est tirée afin de supprimer le jeton de la place *p\_paquet\_list*.

Pour un jeton entrant dans la place *p\_r\_send*, nous obtenons un jeton dans la place *p\_send* pour chaque nœud qui peut recevoir le paquet ainsi qu'un jeton pour l'émetteur.

### 3.4.3.3 3<sup>ème</sup> étape : marquer le canal radio de chaque récepteur comme étant occupé

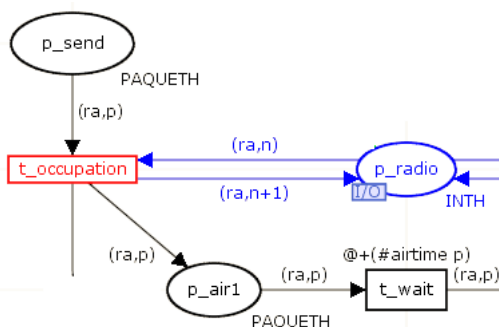


FIGURE 3.17 – Modèle Radio - 3<sup>ème</sup> étape - temporisation et occupation du médium

Comme présenté sur la figure 3.17, chaque jeton de la place *p\_send* correspond à une occupation de canal WiFi pour un nœud qui peut recevoir le paquet à émettre. Pour chacun de ces jetons, la transition *t\_occupation* va incrémenter l'entier qui correspond au nœud récepteur de la place *p\_radio*. Ceci permet d'informer le modèle parent *Network*

de l'occupation globale du canal radio.

De plus, lors du tir de cette transition, le paquet est déplacé dans la place  $p\_air1$  afin que la transition  $t\_wait$  puisse être tirée avec une temporisation de la durée contenue dans le paramètre  $airtime$  du paquet.

Enfin, le tir de  $t\_occupation$  est enregistré afin de pouvoir connaître précisément ce qu'il s'est passé et à quel moment lors de la simulation.

#### 3.4.3.4 4<sup>ème</sup> étape : gestion des interactions entre paquets

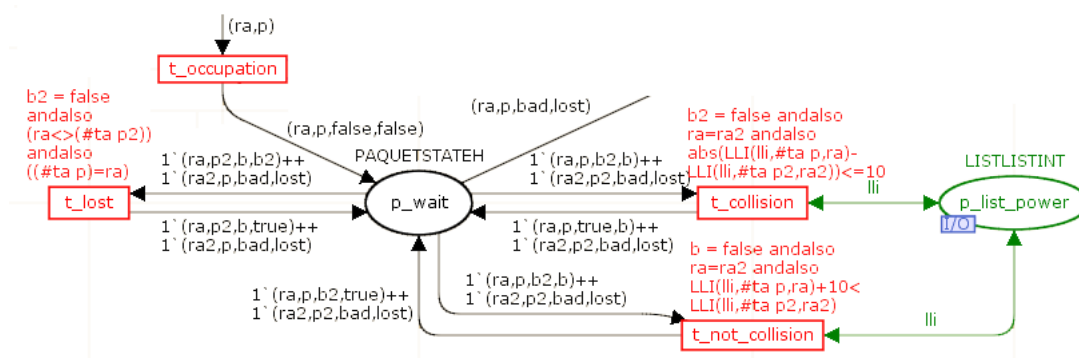


FIGURE 3.18 – Modèle Radio - 4<sup>ème</sup> étape - gestion des interactions entre paquets

La figure 3.18 présente les interactions entre paquets dans la couche *Radio*.

Afin de détecter et signaler des interactions entre des paquets, la place  $p\_wait$  mémorise tous les paquets en cours d'émission. Celle-ci est remplie lors du tir de  $t\_occupation$  par un jeton de couleur *PAQUETSTATEH*. Cette couleur est une composition des couleurs *HOST*, *PAQUET*, *BOOL* et *BOOL*. Elle permet de stocker dans un jeton le nœud auquel le paquet est destiné, le paquet en cours de transmission, un premier booléen qui indique si le paquet a subi une collision, et un second booléen qui indique que le paquet est émis à un nœud qui est en train d'émettre, ce qui ne lui permettra pas de recevoir le paquet.

Tous les paquets en cours d'émission sont dans la place  $p\_wait$  pour la durée de leur transmission. A chaque nouvelle arrivée dans cette place, les conditions de tir des transitions  $t\_lost$ ,  $t\_collision$  et  $t\_not\_collision$  sont testées.

##### Perte de paquet si le receveur est en cours d'émission :

La transition  $t\_lost$  sera tirée s'il existe deux jetons dans la place  $p\_wait$ , dont le récepteur du premier est le même que l'émetteur du second. Ce cas signifie qu'un nœud aurait un paquet à recevoir alors qu'il est en cours d'émission. Le paquet qui aurait dû être reçu est donc marqué comme *lost*.

##### Collision :

La transition  $t\_collision$  consulte la matrice de rapports signal sur bruit contenue dans la place  $p\_list\_power$  et elle est tirée s'il existe deux jetons dans la place  $p\_wait$ , dont les récepteurs sont identiques, et que la valeur absolue de la différence des rapports signal sur bruit de ces deux paquets est inférieure au seuil défini, soit 10 dB dans cet

exemple. En effet, il n'y a collision que si deux paquets reçus ont des rapports signal sur bruit proches.

#### Perte de paquet si le receveur est en cours de réception :

De même, la transition  $t\_not\_collision$  consulte la matrice de rapport signal sur bruit, et elle est tirée s'il existe deux jetons dans la place  $p\_wait$ , ayant le même receveur, et l'écart de niveau de réception est supérieur au seuil. Dans ce cas, le paramètre  $lost$  du paquet le plus faible est marqué, ce paquet sera perdu, et la transmission de l'autre paquet ne subira pas de collision.

Chaque tir de ces trois transitions  $t\_lost$ ,  $t\_collision$  et  $t\_not\_collision$  est enregistré afin d'établir des statistiques à la fin de la simulation.

#### 3.4.3.5 5<sup>ème</sup> étape : libération du canal radio

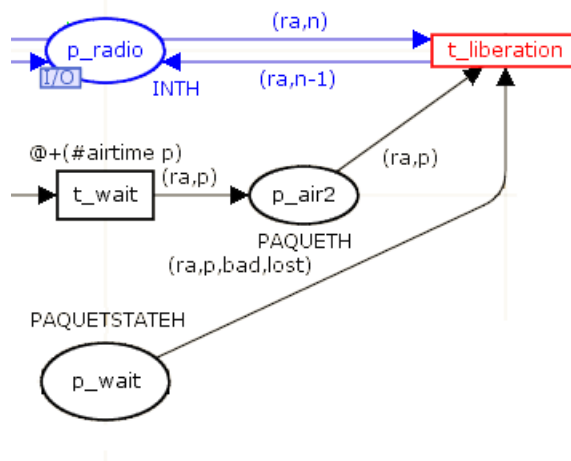


FIGURE 3.19 – Modèle Radio - 5<sup>ème</sup> étape - libération du canal radio

Lorsque la transmission est terminée, la transition temporisée  $t\_wait$ , tirée à la 3<sup>ème</sup> étape pour la durée de transmission, dépose le jeton dans la place  $p\_air2$ , ce qui est présenté figure 3.19. La transition  $t\_liberation$  est alors franchissable. Son tir décrémente l'entier comptant le nombre de fois où le canal radio est utilisé pour un récepteur donné dans la place  $p\_radio$ . Il enlève le jeton correspondant au paquet dans la place  $p\_air2$ , ainsi que dans  $p\_wait$ .

Ce tir est archivé afin de retracer les événements de la simulation.

### 3.4.3.6 6<sup>ème</sup> étape : appliquer la probabilité de perte et délivrer le paquet au modèle parent

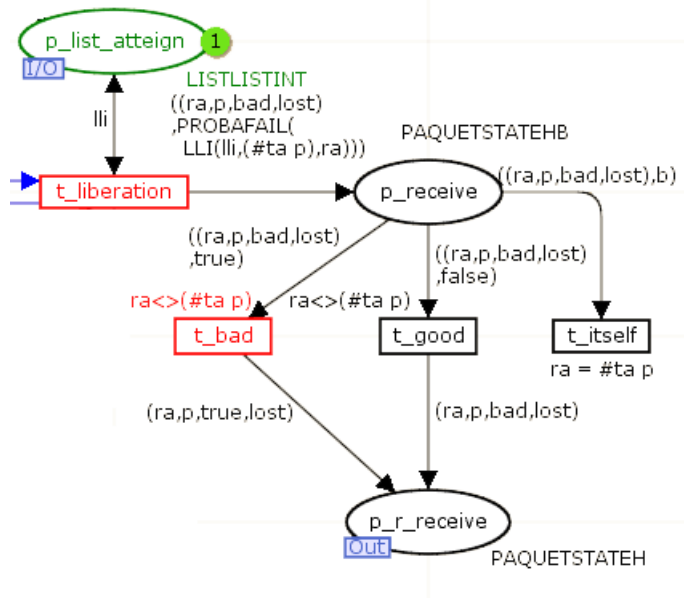


FIGURE 3.20 – Modèle Radio - 6<sup>ème</sup> étape - tir de probabilité de réussite

Comme présenté figure 3.20, le tir de la transition  $t\_liberation$  dépose un jeton de couleur  $PAQUETSTATEHB$  dans la place  $p\_receive$ . Cette couleur est la composition des couleurs  $PAQUETSTATEH$  et  $BOOL$ . La première partie du jeton  $\langle ra,p,bad,lost \rangle$  provient de la place  $p\_wait$ , où le jeton avait été marqué de pertes ou collisions potentielles. La seconde partie applique la probabilité de perte du jeton à cause du canal radio. La matrice d'atteignabilité contenue dans la place  $p\_list\_atteign$  est consultée, afin de déterminer la probabilité de succès de la transmission. Une valeur aléatoire suivant une fonction uniforme comprise entre 0 et 100 est tirée. Si cette valeur est inférieure à la probabilité, alors le paquet est marqué comme altéré.

Le jeton déposé dans la  $p\_receive$  contient donc le paquet avec des booléens signalant si le paquet a été émis correctement ou la cause de sa mauvaise transmission. À partir de cette place, trois possibilités se présentent :

- Si l'émetteur et le récepteur sont identiques, la transition  $t\_itself$  est tirée pour que ce nœud ne reçoive pas le paquet qu'il a émis.
- Si le paquet n'a pas été altéré, la transition  $t\_good$  est franchie.
- S'il a été altéré, la transition  $t\_bad$  est tirée et le paquet est marqué comme étant erroné, car la gestion est identique pour une collision ou une altération du canal radio.

Même si le paquet a été altéré, le jeton est déposé dans la place de sortie  $p\_r\_receive$ . Celle-ci étant liée au modèle *Radio*, les jetons déposés dans cette place sont transmis au modèle parent pour être acheminés à la couche MAC.

La figure 3.21 présente le modèle *Radio*, assemblé avec les modules décrits précédemment.

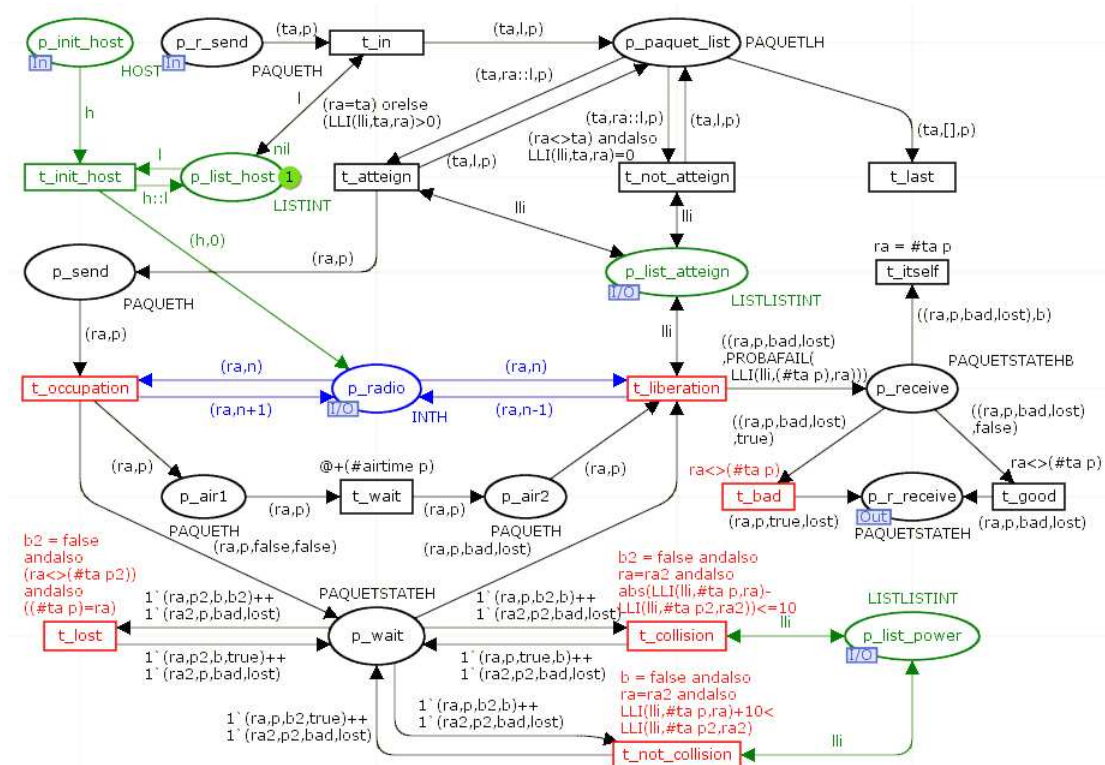
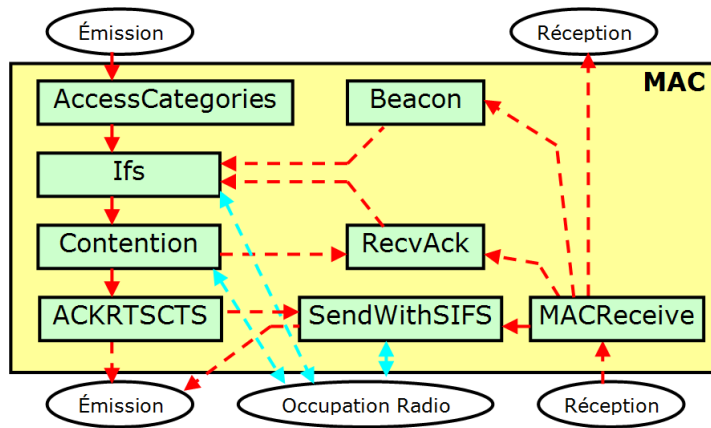


FIGURE 3.21 – Modèle *Radio* complet

### 3.4.4 Modèle *MAC*

Ce modèle définit le fonctionnement de la couche MAC d'un réseau WiFi. Il doit mettre en œuvre les fonctionnalités suivantes :

- émission et gestion d'accusés de réception et de paquets de contrôle RTS et CTS
- utilisation du canal radio uniquement s'il est vu comme libre
- gestion de différents délais intertrames et de périodes de contention
- émission périodique de beacon.

FIGURE 3.22 – Modèle *MAC* décomposé en sous parties

Les interactions entre les modules du modèle *MAC* sont présentées figure 3.22.

Lors de l'émission d'un paquet, celui-ci est envoyé dans le sous modèle *AccessCategories*. Celui-ci stocke les paquets dans 4 files de type FIFO correspondantes aux 4 classes de qualité de service définies dans la norme 802.11e. Lorsque aucun paquet n'est en émission par ce nœud et que le canal radio est libre, un paquet est dépilé afin d'être passé au sous modèle *Ifs*. Celui-là attend pendant un délai intertrame qui est dépendant de la classe de qualité de service du paquet, du succès de réception du dernier paquet reçu, et de la configuration du nœud.

Ensuite, si le canal radio est toujours libre, le paquet passe dans le sous modèle *Contention*. Il attend alors pendant une période de contention tirée aléatoirement entre des valeurs dépendantes du nombre de réémissions du paquet et de la configuration du nœud. Si pendant cette durée, le canal radio est encore libre, le paquet est transmis au modèle *ACKRTSCTS* qui va se charger de mettre en place le mécanisme RTS/CTS si nécessaire et de transmettre le paquet au modèle parent afin d'accéder à la couche *Radio*. De plus, il est dupliqué dans la place *RecvAck* et un compteur (timeout) est lancé afin d'attendre un accusé de réception.

Pour les nœuds de type *AP* (point d'accès), le sous modèle *Beacon* envoie des paquets périodiquement en diffusion. Ce type d'émission est défini par la norme 802.11 et sert à informer les stations voisines de la présence d'un point d'accès. Les paquets générés sont directement transmis au sous modèle *Ifs* afin de suivre le processus décrit précédemment.

Lorsqu'un paquet est reçu par l'interface radio, il est transmis au sous modèle *MACReceive*. Tout d'abord, il ignore les paquets qui ne sont pas destinés à ce nœud et ceux qui ont des erreurs de transmission. Ensuite, les paquets de type beacon sont transmis au sous modèle *Beacon*. Les paquets de type accusé de réception sont transmis au sous modèle *RecvAck* de manière à éviter la réémission de paquet. Si l'accusé de réception n'est pas reçu à temps, ce sous modèle transmet une nouvelle fois le paquet à la couche *Ifs* pour effectuer une nouvelle émission. Ce processus est fait un nombre de fois défini par la norme si l'accusé n'est toujours pas reçu. En cas d'échec, le paquet est perdu. Si le paquet est de type RTS ou CTS, il est transmis au modèle *ACKRTSCTS*.



Lorsque le modèle *MACReceive* reçoit un paquet de données, il est dupliqué au modèle parent afin de délivrer ce paquet reçu, ainsi qu'au sous modèle *ACKRTSCTS* qui génère un accusé de réception, et le transmet à *SendWithSIFS* pour l'émettre après un court délai intertrame si le canal radio est libre.

Nous pouvons décomposer le comportement de ce modèle en plusieurs étapes.

#### 3.4.4.1 1<sup>ère</sup> étape du modèle *MAC* : Initialisation du modèle

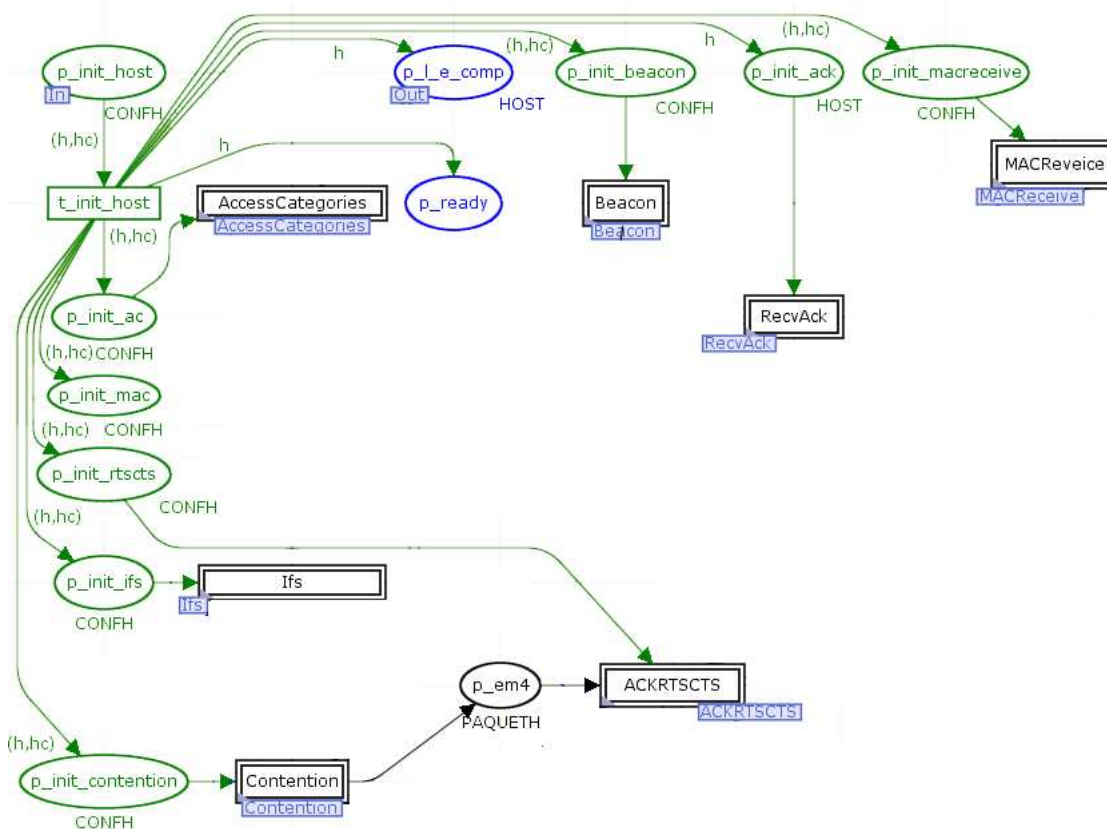


FIGURE 3.23 – Modèle *MAC* - 1<sup>ère</sup> étape - initialisation du modèle

La place *p\_init\_host*, partagée avec le modèle parent *Network* reçoit la configuration de chaque nœud. Celle-ci est diffusée dans les places à initialiser comme présenté sur la figure 3.23. Les places *p\_init\_beacon*, *p\_init\_ack*, *p\_init\_macreceive*, *p\_init\_ac*, *p\_init\_rtscs*, *p\_init\_ifs* et *p\_init\_contention* sont marquées par un jeton de couleur *CONFH* ou *HOST*, et permettent d'initialiser respectivement les sous modèles *Beacon*, *RecvAck*, *MACReceive*, *AccessCategories*, *ACKRTSCTS*, *Ifs* et *Contention*.

De plus, la place *p\_ready* est initialisée avec un jeton pour chaque nœud car elle indique que le nœud n'a pas de paquet en cours d'émission. De même, la place *p\_l\_e\_comp* est aussi marquée par un jeton pour chaque nœud car elle indique que la place d'entrée



des paquets  $p\_l\_send$  ne contient pas de paquet. Enfin, la place  $p\_init\_mac$  mémorise la configuration pour le modèle *MAC*.

### 3.4.4.2 2<sup>ème</sup> étape du modèle *MAC* : Classification et priorisation de paquet par le modèle *AccessCategories*

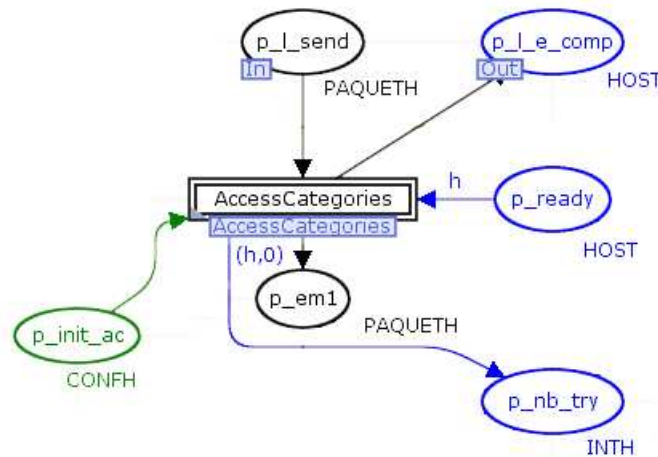


FIGURE 3.24 – Modèle MAC - 2<sup>ème</sup> étape - classification et priorisation

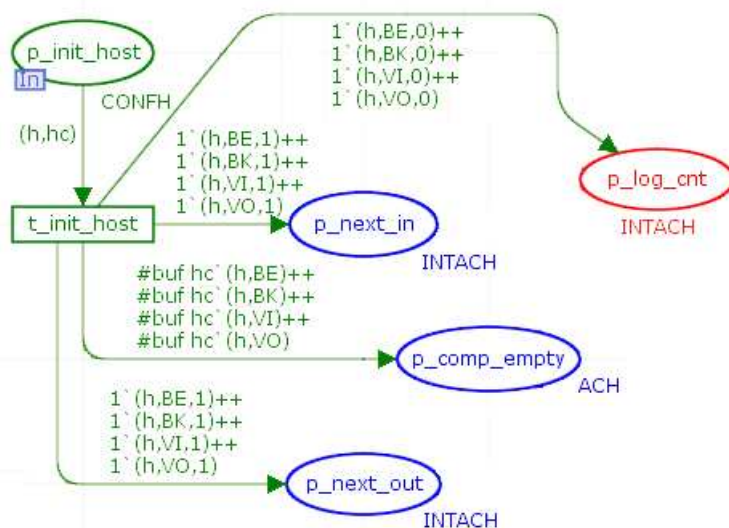
Le modèle *MAC* utilise le sous modèle *AccessCategories* pour classifier les paquets en 4 files d’attente. Ceci est défini par la norme 802.11e, et intégré aux pilotes WiFi. La configuration des nœuds est transmise à ce sous modèle par la place  $p\_init\_ac$ . Ceci est présenté figure 3.24.

Comme décrit précédemment, la place  $p\_l\_e\_comp$  est initialisée avec un jeton de couleur *HOST*25/07/2011 pour chaque nœud. Cette place de sortie vers le modèle parent *Network* est une place complémentaire à la place d’entrée  $p\_l\_send$ . Elle permet de limiter à 1 jeton par nœud le marquage de la place  $p\_l\_send$  et donc d’empêcher l’entrée de nouveaux paquets dans les files du sous modèle *AccessCategories*.

De même, la place  $p\_ready$  est initialisée par un jeton de couleur *HOST* pour chaque nœud. Ainsi, lorsqu’il n’y a pas de paquet en cours d’émission ou en attente d’accusé de réception pour un nœud donné, le jeton de couleur *HOST* correspondant au nœud est tiré par la macro transition *AccessCategories*, et un jeton de couleur *PAQUETH* contenant le paquet est déposé dans la place  $p\_em1$  afin d’être transféré aux étapes suivantes. En même temps, un jeton de couleur *INTH*  $\langle h,0 \rangle$  est déposé dans la place  $p\_nb\_try$ . Ce jeton va permettre de compter le nombre de réémissions de ce paquet.

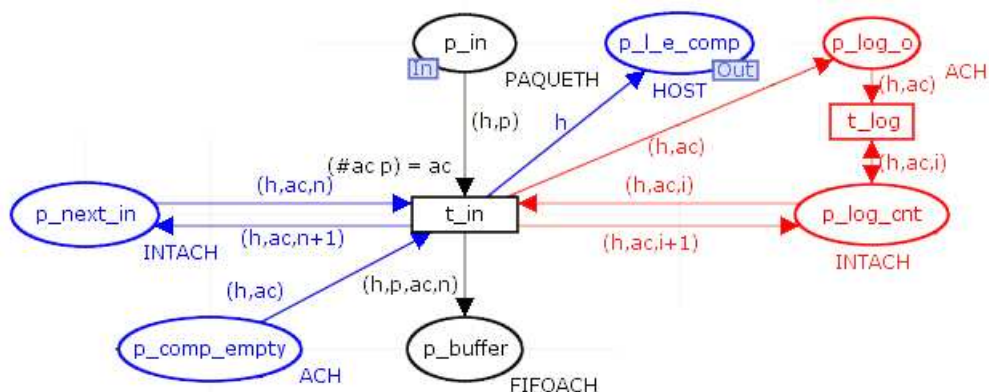
### 3.4.4.3 Description de la macro transition *AccessCategories*

La Macro transition *AccessCategories* représente 4 files d’attente de type fifo, de taille fixée par le paramètre *buf* contenu dans la structure *CONFH* configurant le nœud. De plus, lorsqu’un paquet peut être émis, ce modèle priorise les classes dans l’ordre *VO*, *VI*, *BE* puis *BK*.

FIGURE 3.25 – Modèle *AccessCategories* - 1<sup>ère</sup> étape - initialisation du modèle

L'initialisation de ce module est présentée figure 3.25. Les deux places  $p\_next\_in$  et  $p\_next\_out$  permettent de conserver l'ordre des paquets dans chaque file, et  $p\_log\_cnt$  permet de tracer le remplissage des tampons d'un nœud. Lors de l'initialisation, la place d'entrée  $p\_init\_host$  contient un jeton de couleur *CONFH* pour chaque nœud. Lors du tir de la transition  $t\_init\_host$ , les places  $p\_next\_in$ ,  $p\_next\_out$  et  $p\_log\_cnt$  reçoivent un jeton de couleur *INTACH* pour chaque nœud, et chaque classe de qualité de service. Cette couleur est une composition des couleurs *HOST*, *AC*, *INT*. (*AC* étant une énumération des 4 classes : *VO*, *VI*, *BE* et *BK*.)

De même, lors de l'initialisation, la place  $p\_comp\_empty$  reçoit autant de jetons que de paquets pouvant être stockés dans la FIFO. C'est cette place qui limite la taille des files d'attente.

FIGURE 3.26 – Modèle *AccessCategories* - 2<sup>ème</sup> étape - ajout d'un paquet

Lorsque le modèle *MAC* fait passer un paquet dans la place d'entrée  $p\_in$  comme



jeton de couleur *ACH* correspondant à une place dans la pile est déposé dans la place *p\_comp\_empty* pour permettre une nouvelle entrée dans la file d'attente.

Lorsqu'un paquet est sélectionné, il est déposé dans la place *p\_one*. La transition *t\_out* est franchissable et dépose un jeton contenant le paquet dans la place de sortie *p\_out* afin d'être mis en émission. De plus, l'entier contenu dans le jeton correspondant au nœud et à la qualité de service du paquet mis en émission est incrémenté pour mémoriser le numéro du prochain paquet à émettre. Le tir de *t\_out* entraîne aussi la décrémentation de l'entier correspondant à la file d'attente dans la place *p\_log\_cnt*, et le dépôt d'un jeton dans la place *p\_log\_o* afin de permettre un nouvel enregistrement de l'état du tampon dans le but de tracer l'évolution de la file d'attente.

Pour choisir le jeton à émettre dans la place *p\_buffer*, la transition *t\_sel* doit sélectionner le jeton le plus ancien dans la pile correspondant à la classe de qualité de service de plus forte priorité. L'ordre des priorités, défini par la norme 802.11e, est :

- *VO* (Vo-Ip),
- *VI* (Vidéo),
- *BE* (Best Effort)
- *BK* (Background).

Cette transition consulte donc les places *p\_next\_in* et *p\_next\_out* afin de déterminer si des paquets sont présents dans chaque file, ainsi que l'entier le plus ancien correspondant au paquet à sélectionner. Ces deux arcs de consultation identifient les variables suivantes aux entiers correspondants aux classes de qualité de service :

- *n01* : paquet de classe *VO* le plus récent
- *n11* : paquet de classe *VI* le plus récent
- *n21* : paquet de classe *BE* le plus récent
- *n02* : paquet de classe *VO* le plus ancien
- *n12* : paquet de classe *VI* le plus ancien
- *n22* : paquet de classe *BE* le plus ancien
- *n32* : paquet de classe *BK* le plus ancien

La variable *ac* correspond à la classe de qualité de service du paquet correspondant au jeton tiré dans *p\_buffer*, et *n* correspond à l'entier associé au paquet. La condition associée au tir de la transition *t\_sel* est valide si un de ces 4 cas se présente (car ces 4 cas sont séparés par un opérateur *OR ELSE*).

- **1<sup>er</sup> cas** : Si un paquet de classe de qualité de service *VO* est présent, il est tiré en priorité. Pour sélectionner le paquet le plus ancien de cette file, *n* devra être égal à *n02* qui correspond au prochain numéro de classe *VO* à émettre.
- **2<sup>ème</sup> cas** : Cette condition correspond au tir d'un paquet de classe *VI*. Le tir d'un paquet de cette classe ne peut se faire que si  $[(n01=n02)]$ , ce qui signifie que la file de classe *VO* est vide. La condition  $[(n=n12)]$  permet de sélectionner le paquet le plus ancien de la file *VI*.
- **3<sup>ème</sup> cas** : Pour qu'un paquet de classe *BE* soit tiré, il faut que  $[(n01=n02)]$  ce qui signifie que la file correspondant à la classe *VO* est vide, et que  $[(n11=n12)]$  ce qui signifie que la file de la classe *VI* est aussi vide. La condition  $[(n=n22)]$  permet de sélectionner le paquet le plus ancien dans la file *BE*.
- **4<sup>ème</sup> cas** : Cette condition correspond au tir d'un paquet de classe *BK*. Dans ce cas, les trois autres files doivent être vides. En effet, les conditions  $[(n01=n02)]$ ,  $[(n11=n12)]$  et  $[(n21=n22)]$  vérifient que les files de classes *VO*, *VI* et *BE* sont

vides. Enfin, la condition  $[(n=n32)]$  permet de sélectionner le paquet le plus ancien de la file *BK*.

De cette manière les paquets sont priorisés suivant leurs classes de qualité de service, et sur une même classe, l'ordre des paquets est préservé.

La figure 3.28 présente le module *AccessCategories* complet.

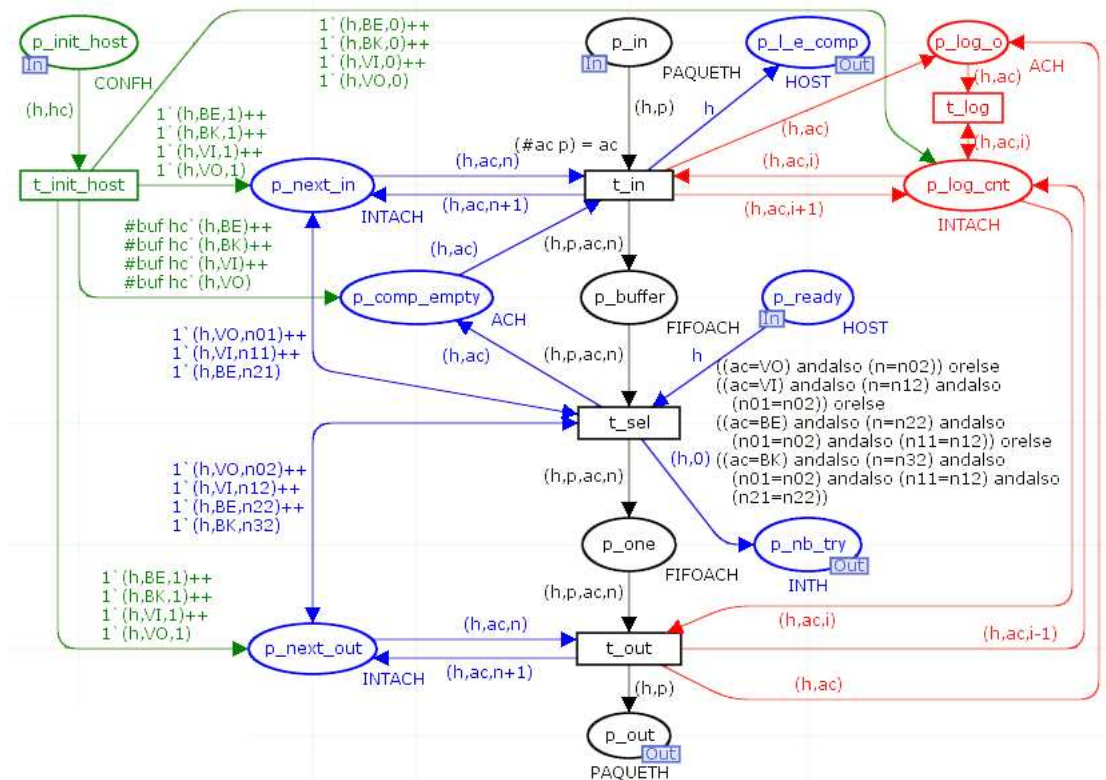


FIGURE 3.28 – Modèle *AccessCategories* complet





des nœuds ainsi que leur type.

Enfin, cette macro transition peut consulter la place  $p\_radio$  qui marque l'état du canal radio pour chaque nœud, la place  $p\_last\_pk$  qui mémorise si la dernière transmission reçue est un paquet sans erreur et sans collision, ainsi que la place  $p\_rtscts\_lock$  qui indique si le nœud vient de recevoir un RTS ou CTS indiquant de ne pas émettre.

### 3.4.4.5 Description de la macro transition $I_{fs}$

La macro transition  $I_{fs}$  permet au paquet d'attendre que le canal radio soit libre et que le nœud n'ait pas reçu de paquet de contrôle RTS ou CTS l'empêchant d'émettre, puis d'attendre la durée de la période intertrame comme le spécifie CSMA/CA. Son initialisation est présentée figure 3.30.

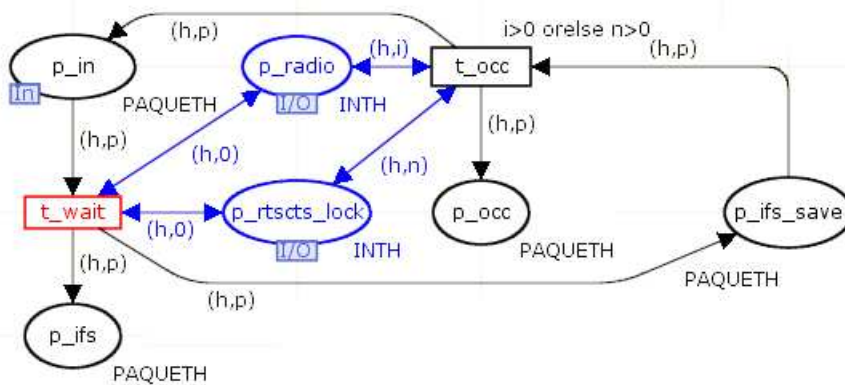


FIGURE 3.30 – Modèle  $I_{fs}$  - 1<sup>ère</sup> étape - entrée du paquet et vérification du canal radio

Lorsqu'un paquet arrive dans ce sous modèle par la place d'entrée  $p\_in$ , il attend que la transition  $t\_wait$  soit franchissable. En effet, cette transition consulte les places d'entrée/sortie  $p\_radio$  et  $p\_rtscts\_lock$  qui contiennent respectivement l'état du canal radio de chaque nœud et l'état du mécanisme de blocage RTS/CTS. Il faudra que ces deux places autorisent une émission pour le nœud courant.

Lors du tir de  $t\_wait$ , le paquet est copié dans la place  $p\_ifs$  afin d'attendre le délai intertrame et dans  $p\_ifs\_save$  pour vérifier l'état du canal radio. La transition  $t\_occ$  est franchissable si la place  $p\_radio$  indique que le nœud reçoit au moins un paquet, ou que la place  $p\_rtscts\_lock$  indique que le nœud a reçu un RTS ou un CTS. Son tir déplace le paquet de  $p\_ifs\_save$  dans la place  $p\_occ$  et dans la place  $p\_in$  afin de réinitialiser le délai intertrame. La présence d'un paquet dans  $p\_occ$  entraîne sa suppression après le délai d'attente.





dans  $p\_ifs\_save$  et la transition  $t\_out$  déplace le paquet dans la place de sortie  $p\_out$  afin de passer en émission, soit le paquet est supprimé par la transition  $t\_del$  car la transition  $t\_occ$  a déjà copié le paquet en entrée du modèle pour attendre un nouveau délai intertrame. En effet, si une transmission est détectée pendant l'attente du délai intertrame, celui-ci est réinitialisé, et redémarre à la fin de la transmission.

La figure 3.33 présente le module *I*fs complet.

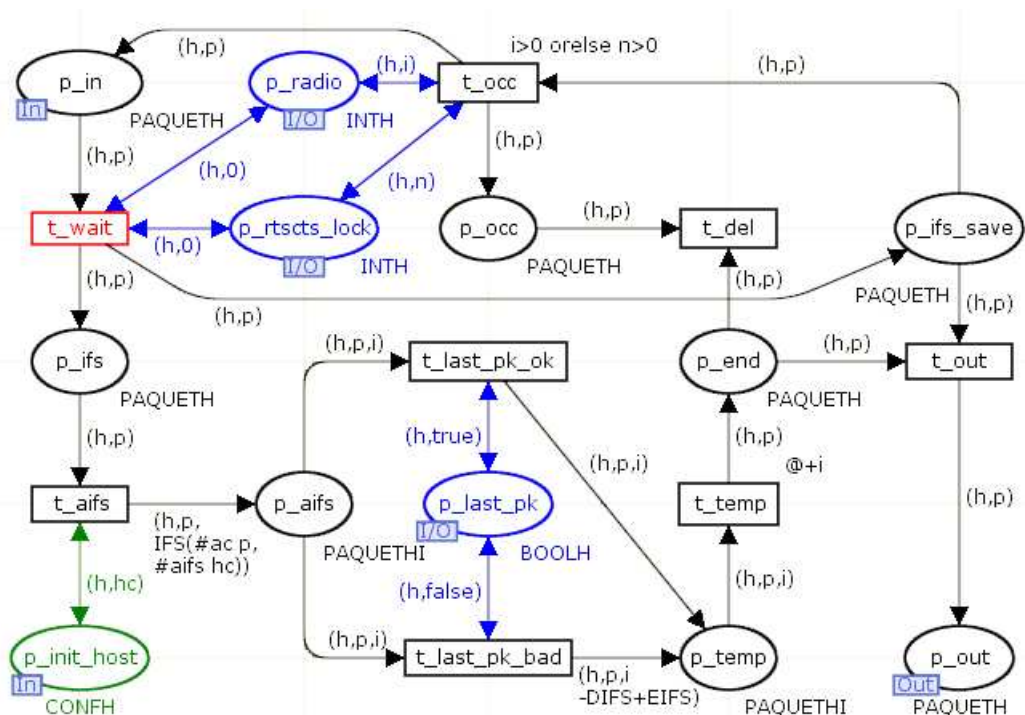


FIGURE 3.33 – Modèle *I*fs complet

### 3.4.4.6 4<sup>ème</sup> étape du modèle *MAC* : Décompte de la période de contention par le modèle *Contention*

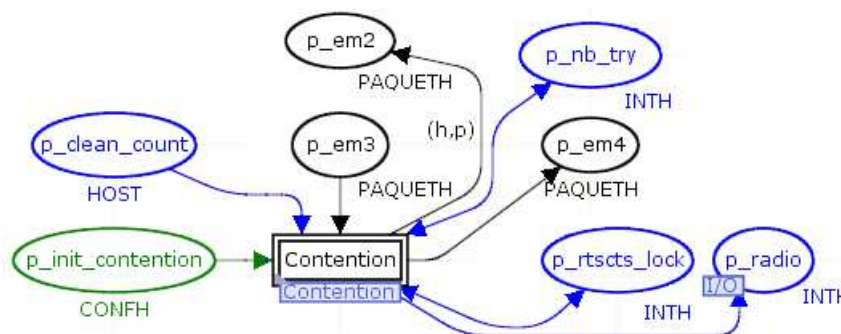


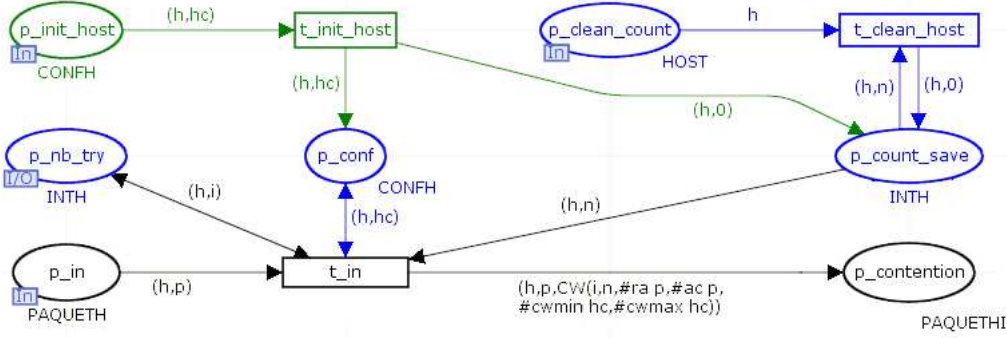
FIGURE 3.34 – Modèle *MAC* - 4<sup>ème</sup> étape - Période de contention

La figure 3.34 présente l'insertion de la macro transition *Contention* dans le modèle *MAC*.

Après avoir attendu le délai intertrame, la norme 802.11 prévoit une période de contention dans le cas où le canal radio était utilisé au moment de la décision d'émettre le paquet, ou à la suite d'échec de transmission. Pour ceci, la macro transition *Contention* consulte les places *p\_radio*, afin de vérifier que le canal est toujours libre et *p\_rtscs\_lock* pour savoir si le nœud peut émettre. Le jeton contenant le paquet qui provient de la place *p\_em3* est déposé soit dans la place *p\_em4* si le canal n'a pas été utilisé, soit dans la place *p\_em2* si le canal a été utilisé, de manière à attendre à nouveau un délai intertrame. De plus, la place *p\_clean\_count* permet à ce sous modèle de réinitialiser le compteur si c'est une nouvelle tentative d'émission. Enfin, le tir aléatoire de la période de contention dépend de la configuration du nœud, ainsi que du nombre de réémissions pour le paquet en cours. C'est pourquoi la place *p\_init\_contention* initialise la macro transition, et la place *p\_nb\_try* est consultée pour indiquer le nombre de réémissions.

### 3.4.4.7 Description de la macro transition *Contention*

Cette macro transition *Contention* a pour but d'attendre un délai aléatoire. La figure 3.35 présente son initialisation.

FIGURE 3.35 – Modèle *Contention* - 1<sup>ère</sup> étape - Entrée d'un paquet

Lors de l'initialisation du modèle, la place l'entrée  $p\_init\_host$  contient un jeton contenant la configuration des nœuds. Pour chaque nœud, la transition  $t\_init\_host$  crée un jeton  $\langle h, 0 \rangle$  dans la place  $p\_count\_save$  qui stocke l'état du compteur de temps d'attente. De plus, la configuration est placée dans  $p\_conf$ .

Dans le cas de la première tentative d'émission d'un paquet, la place d'entrée  $p\_clean\_count$  permet de réinitialiser le compteur à 0.

La place d'entrée  $p\_in$  correspond à la place  $p\_em3$  du modèle *MAC*. Lorsqu'un paquet a attendu le délai intertrame, il est présent dans la place  $p\_in$ . La transition  $t\_in$  consulte les places  $p\_nb\_try$  et  $p\_conf$  pour connaître respectivement le numéro de la réémission de ce paquet et la configuration du nœud. Ce tir prend aussi le jeton contenant l'état précédent du compteur dans la place  $p\_count\_save$ . Si ce compteur est strictement supérieur à 0, c'est qu'une période de contention a déjà commencé à être décomptée pour ce paquet. Dans ce cas, le décompte se poursuit.

Sinon, ce tir calcule la durée de la période de contention et la dépose avec le paquet dans la place  $p\_contention$ .

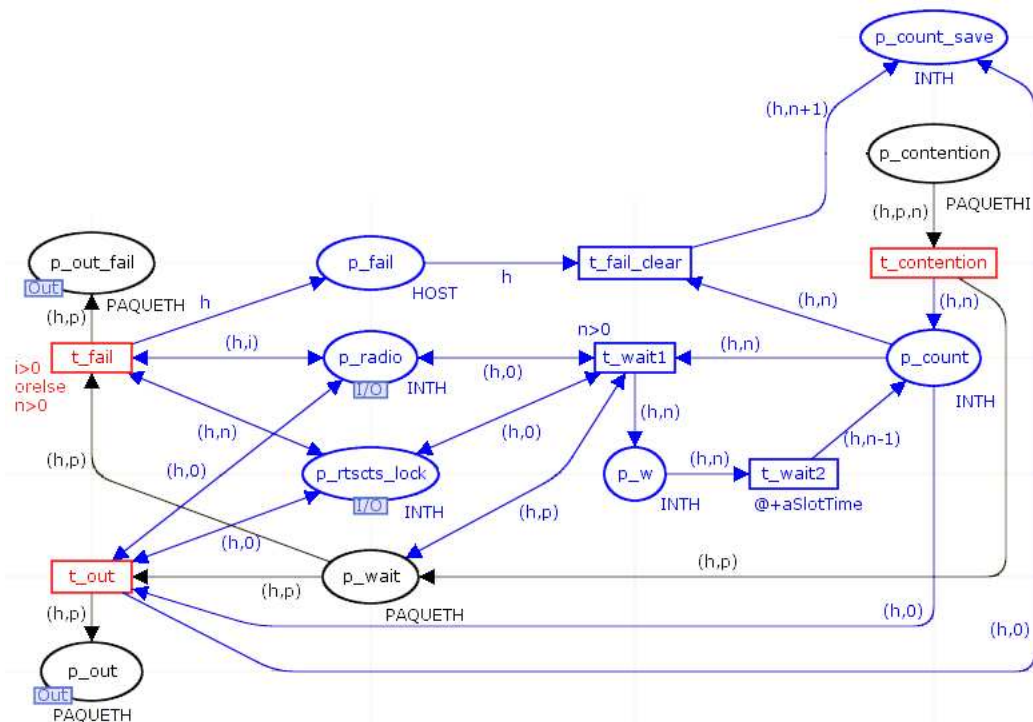
$$CW_n = 2 \cdot (CW_{n-1} + 1) - 1 \quad (3.1)$$

$$CW_0 = CW_{min} \quad (3.2)$$

$$CW_n > CW_{max} \rightarrow CW_n = CW_{max} \quad (3.3)$$

La durée de la période de contention est un nombre entier de durée  $aSlotTime$  tiré aléatoirement dans l'intervalle  $[0; CW_n]$  de manière uniformément répartie.  $CW_n$  est calculé à partir des formules 3.1, 3.2 et 3.3 avec  $n$  étant le numéro de la réémission. Ainsi, la plage de tir de la période de contention croît en fonction du nombre de réémission.

Les constantes  $CW_{min}$  et  $CW_{max}$  ont des valeurs par défaut définies dans la norme 802.11e. Cependant, la technologie que nous utilisons permet de modifier ces valeurs. C'est donc un levier sur lequel nous pouvons jouer. Nous allons donc pouvoir modifier ces valeurs de manière différente sur chaque nœud.

FIGURE 3.36 – Modèle *Contention* - 2<sup>ème</sup> étape - Temporisation

La figure 3.36 présente la temporisation de la période de contention. Si la place  $p_{radio}$  contient un jeton correspondant au nœud émetteur indiquant que le canal radio est occupé ou si la place  $p_{rtscs\_lock}$  contient un jeton indiquant que le nœud ne doit pas émettre, la transition  $t_{fail}$  est immédiatement franchissable. Le tir de cette transition déplace le paquet en attente dans la place de sortie  $p_{out\_fail}$  et crée un jeton de couleur  $h$  dans la place  $p_{fail}$ . Cette place permet de notifier que le compteur doit être réinitialisé. En effet, lorsque la place  $t_{wait2}$  aura fini sa temporisation, les transitions  $t_{out}$  et  $t_{wait1}$  ne sont plus franchissables, et la transition  $t_{fail\_clear}$  est tirée, ce qui supprime le jeton du compteur afin de réinitialiser ce modèle. Le paquet de la place de sortie  $p_{out\_fail}$  est disponible dans la place  $p_{em2}$  dans le modèle *MAC*, ce qui relance le cycle d'émission à partir du délai intertrame.

Ensuite, la transition  $t_{contention}$  initialise le compteur dans la place  $p_{count}$ , et met le paquet en attente dans la place  $p_{wait}$ . Si le canal radio devient occupé, le paquet peut directement retourner avant l'étape de délai intertrame.

La place  $p_{count}$  est marquée par le nombre de temps élémentaires de périodes de contention représenté par la constante  $aSlotTime$ . Pour chaque unité, les transitions  $t_{wait1}$  et la temporisation de  $aSlotTime$  sont franchis si :

- la place d'entrée/sortie  $p_{radio}$  marque toujours le canal radio comme libre pour ce récepteur
- la place d'entrée/sortie  $p_{rtscs\_lock}$  marque que le canal radio n'est pas réservé par un RTS ou CTS,
- l'entier décrémente est supérieur à 0,

– le paquet correspondant au nœud est toujours dans la place  $p\_wait$ .

Dans le cas là, la temporisation prévue est appliquée. Lorsque celle-ci est terminée, l'entier est égal à 0, et la transition  $t\_out$  permet de transmettre le paquet avec succès au modèle parent  $MAC$ . Le compteur de la place  $p\_count$  est aussi retiré afin de remettre ce modèle dans son état initial.

Si la place  $p\_radio$  contient un jeton correspondant au nœud émetteur indiquant que le canal radio est occupé ou si la place  $p\_rtscts\_lock$  contient un jeton indiquant que le nœud ne doit pas émettre, la transition  $t\_fail$  est immédiatement franchissable. Ainsi, cette transition déplace le paquet en attente dans la place de sortie  $p\_out\_fail$  et marque la place  $p\_fail$  pour notifier que le compteur doit être réinitialisé. En effet, à la fin de la temporisation, la transition  $t\_fail\_clear$  supprimera le jeton du compteur afin de réinitialiser ce modèle. Le paquet de la place de sortie  $p\_out\_fail$  sera disponible dans la place  $p\_em2$  dans le modèle  $MAC$ , ce qui va relancer le cycle d'émission à partir du délai intertrame.

La figure 3.37 présente le module *Contention* complet.

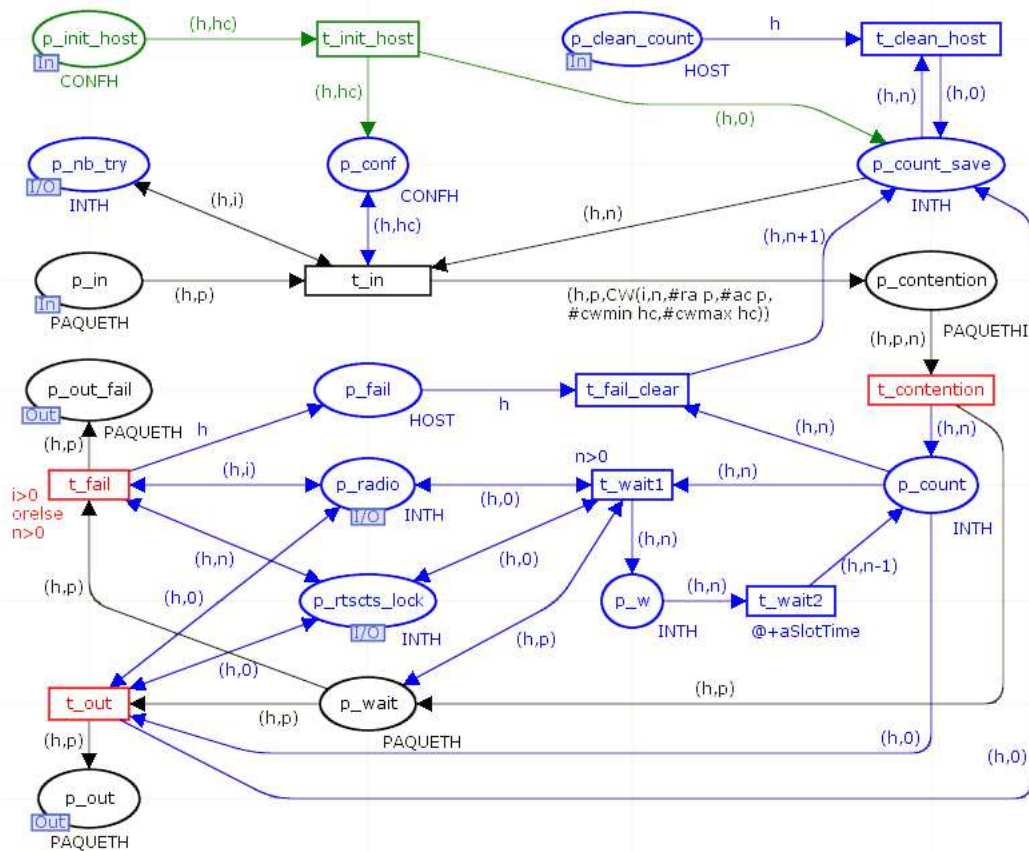


FIGURE 3.37 – Modèle *Contention* complet

### 3.4.4.8 5<sup>ème</sup> étape du modèle MAC : Emission de paquet

La figure 3.38 présente l'émission de paquet dans le module MAC.

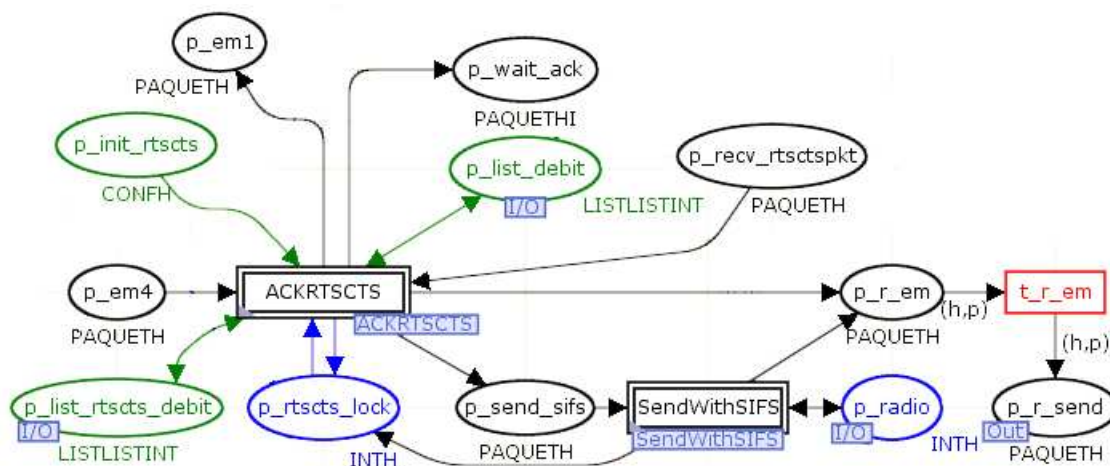


FIGURE 3.38 – Modèle MAC - 5<sup>ème</sup> étape - Emission de paquet

Lorsqu'un jeton contenant un paquet se trouve dans la place  $p_{em4}$ , c'est que le canal est libre et que les délais intertrames se sont écoulés. Le paquet est alors transmis à la macro transition  $ACKRTSCTS$  qui gère l'émission des paquets de contrôle RTS et CTS si nécessaire. Voici les liens entre ce module et le modèle MAC :

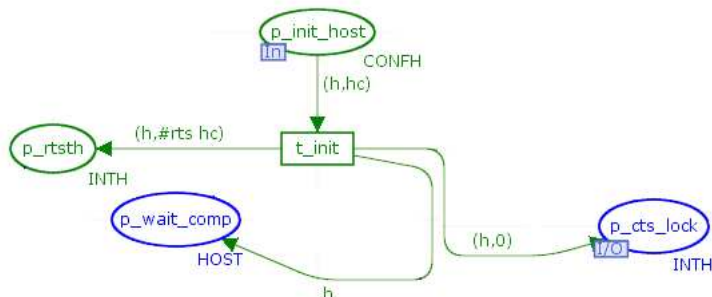
- les places  $p_{list\_rtscs\_debit}$  et  $p_{list\_debit}$  permettent d'assigner dans le jeton paquet la durée d'émission en fonction de la taille du paquet, et de configurer les données contenues dans les paquets RTS et CTS.
- le contenu de la place  $p_{rtscs\_lock}$  est mis à jour afin d'empêcher un paquet d'être émis si le canal radio est réservé.
- la place  $p_{wait\_ack}$  permet d'indiquer pour quel paquet un accusé de réception est attendu. Ceci sera développé dans les parties suivantes.
- le paquet est déposé dans la place  $p_{r\_em}$  s'il doit être émis, sinon le paquet est dans la place  $p_{em1}$  pour être réinjecté dans le processus d'émission.
- la place  $p_{recv\_rtscs}$  permet de transmettre les paquets reçus de type RTS et CTS
- la place  $p_{send\_sifs}$  permet d'émettre des paquets par la macro transition  $SendWithSIFS$  qui attend un délai intertrame court.

Les macro transitions  $ACKRTSCTS$  et  $SendWithSIFS$  peuvent déposer le paquet à émettre dans la place  $p_{r\_em}$  pour être transmis au modèle parent par la place de sortie  $p_{r\_send}$  afin qu'il soit passé à la couche *Radio*.

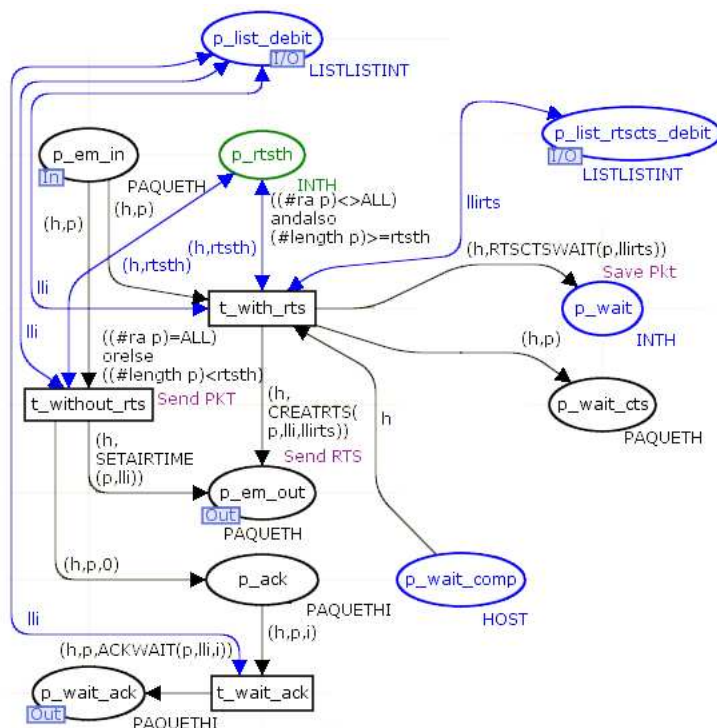
### 3.4.4.9 Description de la macro transition $ACKRTSCTS$

La première étape de la macro transition  $ACKRTSCTS$  est détaillée figure 3.39.



FIGURE 3.39 – Modèle *ACKRTSCTS* - 1<sup>ère</sup> étape - Initialisation

Lors de l'initialisation, la place  $p\_init\_host$  fournit la configuration de chaque nœud. Pour chacun d'eux, la transition  $t\_init$  initialise la place  $p\_wait\_comp$  qui permet de mémoriser si un RTS vient d'être émis et si un CTS est en attente. De plus, elle initialise la place d'entrée/sortie  $p\_cts\_lock$ , qui permettra d'indiquer aux autres parties du modèle que le nœud est bloqué suite à la réception d'un paquet RTS ou CTS. Enfin, la place  $p\_rtsth$  reçoit un jeton de couleur *INTH* contenant l'identifiant du nœud et le seuil à partir duquel un paquet doit suivre le mécanisme RTS/CTS. Ce paramètre est un levier car il active ou non l'émission de RTS avant le paquet afin de réserver le canal radio.

FIGURE 3.40 – Modèle *ACKRTSCTS* - 2<sup>ème</sup> étape - Emission de paquet ou de RTS

A l'arrivée d'un paquet dans la place d'entrée  $p\_em\_in$ , comme présenté figure 3.40, soit le paquet est émis directement, soit un dialogue RTS/CTS précède le paquet.

- Si la taille du paquet est inférieure au seuil  $rtsth$  qui est défini dans la configuration du nœud émetteur, le mécanisme RTS/CTS n'est pas utilisé, et le paquet doit être émis directement. Dans ce cas, la transition  $t\_without\_rts$  déplace le paquet en émission dans la place de sortie  $p\_em\_out$  pour être transmis à la couche *Radio*. Le paquet est copié dans la place  $p\_ack$  afin qu'un accusé de réception soit attendu par ce paquet.
- Si la taille du paquet est supérieure ou égale au seuil, la transition  $t\_with\_rts$  crée un paquet RTS contenant la durée de réservation du canal. Celle-ci est la somme de la durée d'émission du paquet, de la réponse CTS, de l'accusé de réception et de  $3 SIFS$ . Ce RTS est déposé dans la place de sortie  $p\_em\_out$  afin qu'il soit transmis à la couche *Radio*. De plus, le paquet est copié dans les place  $p\_wait$  et  $p\_wait\_cts$  afin de déclencher un timer en attente du paquet CTS. Le jeton contenu dans la place complémentaire  $p\_wait\_comp$  est retiré pour mémoriser l'état d'attente de CTS.

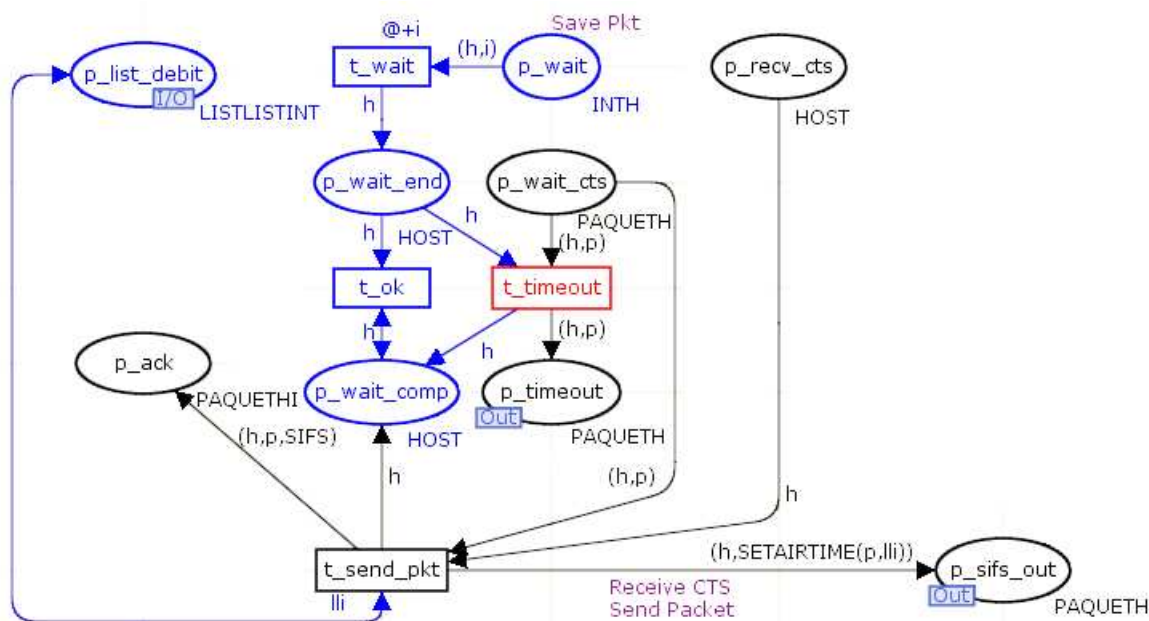


FIGURE 3.41 – Modèle *ACKRTSCTS* - 3<sup>ème</sup> étape - Attente de CTS

Comme présenté figure 3.41, suite à l'émission d'un RTS, la transition temporisée  $t\_wait$  attend la durée d'émission du RTS ainsi qu'un délai intertrame *SIFS*. Ensuite, si le CTS est reçu, il est présent dans la place  $p\_recv\_cts$ , et la transition  $t\_send\_pkt$  permet l'émission du paquet suite la réservation réussie du canal radio. Cependant, un délai intertrame *SIFS* doit être attendu entre le CTS et le paquet. C'est pourquoi le paquet est déposé dans la place de sortie  $p\_sifs\_out$ . Le modèle *MAC* transmet le paquet dans la macro transition *SendWithSIFS* avant de le transmettre à la couche *Radio*.



Si le CTS n'est pas reçu, ce qui arrive si le RTS ou le CTS n'a pas été transmis correctement, la transition  $t\_timeout$  est franchie et le paquet est déplacé dans la place de sortie  $p\_timeout$  qui va permettre de redémarrer le processus d'émission pour ce paquet, avec le délai intertrame et la période de contention.

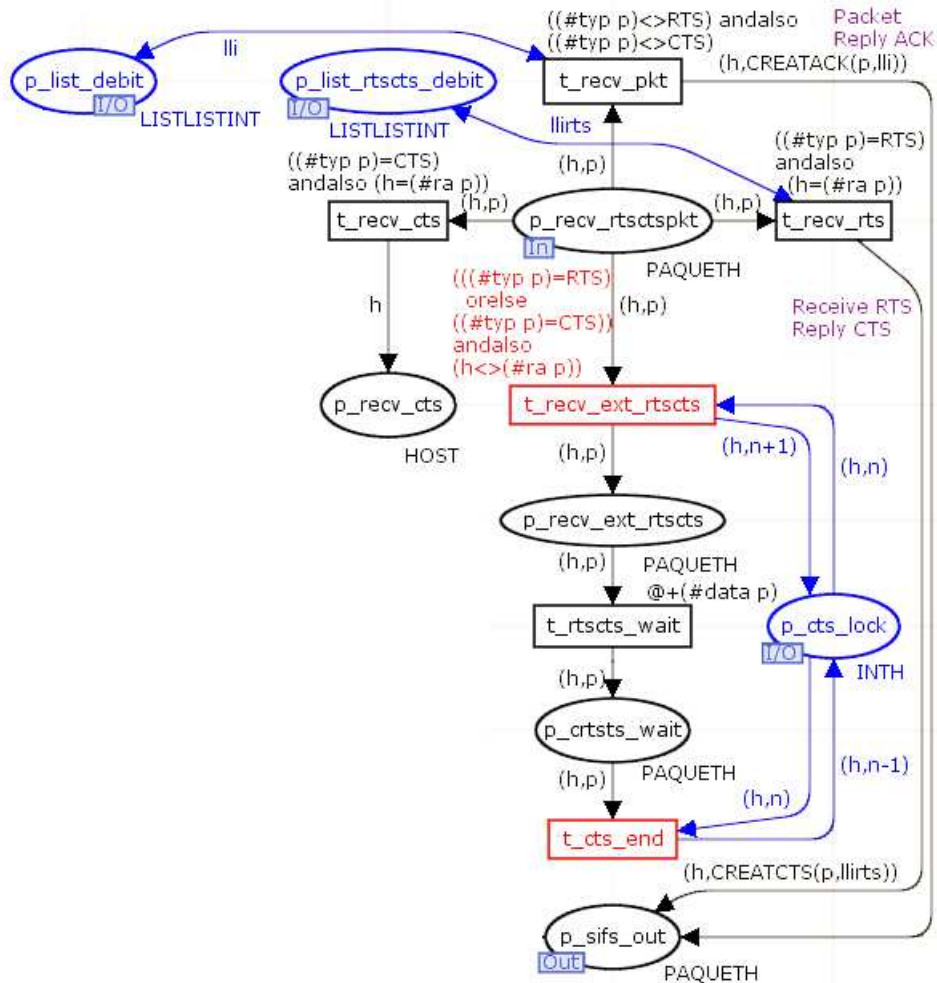


FIGURE 3.42 – Modèle *ACKRTSCTS* - 4<sup>ème</sup> étape - Réception de RTS et de CTS

Lorsqu'un RTS ou un CTS est reçu dans la couche *MAC*, celui-ci est transmis dans la place d'entrée  $p\_recv\_rtscs\_pkt$ . De même, la couche *MAC* copie les paquets reçus dans ce module car c'est elle qui va générer l'accusé de réception. Ceci est présenté figure 3.42.

- Si le paquet est un CTS destiné au nœud, la transition  $t\_recv\_cts$  le déplace dans la place  $p\_recv\_cts$  pour permettre l'émission de paquet décrite dans le paragraphe précédent.
- Si le paquet est un RTS destiné au nœud, la transition  $t\_recv\_rts$  crée un paquet CTS annonçant la durée de réservation du canal à partir de la durée contenue

dans la RTS à laquelle est soustraite la durée du CTS et un *SIFS*. Ce CTS est déposé dans la place de sortie *p\_sifs\_out* afin qu'il soit transmis à la couche *Radio* après un délai intertrame *SIFS*.

- Si un paquet qui n'est ni un RTS, ni un CTS est reçu, la transition *p\_recv\_pkt* crée un accusé de réception dans la place de sortie *p\_sifs\_out* pour qu'il soit émis après un délai intertrame *SIFS*.
- La réception de RTS ou de CTS non destiné au nœud courant, correspond à une demande de réservation de canal. Dans ce cas, la transition *t\_recv\_ext\_rtscts* modifie la place d'entrée/sortie *p\_cts\_lock* afin d'interdire l'émission. En effet, cette place est commune avec les modules *Iifs* et *Contention*. Ensuite la transition temporisée *t\_rtscts\_wait* attend la durée contenue dans le paquet RTS ou CTS, puis la transition *t\_cts\_end* libère la réservation de canal radio.

Ainsi, la figure 3.43 présente l'ensemble de la macro transition *ACKRTSCTS* qui permet l'émission d'accusé de réception et la gestion de la réservation de canal radio par le mécanisme RTS/CTS.

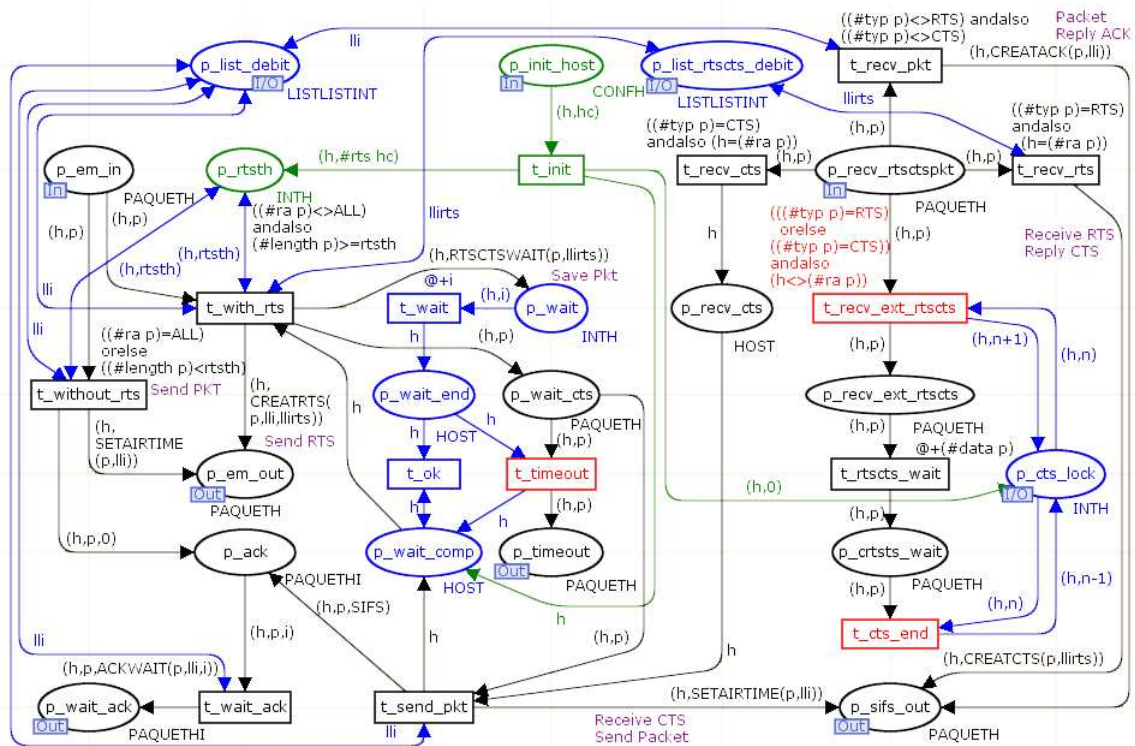
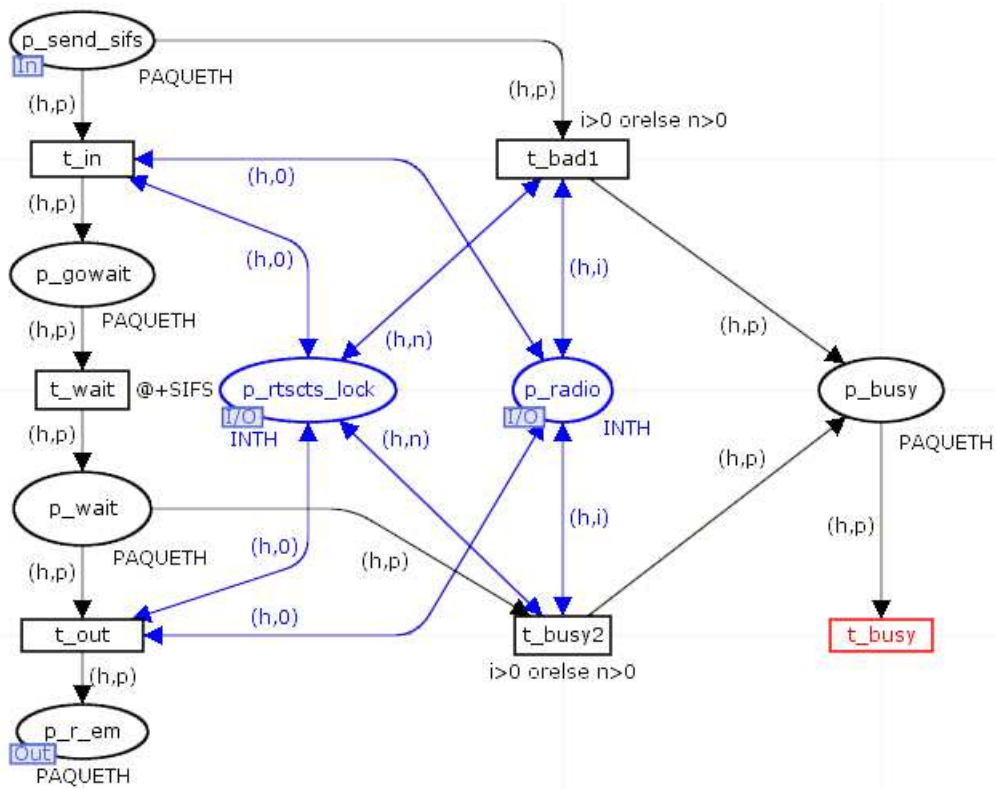


FIGURE 3.43 – Modèle *ACKRTSCTS* complet

#### 3.4.4.10 Description de la macro transition *SendWithSIFS*

La macro transition *SendWithSIFS* présentée sur la figure 3.44 est utilisée dans le modèle *MAC* pour implémenter un délai d'attente *SIFS* précédant les paquets de contrôle CTS, ACK ou les paquets suivant un CTS.

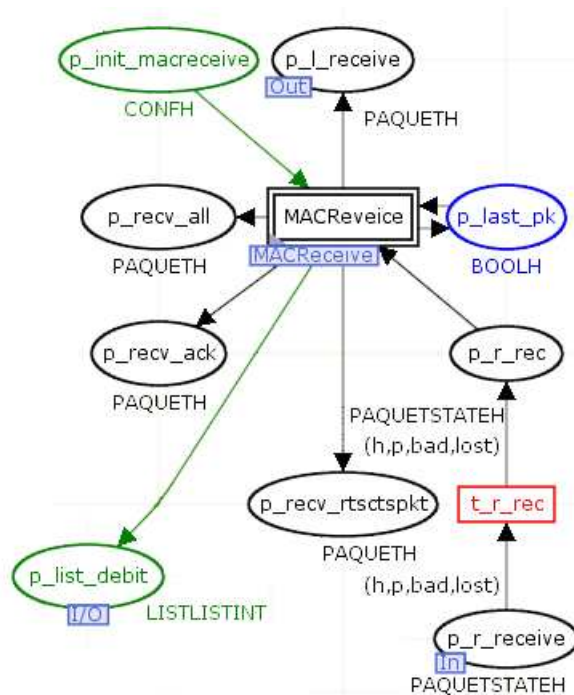
FIGURE 3.44 – Modèle *SendWithSIFS*

Le paquet qui doit être émis se présente par la place d'entrée  $p\_send\_sifs$ . Si le canal est libre, et qu'il n'est pas réservé, ce qui est représenté par la consultation des places  $p\_radio$  et  $p\_rtscs\_lock$ , la transition  $t\_in$  est franchie et permet la temporisation de la transition  $t\_wait$  pour la durée *SIFS*. Ensuite, si le canal est toujours libre et pas réservé, la transition  $t\_out$  dépose le paquet à la place  $p\_r\_em$  afin qu'il soit transmis à la couche *Radio* pour être émis.

Si le canal est occupé, ou s'il est réservé par le mécanisme RTS/CTS, le paquet est déplacé dans la place  $p\_busy$ . Il est ensuite supprimé pour ne pas être émis en même temps qu'une autre transmission ou pendant que le médium est réservé par un autre nœud.

### 3.4.4.11 6<sup>ème</sup> étape du modèle *MAC* : Réception de paquet

La figure 3.45 présente la réception d'un paquet dans le modèle *MAC*.

FIGURE 3.45 – Modèle *MAC* - 6<sup>ème</sup> étape - Réception de paquet

Lorsqu'un nœud reçoit un paquet, un jeton arrive dans la place  $p\_r\_receive$ . Celui-ci provient du module *Radio* et contient le nœud receveur, le paquet et deux booléens renseignant sur l'état du paquet. Le premier booléen est à vrai si le paquet est altéré ou a subi une collision, et le second est à vrai si le paquet a été reçu au moment où le nœud concerné émettait. Ce jeton est injecté dans le module *MACReveice* qui aiguille les paquets reçus.

Cette macro transition peut déposer des jetons dans les places suivantes :

- Si le paquet est correct et destiné au nœud courant, il est transféré à la couche *LLC* par la place de sortie  $p\_l\_receive$ .
- Si le paquet est du broadcast, destiné à tous les nœuds, il est déposé dans la place  $p\_recv\_all$ .
- Les accusés de réception reçus, destinés au nœud en cours, sont déposés dans la place  $p\_recv\_ack$ .
- Les RTS, CTS, ainsi qu'une copie des paquets sont déposés dans la place  $p\_recv\_rtsctsplt$ , pour être transmis au module *ACKRTSCTS*.
- La place  $p\_last\_pk$  doit contenir l'état du dernier paquet reçu pour définir le délai intertrame des prochains paquets à émettre. Elle est mise à jour par ce module.

#### 3.4.4.12 Description de la macro transition *MACReveice*

L'initialisation et l'introduction de paquet sont présentées dans la figure 3.46.

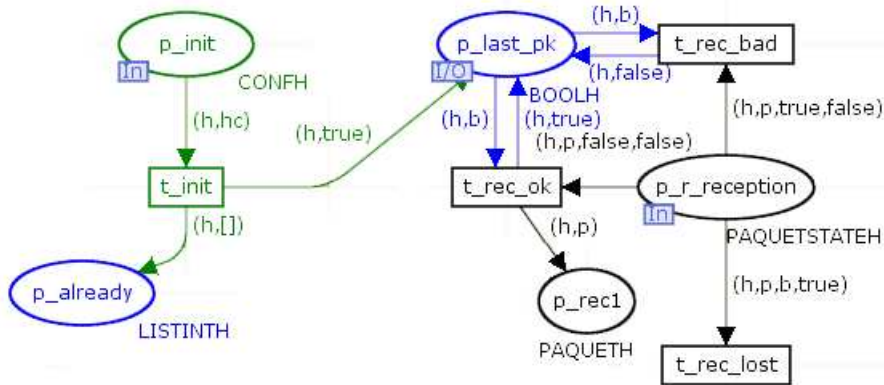


FIGURE 3.46 – Modèle *MACReceive* - 1<sup>ère</sup> étape - Initialisation et introduction de paquet

Lors de son initialisation, ce module reçoit un jeton de configuration pour chaque nœud. Pour chacun d’eux est créé un tableau vide dans la place *p\_already*. Celui-ci permet de mémoriser les identifiants de tous les paquets reçus, afin d’ignorer les paquets reçus en double. De même, la place *p\_last\_pk* est initialisée avec un booléen vrai pour chaque nœud. Celle-ci mémorise si le dernier paquet reçu est correct ou altéré. En effet, le module *Ifs* consulte cette place pour effectuer un délai intertrame plus long si le dernier paquet est altéré.

Lors de l’arrivée d’un paquet, si le second booléen est vrai, ceci signifie que le nœud était en cours d’émission quand le paquet est reçu. La transition *r\_rec\_lost* permet d’ignorer ce paquet. Sinon, si le paquet est reçu mais altéré, la transition *t\_rec\_bad* indique à la place *p\_last\_pk* que le dernier paquet est erroné, et supprime le paquet. Si le paquet est correct, la transition *r\_rec\_ok* indique que le dernier paquet est correct dans la place *p\_last\_pk*, et dépose le paquet dans *p\_rec1*.

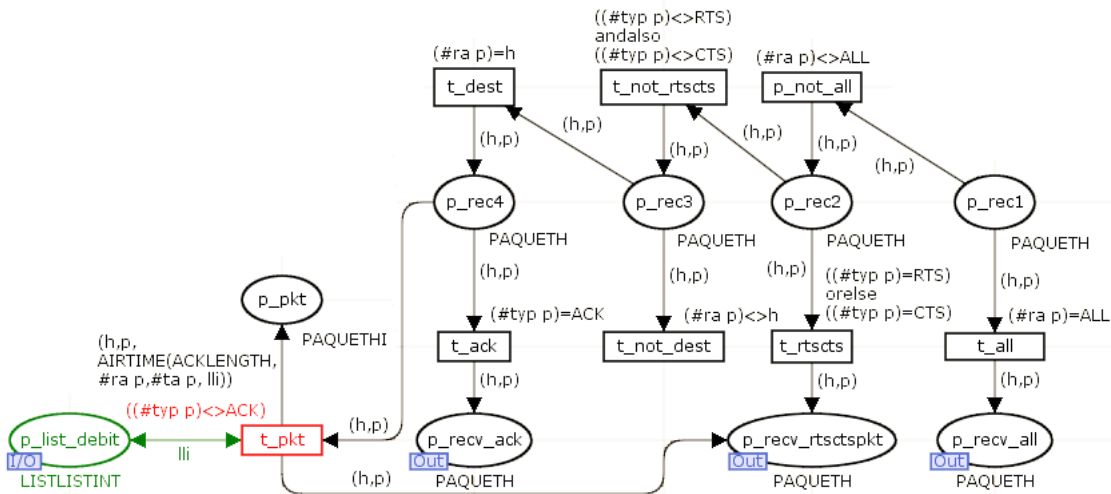


FIGURE 3.47 – Modèle *MACReceive* - 2<sup>ème</sup> étape - Aiguillage du paquet







qu'un accusé de réception doit être attendu pour un paquet. Si *RecvAck* n'a pas reçu d'accusé de réception dans le délai prévu, ce module replace le paquet en émission dans la place *p\_em1*. En cas d'accusé de réception correct, ce module supprime le compteur de réémissions de la place *p\_nb\_try*, et indique dans la place *p\_ready* que le nœud peut débiter une nouvelle émission.

### 3.4.4.14 Description de la macro transition *RecvAck*

La figure 3.51 présente la gestion de l'attente d'accusé de réception.

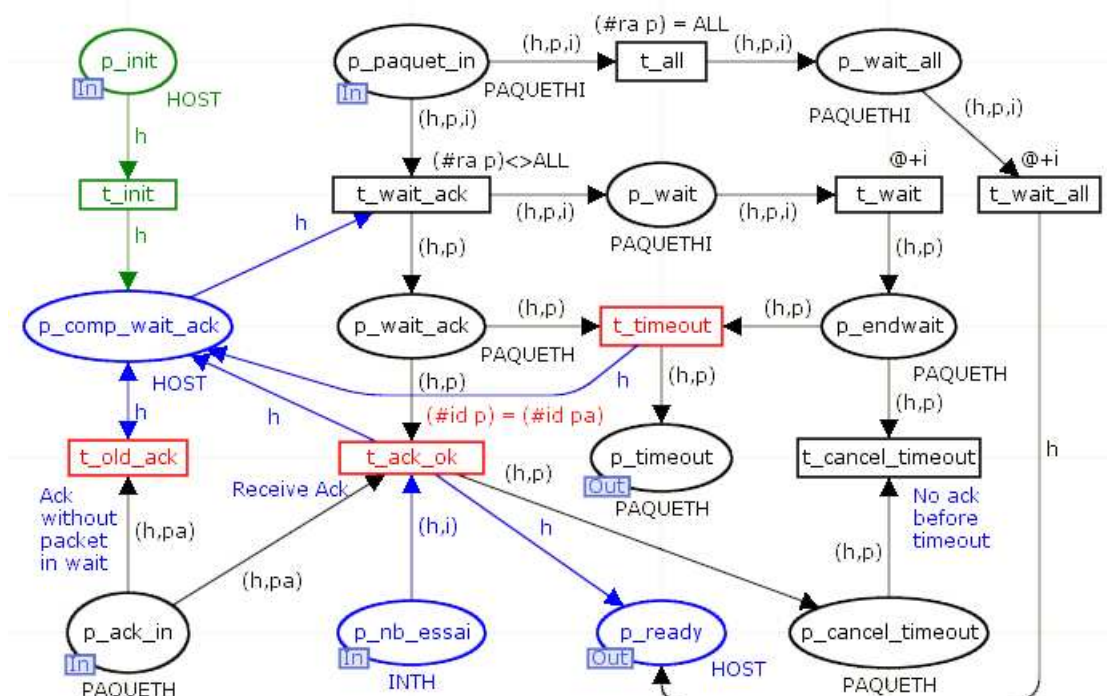


FIGURE 3.51 – Modèle *RecvAck* complet

Une copie du paquet émis arrive dans la place d'entrée *p\_paquet\_in* avec un entier indiquant combien de temps l'accusé de réception doit être attendu. Si le paquet n'est pas un paquet de diffusion, la transition *t\_wait\_ack* copie le paquet dans la place *p\_wait* pour que la transition *t\_wait* attende le délai prévu. Le paquet est aussi copié dans la place *p\_wait\_ack* en attente de l'accusé de réception correspondant. De plus, la transition *p\_wait\_ack* indique que le nœud est en attente d'accusé de réception en prenant le jeton dans la place *p\_comp\_wait\_ack*.

Si un accusé de réception n'est pas reçu suite à la transition temporisée *t\_wait*, la transition *t\_timeout* réinitialise ce module et dépose le paquet dans la place *p\_timeout* pour qu'il soit réémis. L'évènement associé au tir de cette transition est enregistré afin de retracer et de comptabiliser les échecs d'émission.

Si un accusé est reçu par la place d'entrée *p\_ack\_in* alors qu'un paquet est en



attente, la transition  $t\_ack\_ok$  est tirée, ce qui signifie que le paquet a été acquitté. Ceci permet :

- la suppression du compteur de réémissions dans la place  $p\_nb\_essai$ . En effet, celui sera recréé lors de la prochaine émission
- la création d'un jeton pour le nœud en cours dans la place  $p\_ready$ . Ceci indique que le nœud a terminé une émission et est disponible pour la suivante.
- la suppression du paquet dans la place  $p\_wait\_ack$  et le dépôt dans la place  $p\_cancel\_timeout$  pour réinitialiser le modèle à la fin de l'émission.

Cet évènement est aussi enregistré pour indiquer les acquittements de paquets.

Si un accusé de réception est reçu alors que nœud ne l'attendait pas, la transition  $t\_old\_ack$  permet de l'ignorer.

#### 3.4.4.15 8<sup>ème</sup> étape du modèle MAC : Emission périodique de beacon

La figure 3.52 présente l'émission périodique de beacon dans le module MAC.

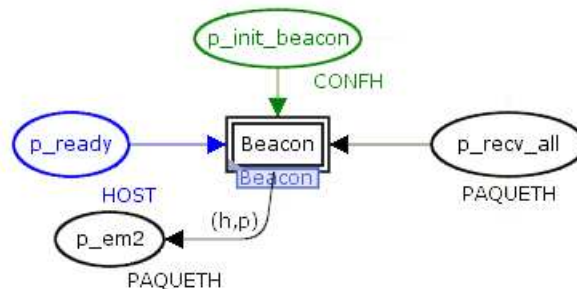


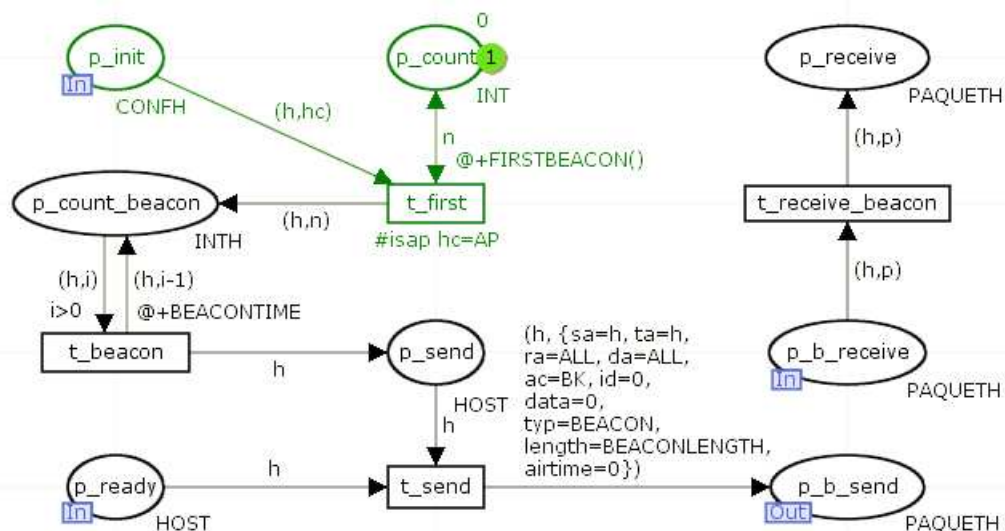
FIGURE 3.52 – Modèle MAC - 8<sup>ème</sup> étape - Emission périodique de beacon

Chaque nœud de type point d'accès émet périodiquement des paquets en diffusion, nommés beacons, pour indiquer qu'il est présent. Lorsqu'un beacon doit être émis, ce module prend le jeton indiquant que le nœud n'est pas en cours d'émission dans la place  $p\_ready$ , et dépose le paquet de type beacon généré dans la place  $p\_em2$  pour qu'il soit émis. Ce module est initialisé avec la configuration de tous les nœuds afin d'activer l'émission de beacon pour les points d'accès seulement.

La place  $p\_recv\_all$  permet de transmettre à ce module les paquets de type beacon reçus.

#### 3.4.4.16 Description de la macro transition Beacon

La figure 3.53 présente le modèle beacon.

FIGURE 3.53 – Modèle *Beacon* complet

Pour chaque nœud de type point d'accès, la transition  $t\_first$  crée un jeton identifiant le nœud associé au nombre de beacons à émettre. En effet, nous spécifions ce nombre pour que la simulation s'arrête au bout d'un temps défini. Cette transition est temporisée avec une durée aléatoire afin que tous les nœuds, démarrant la simulation en même temps, ne soient pas synchronisés. Ceci permet de se rapprocher de la réalité où il est très improbable que tous les nœuds démarrent strictement au même moment.

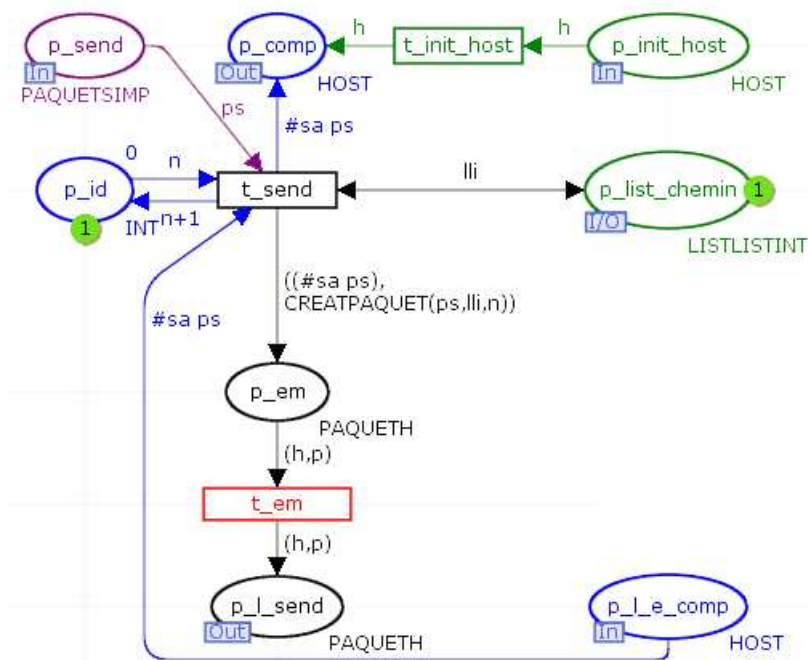
Ensuite, la transition  $t\_beacon$  génère un identifiant de nœud périodiquement pour chaque nœud dans la place  $p\_send$ . La fréquence est spécifiée par la constante *BEACONTIME*. Par défaut, celle-ci est configurée à 100ms. Cependant, ceci peut être un levier car cette fréquence peut être configurée dans le matériel que nous utilisons.

Ainsi, la transition  $t\_send$  génère un beacon si la place d'entrée  $p\_ready$  indique que le nœud n'est pas en cours d'émission. Le paquet généré est déposé dans  $p\_b\_send$  pour être émis par le modèle parent *MAC*.

#### 3.4.4.17 Description du modèle *MAC* complet

Les parties du modèle *MAC* présentées précédemment sont assemblées sur la figure 3.54. Ce modèle émet les paquets fournis par le modèle *LLC* et lui transmet les paquets reçus.



FIGURE 3.55 – Modèle LLC - 1<sup>ère</sup> étape - Emission de paquet

Les couches applications *ICMP* et *UDP* fournissent des paquets de couleur *PAQUETSIMP*. Cette couleur est une composition de :

- *sa* : identifiant du nœud source
- *da* : identifiant du nœud destination
- *ac* : classe de qualité de service de ce nœud
- *data* : identifiant du nœud
- *typ* : type du paquet (UDP ou ICMP)
- *length* : taille du paquet

Dans cette structure, nous n'avons pas les identifiants des nœuds intermédiaires transmetteur et receveur, car le rôle de ce module est de les configurer.

Le paquet est fourni par les couches supérieures via la place d'entrée  $p\_send$ . Le tir de  $t\_send$  n'est possible que si la place complémentaire  $p\_l\_e\_comp$  est marquée. Ceci indique que la file d'attente du module *MAC* n'est pas pleine. Ce tir détermine le prochain nœud qui doit recevoir le paquet en consultant la matrice de chemin contenue dans la place  $p\_list\_chemin$ . De plus, il indique dans la place complémentaire  $p\_comp$  que le paquet de la place  $p\_send$  a bien été consommé, et qu'un nouveau paquet peut être pris en compte. La place  $p\_id$  incrémente un entier pour compter le nombre de paquets introduits dans le réseau maillé.

Ensuite, la transition  $t\_em$  est tirée pour transmettre le paquet au module *MAC*. Celle-ci archive l'évènement indiquant qu'un paquet est transmis au module *MAC* pour être émis.

### 3.4.5.2 2<sup>ème</sup> étape du modèle *LLC* : Répétition de paquet

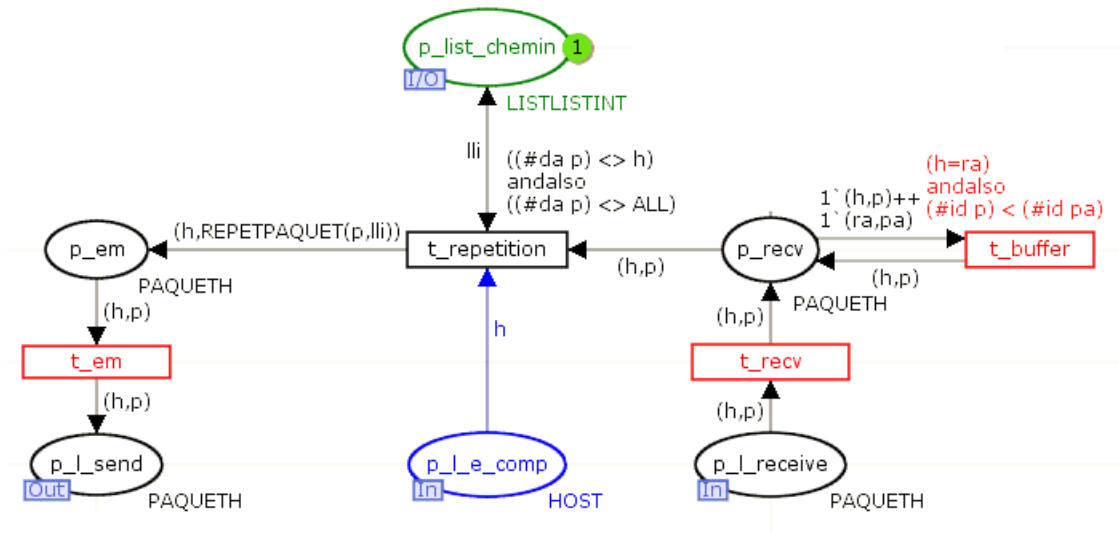


FIGURE 3.56 – Modèle *LLC* - 2<sup>ème</sup> étape - Répétition de paquet

Comme le présente la figure 3.56, le module *MAC* fournit les paquets reçus, ayant comme receveur le nœud en cours dans la place d'entrée  $p\_l\_receive$ . La transition  $t\_recv$  est tirée et archive l'évènement indiquant qu'un paquet est fourni par le module *MAC*. Le paquet est déposé dans la place  $p\_recv$ .

Si le paquet n'est pas destiné au nœud en cours, il doit être répété pour être acheminé jusqu'à sa destination. Si la place complémentaire  $p\_l\_e\_comp$  est marquée, ce qui indique que la file d'attente du module *MAC* n'est pas pleine, la transition  $t\_repetition$  consulte la matrice de chemin pour indiquer le prochain receveur, et le paquet est placé en émission dans la place  $p\_em$ .

Si la transition  $t\_repetition$  ne peut pas être tirée car la file d'attente du modèle *MAC* est pleine, le paquet reste dans  $p\_recv$ . Si un second paquet arrive dans cette place, la transition  $t\_buffer$  est tirée pour supprimer ce second paquet. Ceci interdit aux paquets d'être en attente dans le modèle *LLC*.

### 3.4.5.3 3<sup>ème</sup> étape du modèle *LLC* : Réception de paquet pour le nœud courant

La figure 3.57 présente l'acheminement de paquet reçu dans la couche supérieure.

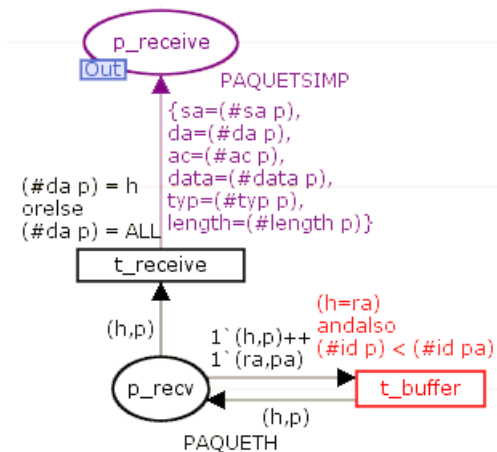


FIGURE 3.57 – Modèle *LLC* - 3<sup>ème</sup> étape - Réception de paquet pour le nœud courant

Si le paquet reçu est destiné au nœud courant, la transition  $t\_receive$  déplace le paquet dans la place  $p\_receive$  en le transformant en *PAQUETSIMP*. Les informations indiquant les nœuds transmetteur et receveur sont donc supprimées car elles ne concernent pas les couches supérieures.

### 3.4.5.4 Description du modèle *LLC* complet

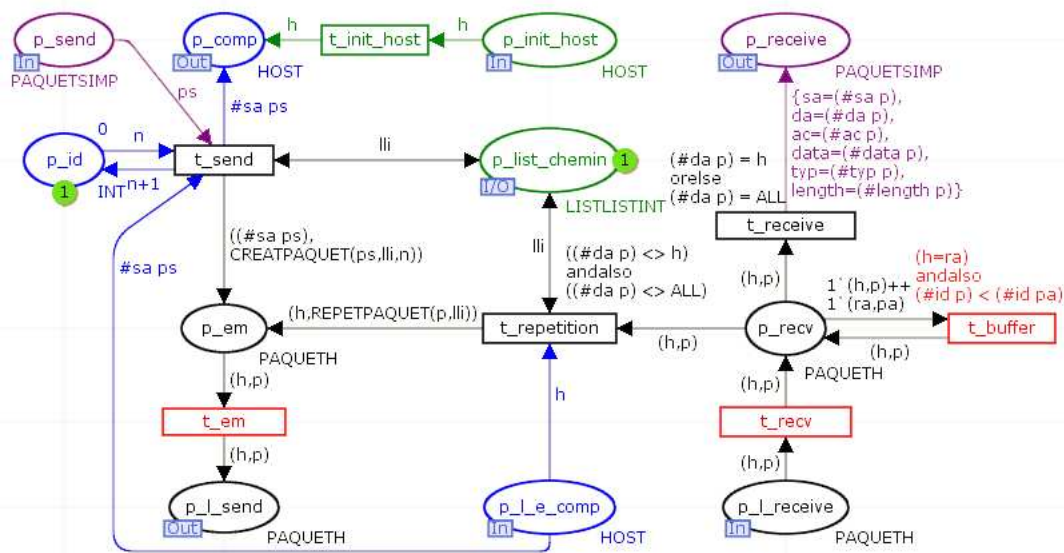


FIGURE 3.58 – Modèle *LLC* complet

La figure 3.58 présente le modèle *LLC* complet. Celui-ci ne permet des relais de paquet qu'en suivant une matrice de chemin configurée de manière statique. Cependant,

faire évoluer ce module pour implémenter des chemins dynamiques est possible sans toucher aux autres parties du modèle.

### 3.4.6 Modèle *UDP*

Le module *UDP* permet de définir la date, et la quantité de paquets à émettre tout en choisissant les sources et destinations. La description de ce modèle est décomposée en deux parties. L'émission de paquet, puis la réception.

#### 3.4.6.1 1<sup>ère</sup> étape du modèle *UDP* : Emission de paquet

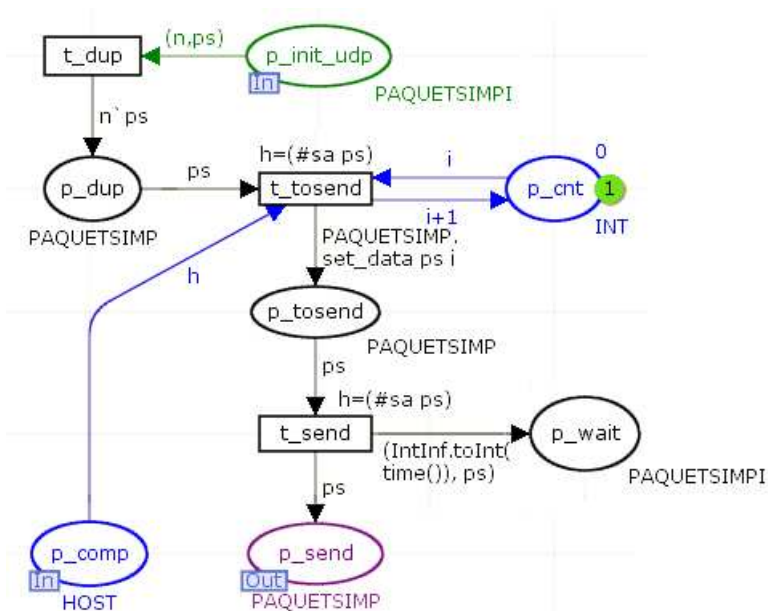


FIGURE 3.59 – Modèle *UDP* - 1<sup>ère</sup> étape - Emission de paquet

La figure 3.59 présente la génération de paquets. Ce modèle est configuré dans le modèle parent *Network* qui transmet la configuration dans la place *p\_init*. Ces paquets sont dupliqués par la transition *t\_dup* en fonction de la valeur configurée dans le modèle parent, et déposés dans *p\_dup*.

Si le module *LLC* accepte des paquets, ce qui est le cas si la file d'attente du modèle *MAC* n'est pas pleine, la place complémentaire *p\_comp* permet le tir de *t\_tosend*. Ce tir numérote le paquet à partir d'un compteur présent dans la place *p\_cnt* afin d'individualiser le champ *data* du paquet. Ensuite, la transition *t\_send* copie le paquet dans la place *p\_wait* en le datant afin de faire des statistiques de transmissions lors de l'arrivée du paquet sur un autre nœud, puis le dépose dans la place *p\_send* pour le transmettre au module *LLC*.



### 3.4.6.2 2<sup>ème</sup> étape du modèle *UDP* : Réception de paquet

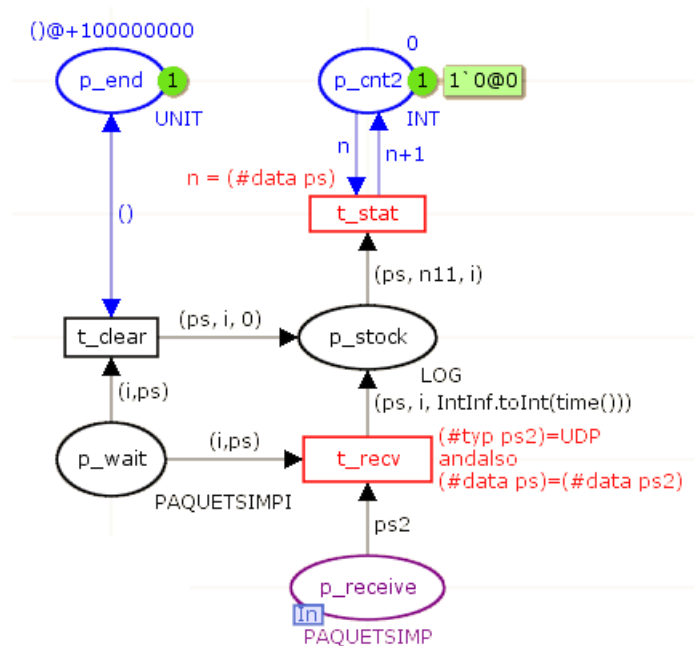


FIGURE 3.60 – Modèle *UDP* - 2<sup>ème</sup> étape - Réception de paquet

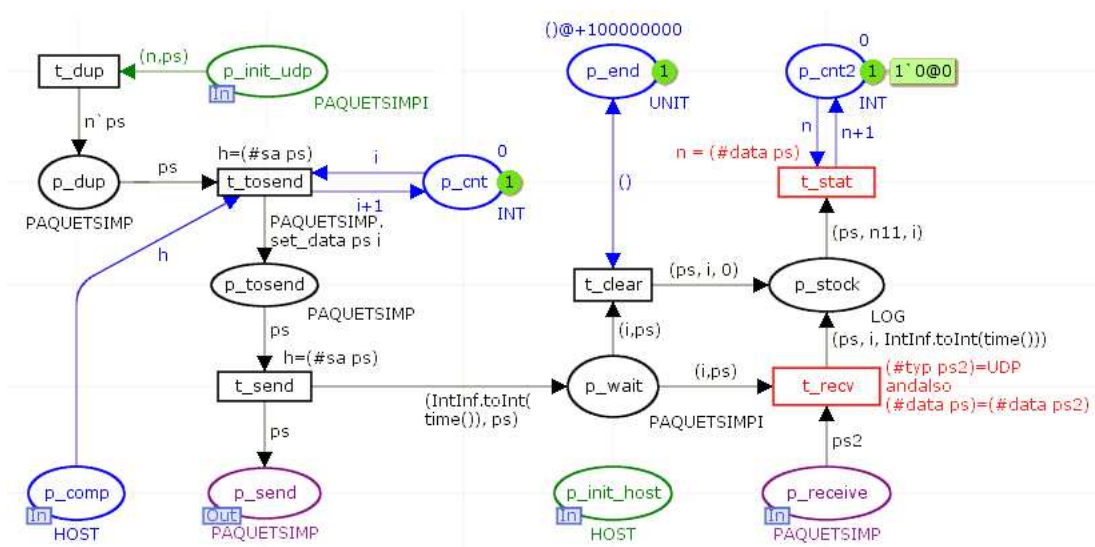
La réception de paquet est présentée sur la figure 3.60. Ils arrivent dans la place d'entrée *p\_receive*. La transition *t\_recv* filtre les paquets UDP, et lit la date de la génération du paquet dans la place *p\_wait*. Les dates de départ et d'arrivée sont déposées dans la place *p\_stock*. Si un paquet n'est pas arrivé au bout de 100 secondes, la transition *t\_clear* est tirée, ce qui enregistre une date d'arrivée de 0 dans la place *p\_stock*.

La transition *t\_stat* archive les dates de départ et d'arrivée des paquets dans un fichier dans l'ordre de création des paquets afin de permettre d'établir des statistiques de transmissions.

### 3.4.6.3 Description du modèle *UDP* complet

La figure 3.61 présente le modèle *UDP* complet.



FIGURE 3.61 – Modèle *UDP* complet

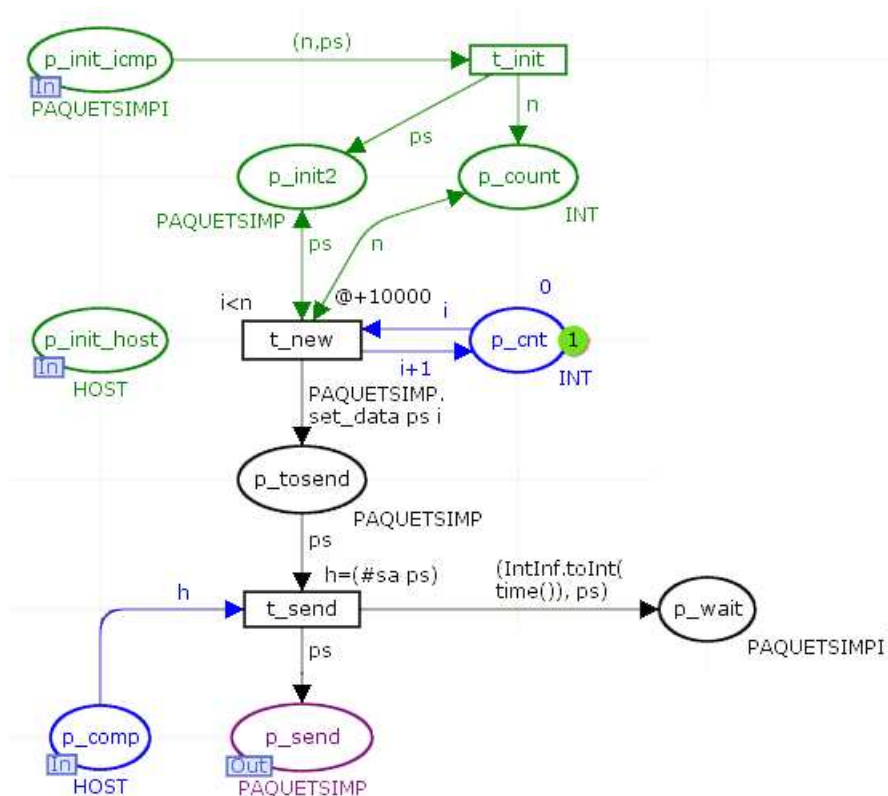
### 3.4.7 Modèle *ICMP*

De même que le modèle *UDP*, le modèle *ICMP* permet de générer des paquets. Cependant, il permet de mesurer le délai aller/retour des paquets.

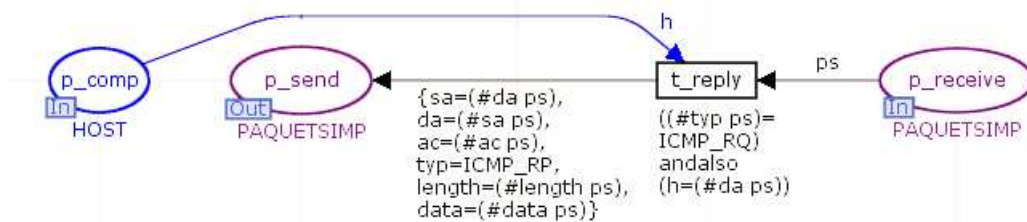
#### 3.4.7.1 1<sup>ère</sup> étape du modèle *ICMP* : Emission d'une requête

La configuration du flux à générer est faite dans le modèle parent *Network*. La figure 3.62 présente la génération de paquets. Les paquets se présentent dans la place d'entrée  $p\_init\_icmp$ . La transition  $t\_init$  sépare le type du paquet dans la place  $p\_init2$ , et le nombre de paquets à émettre dans la place  $p\_cout$ .

La fréquence d'émission de ce type de paquet est configurable au niveau de la transition  $t\_new$ . Cette transition est tirée pour chaque requête à émettre, toutes les 100 ms dans cet exemple. La place  $p\_cnt$ , modifiée à chaque tir de  $t\_new$ , permet d'individualiser le champ *data* de chaque paquet. Ensuite, la transition  $t\_send$  permet de transférer le paquet à émettre au module *LLC* par la place  $p\_send$ , si la place  $p\_comp$  indique que *LLC* peut mettre en file d'attente les paquets dans le modèle *MAC*. De plus, la transition  $t\_send$  copie le paquet en le datant dans la place  $p\_wait$  afin d'archiver les dates d'émission de ces paquets *ICMP*.

FIGURE 3.62 – Modèle *ICMP* - 1<sup>ère</sup> étape - Emission d'une requête

### 3.4.7.2 2<sup>ème</sup> étape du modèle *ICMP* : Réponse à une requête

FIGURE 3.63 – Modèle *ICMP* - 2<sup>ème</sup> étape - Réponse à une requête

La réponse aux requêtes ICMP est présentée sur la figure 3.63. Lorsque ce paquet est arrivé à son destinataire, il se présente dans la place  $p\_receive$ . C'est alors une requête ICMP. La transition  $t\_reply$  génère alors une réponse dans la place  $p\_send$  en inversant source et destination, et en spécifiant le type de paquet *ICMP\_RP* pour indiquer que c'est une réponse. Cette transition n'est franchissable que si les couches inférieures peuvent mettre ce paquet en file d'attente, selon la place complémentaire  $p\_comp$ .

### 3.4.7.3 3<sup>ème</sup> étape du modèle *ICMP* : Réception de réponses

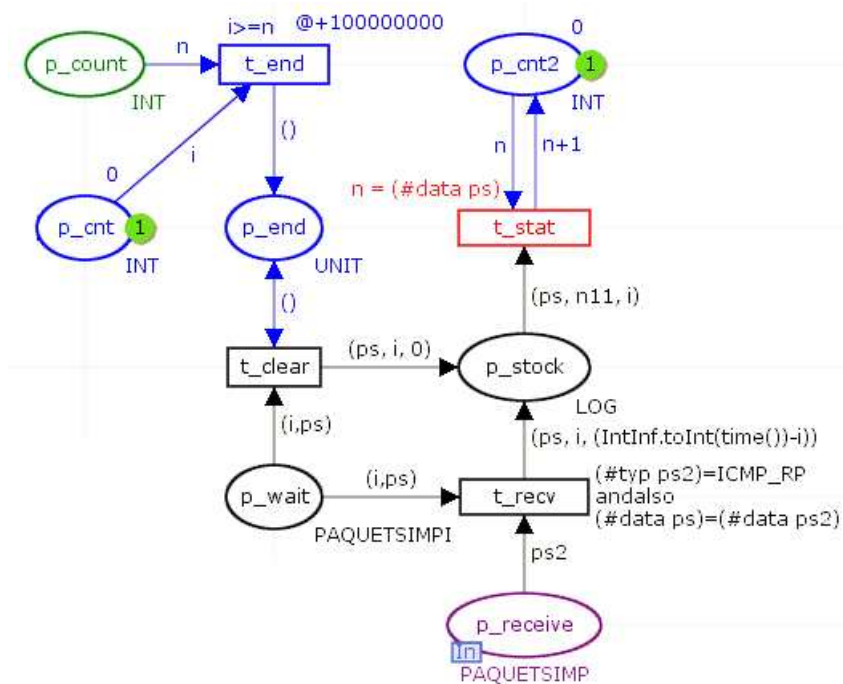


FIGURE 3.64 – Modèle *ICMP* - 3<sup>ème</sup> étape - Réception de réponses

La figure 3.64 présente l'arrivée d'une réponse ICMP. Dans ce cas, la transition *t\_recv* date cette arrivée, lit la date de départ dans la place *p\_wait*, et dépose ces dates dans la place *p\_stock*. Lorsque le compteur d'émission *p\_cnt* est à 0, la transition *t\_end* permet d'attendre 100 secondes que toutes les réponses arrivent. Ensuite, la transition *t\_clear* indique les paquets qui ont été perdus dans la place *p\_stock* en spécifiant une date d'arrivée de 0.

La transition *t\_stat* archive alors dans l'ordre des générations de paquets les dates de départ et d'arrivée de paquets.

### 3.4.7.4 Description du modèle *ICMP* complet

La figure 3.65 présente le modèle *ICMP* complet.

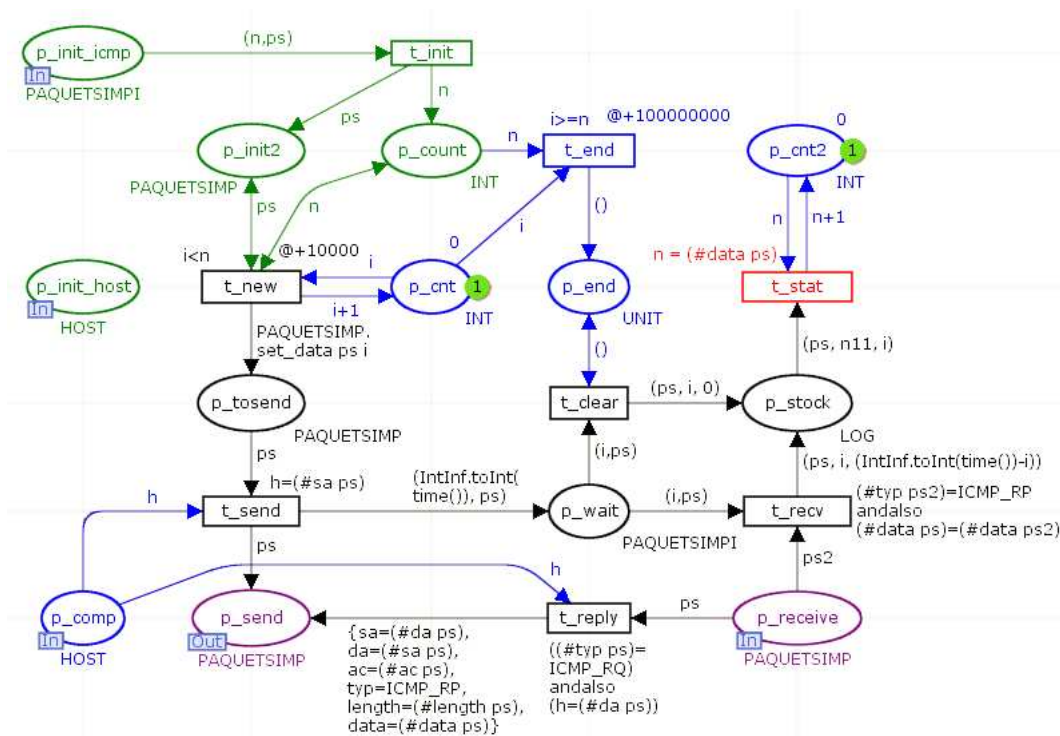


FIGURE 3.65 – Modèle ICMP complet

Ce module permet de mesurer le RTT entre plusieurs nœuds, avec ou sans autre transmission concurrente. Cependant, une première utilité est de valider ce modèle avec, dans un premier temps, la simulation d'une communication simple.

### 3.5 Validation

Il est nécessaire de valider que le simulateur est conforme aux spécifications initiales, et donc de vérifier que la modélisation effectuée correspond bien au comportement attendu. C'est pourquoi, dans un premier temps, nous pouvons énumérer les mécanismes implémentés dans le modèle afin de présenter par la suite la méthode de validation de leurs comportements dans le modèle. Ensuite, nous montrerons que le modèle met correctement en œuvre ces différents mécanismes, dont certains constituent des leviers sur lesquels nous allons agir pour estimer leur contribution à la gestion de la QoS.

De multiples simulations ont été effectuées, mais nous n'en présenterons que certaines dont les scénarii permettent de valider les différents mécanismes énumérés.

#### 3.5.1 Énumération des mécanismes à valider

Rappelons couche par couche les mécanismes qui doivent être validés au niveau du simulateur. Nous identifierons ces mécanismes par un label (**MRi**, **MMi**...) dans la mesure où nous les citerons dans la partie analyse.

### 3.5.1.1 Mécanismes de la couche Radio

L'objectif de la validation pour cette couche est de s'assurer de la conformité du modèle au regard des contraintes d'atteignabilité et de possibles altérations ou collisions.

- **MR1 : Transmission de paquet en fonction de la matrice d'atteignabilité**  
Lorsqu'un paquet est transmis à la couche radio, tous les récepteurs à portée doivent le recevoir, et ce en même temps. Cette couche va devoir interroger la matrice d'atteignabilité afin de dupliquer le paquet pour chacun des récepteurs pouvant capter ce paquet.
- **MR2 : Occupation des canaux radio par récepteurs**  
Les récepteurs doivent savoir s'ils captent une transmission ou non. Cette information permet à la couche MAC de décider d'émettre un paquet ou non car la norme 802.11 définit qu'une émission n'est possible que si le canal est libre.
- **MR3 : Détection de collision**  
Si plusieurs paquets sont captés en même temps au niveau d'un récepteur, avec un écart de niveau de réception inférieur à un seuil, celui-ci ne peut pas décoder de paquets. Les paquets devront être notés comme ayant subi une collision. Cependant, les paquets doivent être transmis à la couche MAC afin qu'elle puisse savoir que le dernier paquet reçu est erroné. En effet, le délai intertrame est différent en fonction de cette information.
- **MR4 : Perte de paquet si un émetteur reçoit un paquet alors qu'il est en train d'émettre**  
De la même manière, si un nœud capte un paquet alors qu'il est en cours d'émission, le paquet sera perdu car il ne pourra pas être capté par l'interface radio occupée. Dans le cas-là, le paquet sera ignoré à l'arrivée dans la couche MAC.
- **MR5 : Application d'une probabilité d'altération de paquet**  
Les perturbations radio sur le canal utilisé doivent affecter les transmissions. Certains paquets devront être altérés aléatoirement afin de simuler des erreurs de transmission. Cette probabilité est renseignée dans la matrice d'atteignabilité, et peut être différente entre chaque lien.

### 3.5.1.2 Mécanismes de la couche MAC

L'objectif de la validation pour cette couche est de s'assurer de la conformité du modèle au regard des spécifications de la norme 802.11, plus particulièrement de sa méthode d'accès CSMA/CA. L'effet du levier relatif à la classification des paquets sera également vérifié en terme d'application effective des priorités définies.

- **MM1 : Classification de paquet**  
Lorsqu'un paquet doit être émis, il entre dans un tampon d'entrée de la couche MAC. Celui-ci est composé de quatre files d'attente, classées de la priorité la plus forte à la moins élevée : VO (voix), VI (vidéo), BE (best effort), et BK (background). Cette classification devrait améliorer la qualité de service pour les paquets lorsque cela est nécessaire. En effet, ce mécanisme doit permettre, par exemple, d'émettre en priorité un paquet de voix et de laisser en attente un paquet moins urgent.
- **MM2 : Méthode d'accès CSMA/CA avec accusé de réception**

Cette méthode d'accès au médium définit que lorsqu'un nœud veut émettre, il doit attendre que le canal radio soit libre. De plus, un nœud ne pourra pas détecter d'erreur de transmission lorsqu'il émettra car il ne peut pas écouter le canal radio. Le récepteur devra donc émettre un accusé de réception afin d'acquitter le paquet reçu.

– **MM3 : Temporisation d'un temps intertrame**

La méthode d'accès du WiFi définit que le nœud émetteur doit vérifier que le canal soit libre pendant une durée appelée temps intertrame. Cette durée dépend de plusieurs paramètres. Ce délai est défini d'une part pour chaque classe de qualité de service et type du nœud émetteur (point d'accès ou station), et d'autre part en fonction de la bonne réception du dernier paquet.

– **MM4 : Temporisation d'une période de contention**

Si le paquet attend que le canal soit libre pour être émis, après avoir attendu la durée intertrame, il tire une durée de période de contention. La norme 802.11 définit que le nœud tire aléatoirement une durée décrite dans le chapitre modélisation.

– **MM5 : Suppression de paquet reçu en double**

Si un accusé de réception est perdu, l'émetteur du paquet ne sait pas que le receveur a bien reçu le paquet. Il va donc réémettre. La norme 802.11 prévoit de ne pas transmettre à la couche supérieure les doublons. Il doit cependant les acquitter de manière à éviter une nouvelle émission.

– **MM6 : Retransmission si l'accusé de réception n'est pas reçu**

Si un nœud ne reçoit pas d'accusé de réception à la suite d'un envoi de paquet, il doit réémettre. Si ce scénario se produit plus du nombre maximum de réémissions autorisées, le paquet ne doit plus être réémis.

– **MM7 : Utilisation conditionnée de paquet de contrôle RTS/CTS**

Si la taille du paquet à émettre est supérieure au seuil d'activation de ce mécanisme, un paquet de contrôle RTS devra être émis avant d'émettre le paquet afin de réserver le médium. Celui-ci devra contenir la durée pendant laquelle il réserve le canal WiFi.

– **MM8 : Réponse d'un CTS à un RTS**

Les nœuds respectant la norme 802.11 doivent répondre un CTS si un RTS leur est destiné.

– **MM9 : Blocage de l'émission en cas de réception de RTS ou CTS**

Les nœuds recevant des paquets de contrôle RTS ou CTS ne doivent plus émettre de paquet, à l'exception du nœud qui doit répondre le CTS à la suite d'un RTS, et du nœud devant transmettre le paquet à la suite du CTS.

– **MM10 : Emission périodique de beacon**

Un nœud de type point d'accès doit diffuser périodiquement des paquets pour signaler sa présence à des stations voulant s'y connecter.

### 3.5.1.3 Mécanismes de la couche LLC

Concernant la couche LLC, l'objectif de la validation porte principalement sur l'aiguillage des paquets au sein d'un nœud, notamment sur la bonne utilisation de la matrice de chemins pour ce faire.

– **ML1 : Recherche du receveur pour l'émission d'un paquet**

Lorsqu'un paquet provient de la couche supérieure, la couche LLC doit interroger la matrice de chemins afin de rajouter dans la structure du paquet l'identifiant du prochain nœud qui doit recevoir le paquet.

- **ML2 : Livraison d'un paquet arrivé à destination à la couche supérieure**  
Pour un paquet qui provient de la couche MAC, et qui est destiné au nœud courant, c'est-à-dire celui qui l'a reçu, le paquet doit être transmis à la couche supérieure.
- **ML3 : Acheminement d'un paquet reçu vers son prochain relais**  
Lorsqu'un paquet provient de la couche MAC, et qu'il n'est pas destiné au nœud courant, il doit être réémis afin d'être transmis vers son prochain relais. La couche LLC doit consulter la matrice de chemins afin de modifier l'identifiant du prochain nœud qui doit recevoir le paquet dans la structure de celui-ci.
- **ML4 : Suppression des paquets à répéter si la file d'attente d'émission est pleine**

Les files d'attente ont une taille limitée. Si la file d'attente correspondante à la classe d'un paquet à réémettre est pleine, celui-ci ne peut pas être fourni à la couche MAC. Le paquet est donc supprimé.

#### 3.5.1.4 Mécanismes du protocole ICMP

L'objectif ici est de valider l'implémentation (partielle) du protocole ICMP sur lequel nous allons nous appuyer pour effectuer des mesures sur l'architecture étudiée.

- **MI1 : Emission périodique de paquet**  
Ce protocole doit émettre périodiquement des requêtes ICMP afin qu'elles soient acheminées à travers le réseau. La date d'émission doit être mémorisée afin de pouvoir calculer le RTT (Round Trip Time - temps aller-retour)
- **MI2 : Réponse aux requêtes ICMP**  
Lorsqu'une requête ICMP est reçue par le modèle de ce protocole, une réponse doit être émise à son expéditeur.
- **MI3 : Mémorisation des RTT**  
A chaque réception de réponse ICMP, le RTT doit être calculé par rapport à la date de l'émission de la requête. Cette valeur doit être mémorisée afin d'être rapportée à la fin de la simulation.

#### 3.5.1.5 Mécanismes du protocole UDP

Le recours au protocole UDP a pour finalité l'injection de trafic sur l'architecture étudiée, selon différents scénarii ; l'objectif ici est donc de s'assurer du bon fonctionnement de son implémentation (partielle).

- **MU1 : Emission de paquet**  
Cette couche permet d'émettre des paquets suivant des scénarii définis par l'utilisateur. Il peut décider notamment de la taille et de la date d'émission du paquet.



### 3.5.2 Présentation du cas d'étude

Nous allons commencer par présenter la topologie que nous allons utiliser. Celle-ci est tirée d'un cas réel correspondant à une installation de WiFi maillé existante dans un camping nommé "le Plein Air des Chênes" dans l'Hérault. En effet, utiliser un site où les caractéristiques sont connues permet de configurer le simulateur avec des valeurs de paramètres pertinentes et, dans une certaine limite, de confronter la simulation à la réalité.

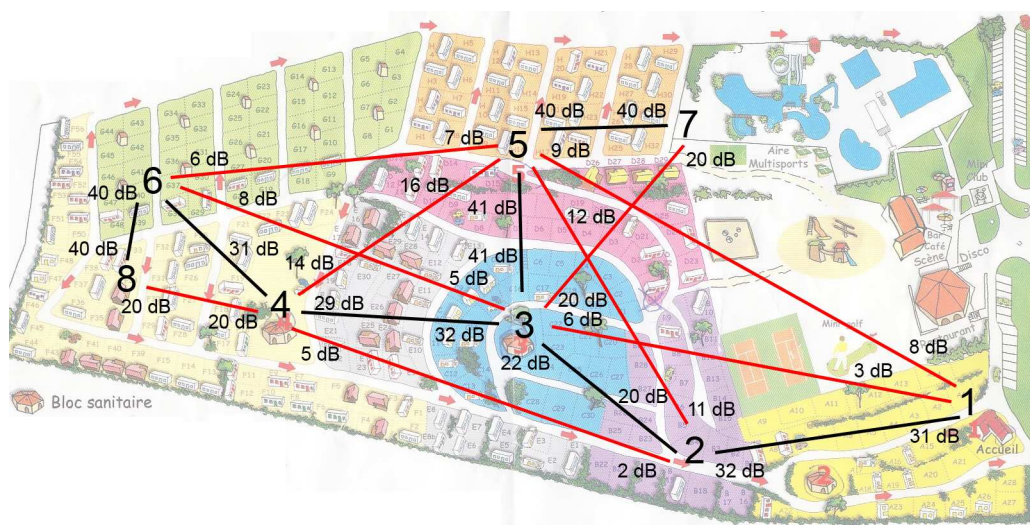


FIGURE 3.66 – Plan du cas d'étude

Le cas d'un camping permet de relever plusieurs problèmes. Tout d'abord, des arbres permettent aux vacanciers d'avoir de l'ombre, mais altèrent aussi les communications radio sur la bande de fréquence de 2,4 GHz. De plus, le relief du terrain qui ne peut être lisible sur la carte affaiblit aussi certaines communications. Ainsi, des erreurs de communications dues à l'environnement ou au problème du nœud caché apparaissent.

Identifiant	<i>isap</i>	<i>rts</i>	<i>buf</i>	<i>aifs</i>				<i>cwmin</i>				<i>cwmax</i>			
				vo	vi	be	bk	vo	vi	be	bk	vo	vi	be	bk
1	AP	2000	100	2	2	2	2	2	3	3	4	3	4	10	10
2	AP	2000	100	2	2	2	2	2	3	3	4	3	4	10	10
3	AP	2000	100	2	2	2	2	2	3	3	4	3	4	10	10
4	AP	2000	100	2	2	2	2	2	3	3	4	3	4	10	10
5	AP	2000	100	2	2	2	2	2	3	3	4	3	4	10	10
6	AP	2000	100	2	2	2	2	2	3	3	4	3	4	10	10
7	STA	2000	100	7	7	7	7	2	3	3	4	3	4	10	10
8	STA	2000	100	7	7	7	7	2	3	3	4	3	4	10	10

TABLE 3.6 – Configuration des nœuds du cas d'étude



La figure 3.66 présente la position des bornes sur un plan, ainsi que le niveau de réception des liens mesuré entre chaque nœud ainsi que le débit physique réellement utilisé en pratique. Les nœuds 1 à 6 sont des points d'accès maillés aux positions indiquées sur le plan, et les nœuds 7 et 8 sont des stations respectivement proches des bornes 5 et 6. Le tableau 3.6 présente les différents nœuds avec leurs paramètres. De plus, cette topologie permet de renseigner les matrices de rapports signal sur bruit (table 3.7) et de chemins (table 3.8) qui pourront être intégrées dans le simulateur.

		<i>Nœud récepteur</i>							
		1	2	3	4	5	6	7	8
<i>Nœud émetteur</i>	1	0	33	1	0	1	0	0	0
	2	35	0	16	6	18	0	0	0
	3	1	16	0	19	29	3	20	0
	4	0	6	18	0	9	34	20	20
	5	1	17	29	8	0	2	40	0
	6	0	0	3	35	4	0	0	40
	7	0	0	20	20	40	0	0	0
	8	0	0	0	20	10	40	0	0

TABLE 3.7 – Matrice de rapport signal sur bruit du cas d'étude

		<i>Nœud destinataire</i>							
		1	2	3	4	5	6	7	8
<i>Nœud source</i>	1	1	2	2	2	2	2	2	2
	2	1	2	3	3	3	3	3	3
	3	2	2	3	4	5	4	5	4
	4	3	3	3	4	3	6	3	6
	5	3	3	3	3	5	3	7	3
	6	4	4	4	4	4	6	4	8
	7	5	5	5	5	5	5	7	5
	8	6	6	6	6	6	6	6	8

TABLE 3.8 – Matrice de chemins du cas d'étude

### 3.5.3 Graphiques générés

Les simulations effectuées par notre logiciel fournissent des fichiers texte reportant les événements. Ceci est difficilement exploitable. Ces traces sont donc converties sous forme graphique par un script conçu à cet effet.

Ces graphes représentent ce qu'il s'est passé lors d'une simulation au cours du temps. Le temps est donc en abscisse en microseconde. En ordonnée, chaque ligne correspond à un nœud du réseau. La légende est présentée figure 3.67.

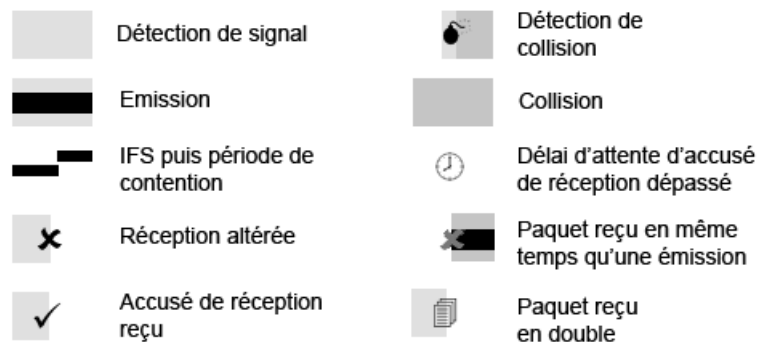


FIGURE 3.67 – Légende des graphes

- **Détection de signal**  
La surface correspondante à un nœud qui est dans une zone où se propage un signal sera grisée. Ceci permet de voir quels nœuds sont influencés par une émission.
- **Emission**  
Le trait noir épais signifie qu'un nœud est en train d'émettre.
- **IFS puis période de contention** : Les deux traits précédant une émission représentent successivement le temps d'attente intertrame, puis la durée de la période de contention.
- **Réception altérée**  
Cette croix peut être présente à la fin d'une transmission pour chaque receveur. Sa présence signifie que la réception a échoué suite au tir aléatoire de la décision.
- **Accusé de réception reçu** : Ce symbole précise qu'un nœud a reçu l'accusé de réception qui lui était destiné, ce qui acquitte la transmission.
- **Détection de collision**  
Lorsqu'un récepteur reçoit deux paquets à la fois, il y a collision. Ce symbole apparaît lorsqu'un paquet est marqué comme ayant reçu une collision.
- **Collision**  
Lorsque deux paquets sont en collision, la surface correspondante au nœud qui reçoit plusieurs paquets en même temps est en gris foncé.
- **Délai d'attente d'accusé de réception dépassé**  
Ce symbole définit qu'un accusé de réception n'a pas été reçu dans le délai imparti par le nœud qui a émis le paquet.
- **Paquet reçu en même temps qu'une émission**  
Si deux nœuds voisins décident d'émettre au même moment, ils ne pourront pas recevoir le paquet de l'autre car ils sont en émission. Ce symbole, situé en début d'émission indique la perte d'un paquet à cause de ce mécanisme.
- **Paquet reçu en double**  
Si un nœud reçoit deux fois le même paquet, la norme 802.11 prévoit de ne délivrer que le premier paquet à la couche supérieure. Ce symbole indique que ce paquet est en double, et qu'il ne sera pas acheminé à la couche supérieure.

### 3.5.4 Simulation d'une requête ICMP sur le cas d'étude

#### 3.5.4.1 Configuration des paquets à transmettre lors de cette simulation

Afin d'effectuer seulement une requête ICMP, les modules *UDP* et *Beacon* sont désactivés. En effet, un jeton contenant l'entier 0 initialisera la place *p\_count* du modèle *Beacon*, et la place *p\_conf\_udp* du modèle *Network* sera vide. Le modèle *ICMP* est configuré en plaçant le jeton suivant dans la place *p\_conf\_icmp* du modèle *Network* :

- Source : Nœud 1
- Destination : Nœud 8
- Classe de qualité de service : BK (background)
- Type de paquet : ICMP\_RQ (Requête ICMP)
- Taille du paquet : 500 octets.
- Identifiant du paquet ICMP : 0

#### 3.5.4.2 Analyse du graphe généré

Le graphe généré à partir des traces enregistrées lors de cette simulation est présenté sur la figure 3.68. Nous allons détailler ce graphe pour mettre en évidence que les différents mécanismes mis en jeu ont été modélisés correctement.

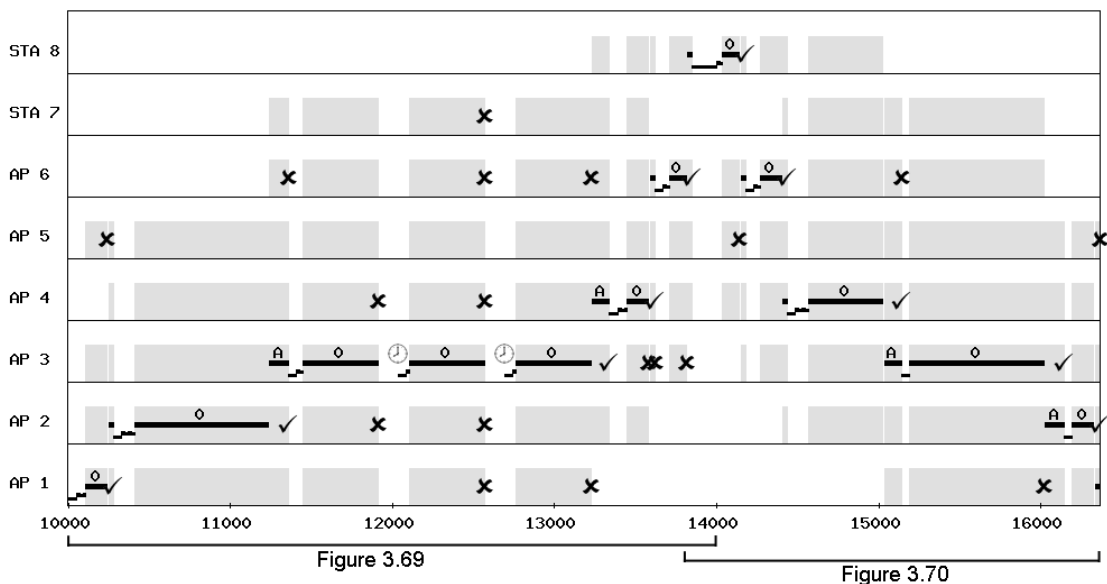


FIGURE 3.68 – Graphe généré pour la transmission d'un échange ICMP

Nous allons examiner dans un premier temps la transmission entre le nœud 1 et le nœud 8 (zoom présenté sur la figure 3.69). Ensuite, nous pourrions étudier la réponse.

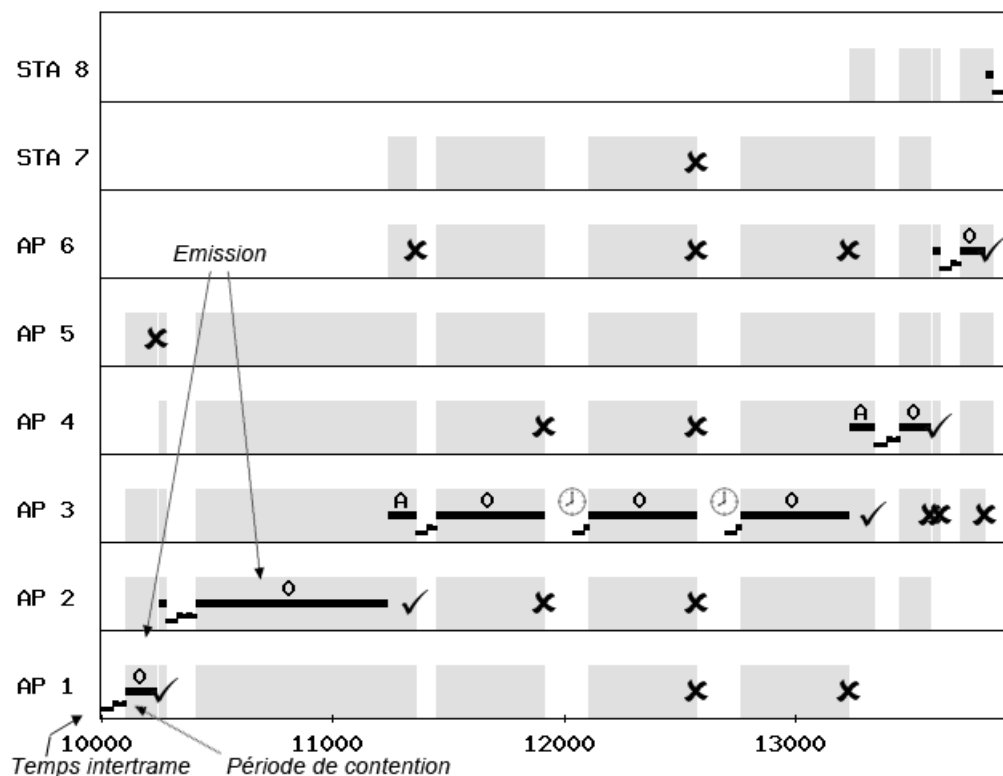


FIGURE 3.69 – Première partie du graphe généré lors d'un échange ICMP

Au temps 0, le nœud AP1 doit émettre la requête ICMP vers STA8. La couche *LLC* consulte la matrice de chemins pour connaître le prochain receveur : il s'agit de AP2, comme le définit le mécanisme ML1. Après avoir vérifié que le canal radio soit libre, comme le spécifie le CSMA/CA (MR2 et MM7), il attend la durée d'un temps intertrame pour station tel que défini par MM3. Ensuite, il va tirer aléatoirement une période de contention dans l'intervalle  $[0; 4]$  comme le prévoit le mécanisme MM4. La valeur 3 a été tirée, ce qui correspond à un délai de 60  $\mu$ s.

Ensuite, le nœud AP1 peut émettre le paquet, ce qui est représenté par le trait désigné par *Emission*. L'entier 0 au dessus du trait d'émission indique la valeur du champs data du paquet. Les nœuds AP1, AP2, AP3 et AP5 sont grisés comme cela est défini par le mécanisme MR1, pour indiquer qu'ils sont dans une zone détectant cette transmission. Donc, la couche Radio a bien consulté la matrice d'atteignabilité puisque les autres nœuds ne perçoivent pas cette transmission ; en effet, la configuration définit que seuls ces nœuds sont dans la zone de réception de AP1. Notons que les nœuds AP3 et AP5 ont aussi reçu le paquet, ce qui est représenté par le fond grisé. Cependant, ils ne donnent pas suite car ils n'étaient pas concernés par ce paquet. De plus, la croix au niveau de AP5 indique que ce nœud n'a pas reçu le paquet correctement car la réception a été altérée par l'environnement.

En réponse à ce paquet, le nœud AP2 qui a correctement reçu le paquet de AP1, attend un délai intertrame d'une durée SIFS définie dans la norme (MM3), puis émet

un accusé de réception nécessaire à CSMA/CA (MM2). Celui-ci est symbolisé par le premier trait noir dans la ligne AP2. La durée de cette émission est grisée pour les nœuds AP1, AP2, AP3, AP4 et AP5 car ils reçoivent cette transmission conformément à ce qu'indique la matrice d'atteignabilité. Le nœud AP1 ayant correctement reçu l'accusé de réception, il ne devra pas effectuer de réémission. Ceci est représenté par le signe "accusé de réception reçu".

Le nœud AP2 ayant correctement reçu ce paquet, il doit le réémettre à AP3 afin qu'il soit acheminé vers son destinataire. La couche *LLC* a consulté la matrice de chemins comme défini par le mécanisme ML3. Il attend donc un délai intertrame puis une période de contention. Le délai intertrame est plus court dans le cas d'un point d'accès que dans le cas d'une station tel que spécifié par le mécanisme MM3. La période de contention est tirée aléatoirement comme le définit le mécanisme MM4. Ensuite, le nœud AP2 émet le paquet. La transmission dure plus longtemps car le débit physique de ce lien est configuré plus lent mais elle se déroule correctement et le nœud AP3 reçoit le paquet. Celui-ci émet le paquet à AP4 mais lors de cette transmission le tir aléatoire du mécanisme MR5 indique que le paquet reçu par le nœud AP3 a été altéré. Ceci est indiqué par la croix sur la ligne de AP4. Vu qu'il n'a pas reçu correctement le paquet, il ne va pas émettre un accusé de réception. Le délai d'attente d'accusé de réception du nœud AP2 est écoulé (MM6), ce qui est représenté par le symbole "délai d'attente d'accusé de réception dépassé", et le paquet est alors réémis par le nœud AP3.

Le paquet est ensuite acheminé jusqu'au nœud STA8, selon les mêmes mécanismes que ceux que nous venons de valider. Les nœuds AP5 et STA7 n'ont pas été sollicités car ils ne sont pas impliqués dans le chemin allant du nœud 1 au nœud 8.

La réponse par le nœud STA8 à la requête ICMP provenant du nœud AP1 est disponible sur la figure 3.70.

Une fois la requête ICMP reçue, le nœud STA8 doit alors émettre sa réponse au nœud AP1, tel que défini par le mécanisme MI2. L'émission du nœud STA5 valide que la couche *LLC* (mécanisme ML2) a bien délivré le paquet à la couche ICMP du destinataire.

Cette simulation met en œuvre les mécanismes MR1, MR2, MR5, MM1, MM2, MM3, MM4, MM6, ML1, ML2, ML3 et MI2 implémentés dans le modèle. Leurs fonctionnements correspondent à la norme 802.11 et/ou aux spécifications initiales représentées sur notre modèle.

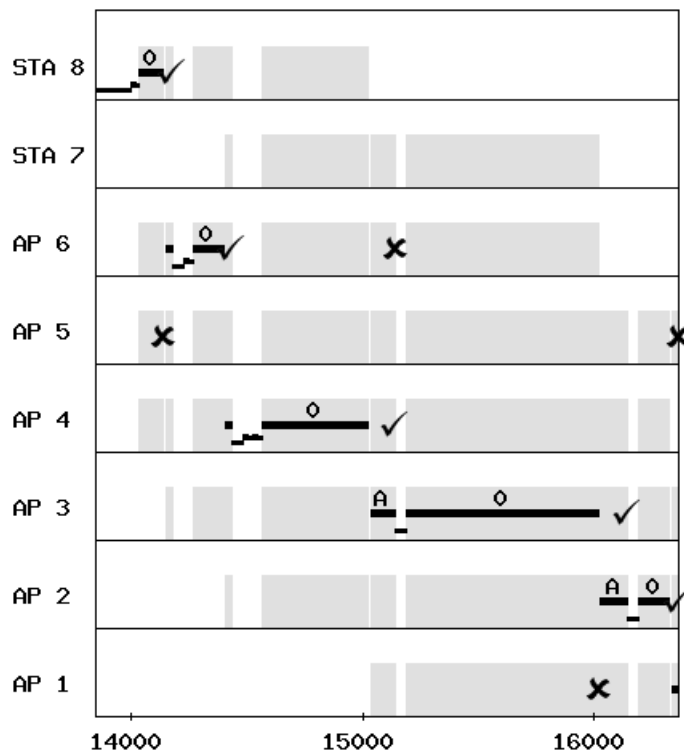


FIGURE 3.70 – Seconde partie du graphe généré lors d'un échange ICMP

### 3.5.5 Simulation de requêtes ICMP en concurrence avec des requêtes UDP

Cette simulation nous permet de compléter la validation des mécanismes exposés précédemment.

#### 3.5.5.1 Configuration des paquets à transmettre lors de cette simulation

Lors de cette simulation, nous utiliserons d'une part des requêtes ICMP, et d'autre part des émissions UDP. Le modèle *Beacon* sera désactivé en initialisant la place *p\_count* par un jeton contenant l'entier 0.

Les paramètres des paquets ICMP sont identiques à la simulation précédente à la différence que nous émettrons 3 requêtes ICMP. Ces paquets ayant pour classe de qualité de service BK, ils auront la priorité la moins importante. La temporisation de la transition *t\_new* est de 10000  $\mu$ s. Ceci signifie qu'une requête ICMP sera émise chaque 10ms, et que la première partira à la date 10 ms.

Trois émissions UDP sont programmées pour cette simulation. La place *p\_conf\_udp* du modèle *UDP* contiendra les paquets suivants :

- à la date 26000  $\mu$ s, émission d'un paquet UDP de classe VI (vidéo), et de taille 1500 octets allant du nœud 1 au nœud 5.

- à la date 34000  $\mu\text{s}$ , émission d'un paquet UDP de classe VI (vidéo), et de taille 1500 octets allant du nœud 1 au nœud 5.
- à la date 39000  $\mu\text{s}$ , émission d'un paquet UDP de classe VO (voix), et de taille 1000 octets allant du nœud 2 au nœud 4.

Cette date étant la date de mise en émission, le paquet sera placé dans la file d'attente correspondante à la classe du paquet et ne sera émis que lorsque le canal sera libre.

### 3.5.5.2 Analyse du graphe généré

Nous allons détailler ce graphe, généré à partir du scénario décrit précédemment.

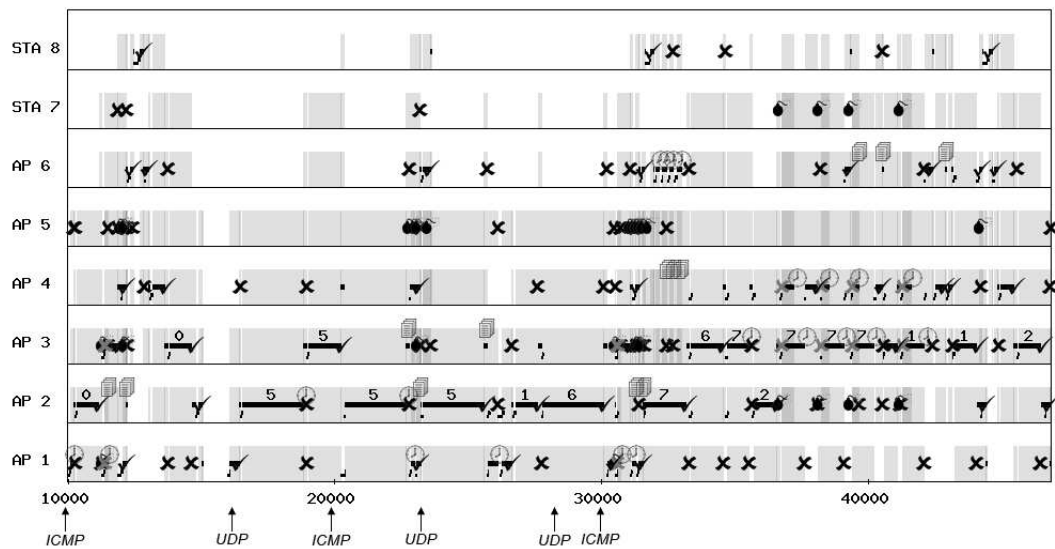


FIGURE 3.71 – Graphe généré pour la transmission d'échanges ICMP et UDP

Le graphique 3.71 nous permet de valider la mise en file d'attente périodique des requêtes ICMP, décrite par le mécanisme MI1. En effet, chaque flèche sous le graphe indique une date à laquelle est programmée la mise en file d'attente de paquet. La première requête ICMP est émise à la date 10000  $\mu\text{s}$ . Le canal étant occupé à la date 20000  $\mu\text{s}$ , la seconde est émise légèrement plus tard. De même pour la troisième, elle n'est émise qu'un peu après 30000  $\mu\text{s}$ . De plus, les paquets UDP sont bien émis aux dates demandées, ce qui permet de valider le mécanisme MU1.

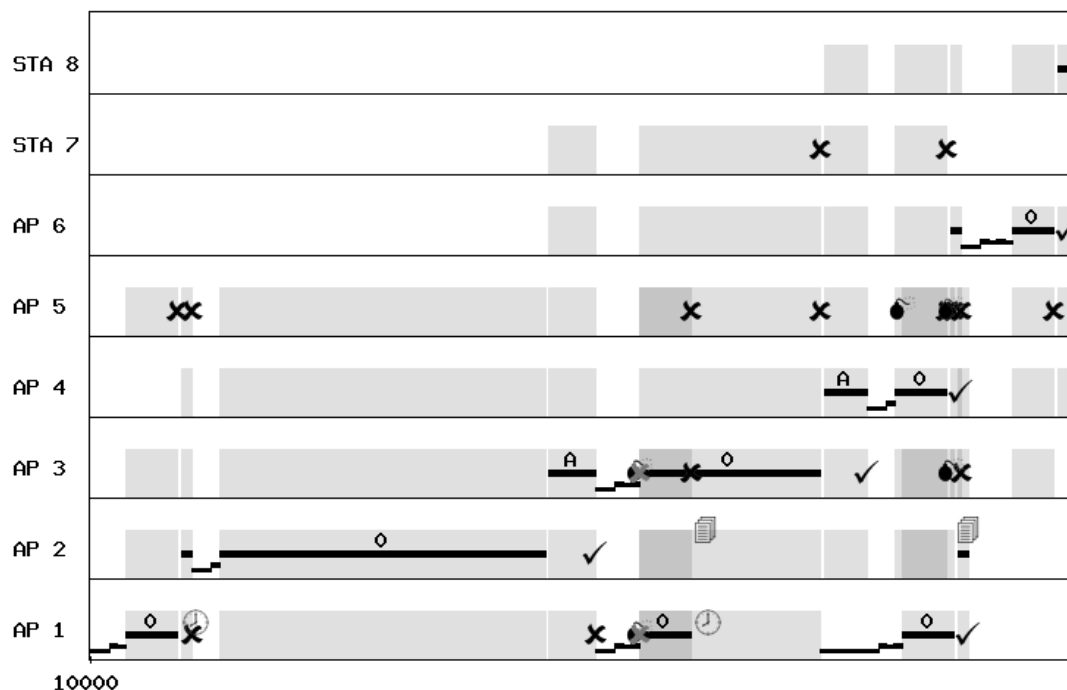


FIGURE 3.72 – Echanges ICMP et UDP zoomée entre 10000  $\mu$ s et 13000  $\mu$ s

Le graphique 3.72 correspond au résultat de la simulation zoomé entre les dates 10000  $\mu$ s et 13000  $\mu$ s.

Lorsque le nœud AP1 émet sa première requête ICMP, le nœud AP2 la reçoit correctement. Ce dernier émet donc un accusé de réception qui a été altéré. Le nœud AP1 ne sait donc pas que le nœud AP2 a correctement reçu le paquet. Le nœud AP2 transmet correctement le paquet au nœud AP3. Puis, les nœuds AP1 et AP3 tirent la même valeur de période de contention. Vu qu'ils sont en train d'émettre, ils ne voient pas qu'ils émettent en même temps. La croix grise indique que ces paquets sont perdus car ces nœuds ne détectent pas l'émission de l'autre. Ceci valide le mécanisme MR4. Cependant, le nœud AP2 reçoit correctement le paquet de AP1 car son niveau de réception est beaucoup plus fort que celui de AP3 (mécanisme MR3). Le nœud AP2 ayant reçu le paquet en double, il l'ignore en indiquant l'icône "Paquet reçu en double" comme défini dans le mécanisme MM5. Mais il ne peut pas émettre un accusé de réception car le nœud AP3 est en train d'émettre. Le nœud AP4 recevant le nœud AP3 plus fort que AP1, il émet un accusé de réception. Le nœud AP1 effectue une troisième tentative, et le nœud AP2 émet un accusé de réception puis ignore ce paquet qu'il a déjà reçu.

En même temps, le nœud AP4 ne détectant pas l'émission de AP1, il décide d'émettre. Le nœud AP3 capte à des niveaux de réception proches les nœuds AP1 et AP5, et ne peut donc pas comprendre le contenu des paquets. Il signale donc une collision, comme indiqué sur le mécanisme MR3.



FIGURE 3.73 – Echanges ICMP et UDP zoomée peu après la date 30000  $\mu\text{s}$ 

Afin de valider les priorités des classes de file d'attente, le paquet numéro 7 de type UDP et de classe de qualité de service *VO* (voix, classe la plus prioritaire) est mis en file d'attente à la date 29000  $\mu\text{s}$  sur la figure 3.73. Ce paquet a été acheminé jusqu'au nœud AP2, tout comme le paquet ICMP numéro 2. Ces deux paquets sont donc en concurrence, et c'est le paquet UDP de classe *VO* qui est émis en priorité. Le mécanisme MMI est donc validé.

Numéro	Source	Destination	Date d'émission	RTT
0	1	5	26000 $\mu\text{s}$	5068 $\mu\text{s}$
1	1	5	34000 $\mu\text{s}$	Perdu
2	1	5	39000 $\mu\text{s}$	16814 $\mu\text{s}$

TABLE 3.9 – RTT des requêtes ICMP

A la fin de cette simulation, le logiciel nous rapporte les RTT correspondants aux différentes requêtes ICMP. Ceux-ci sont présentés dans le tableau 3.9, où les requêtes 0 et 2 ont eu une réponse, alors que la requête 1 a eu un délai d'attente dépassé. Ce tableau valide le mécanisme MI3.

Cette simulation met en œuvre les mécanismes MR3, MR4, MM5, MI1, MI3 et MU1. Leurs fonctionnements correspondent à la norme 802.11 et aux spécifications initiales représentées sur notre modèle.

### 3.5.6 Utilisation du mécanisme de contrôles RTS/CTS

Cette simulation permet de valider les mécanismes liés au paquet de réservation de médium RTS et CTS. Seul le paramètre indiquant la taille à partir de laquelle ces mécanismes sont activés, est modifié sur tous les nœuds. Cette limite est fixée à 1000 octets.

#### 3.5.6.1 Configuration des paquets à transmettre lors de cette simulation

Pour cette simulation, deux requêtes ICMP concurrentes seront programmées. Ces deux requêtes auront 1500 octets de données. La première sera programmée du nœud 1 au nœud 8 à la date 100  $\mu$ s, et la seconde sera programmée du nœud 6 au nœud 1 à la date 1000  $\mu$ s. La classe de qualité de service de ces paquets sera *BE* (Best Effort).

Les modules *UDP* et *Beacon* seront désactivés.

#### 3.5.6.2 Analyse du graphe généré

Une nouvelle information est présente sur le graphique 3.74. Il s'agit de la réservation de médium. Celle-ci est représentée sur la partie haute de la ligne correspondante à un nœud. Un fond gris indique que le nœud a reçu un RTS ou un CTS et qu'il bloque son émission pour la durée indiquée dans le RTS ou dans le CTS. La partie basse indique que le nœud reçoit une transmission lorsqu'elle est grisée de la même manière que sur les graphiques précédents.

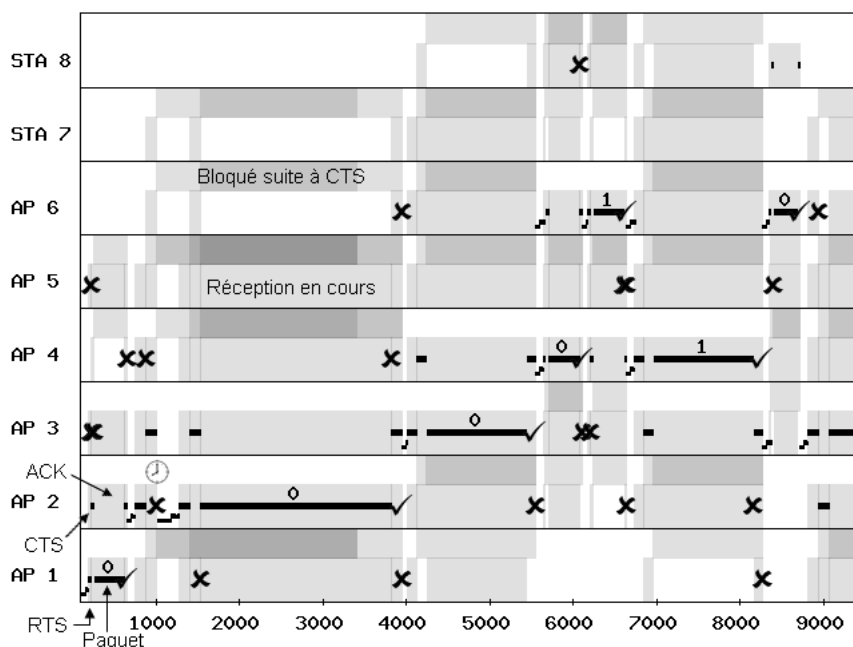


FIGURE 3.74 – Graphe généré par une simulation de RTS CTS

Nous pouvons constater sur le graphique 3.74 la présence de dialogue RTS CTS

avant l'émission de chaque paquet. Au début de la simulation, le point d'accès AP1 émet un RTS à AP2. Celui-ci répond un CTS, puis AP1 émet le paquet. Enfin, AP2 répond un *ACK*. Les mécanismes MM7 et MM8 sont donc validés.

A la date 1000, le nœud AP6 doit émettre une requête ICMP. Il ne reçoit pas de transmission, mais il a reçu un CTS. Il empêche alors l'émission. Ceci valide donc le mécanisme MM9.

Cette simulation permet de valider les derniers mécanismes présentés MM7, MM8 et MM9. Le mécanisme de réservation du médium implémenté correspond à la norme et à la description qui a été faite de celui-ci.

### 3.5.7 Simulation d'émission de Beacon

Lors de cette simulation, nous activons l'émission périodique de beacon par les points d'accès.

#### 3.5.7.1 Configuration des paquets à transmettre lors de cette simulation

Seul le module *Beacon* est activé et initialisé avec l'entier 4 dans la place *p\_count* du modèle *Beacon*. Celui-ci spécifie le nombre de paquets à émettre pour chaque point d'accès. Sur le terrain, un point d'accès émettrait ce type de paquet tant qu'il est allumé, mais nous limitons ce module avec une valeur faible afin de rendre plus lisible le résultat de cette simulation.

De même, les modules *UDP* et *ICMP* sont désactivés pour faciliter la lecture des résultats.

#### 3.5.7.2 Analyse du graphe généré

Sur le graphique de la figure 3.75, l'émission de paquet étant difficilement repérable, nous les avons indiqués par des flèches. En effet, la durée d'émission d'un beacon est beaucoup plus petite que la durée de la simulation.

Nous pouvons constater que les trois points d'accès n'émettent pas les paquets en même temps. Ceci provient de l'utilisation de la fonction *FIRSTBEACON()* qui tire le premier délai aléatoirement afin que les points d'accès n'émettent pas systématiquement les beacons en même temps.

Ensuite, à partir du premier, les beacons sont émis périodiquement par chaque nœud point d'accès, ce qui valide le fonctionnement du mécanisme MM10.

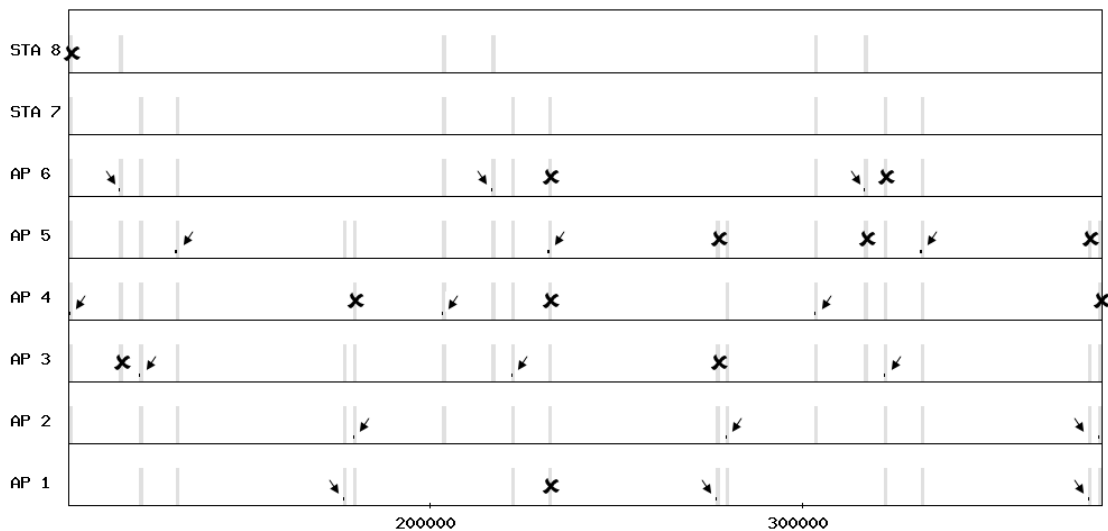


FIGURE 3.75 – Graphe généré lors d’une simulation de beacon

### 3.5.8 Conclusion

Cette validation permet d’assurer que le modèle correspond bien à la norme 802.11, avec une granularité connue pour chaque module. Ainsi, les simulations permettent de détailler les transmissions sur un réseau WiFi maillé et de mettre en avant des situations affectant la qualité de service. Des modifications pourront être apportées via certains leviers, et de nouvelles simulations permettront d’observer une amélioration ou une dégradation de la qualité de service.

De plus, l’évolutivité de ce modèle pourra permettre d’une part de modéliser plus en détail les parties où la granularité sera déterminée comme insuffisante, et d’autre part de modéliser d’autres leviers, dès lors qu’ils pourront être appliqués sur le matériel utilisé.

Le simulateur et ses modèles sous-jacents ayant été validés, il est désormais possible d’exploiter le simulateur, dans son domaine de validité, en ayant confiance dans la pertinence des résultats générés.



# Chapitre 4

## Exploitation du simulateur

### 4.1 Introduction

Cet outil nous permet de simuler des transmissions au travers d'un réseau maillé dans le but de trouver des actions permettant l'amélioration de la qualité de service.

#### 4.1.1 Synthèse des leviers disponibles

Ce simulateur permet de jouer sur les leviers suivants. Ceux-ci ont été choisis car ils peuvent être modifiés sur le matériel que nous utilisons. Les améliorations apportées par la simulation pourront ainsi être répercutées sur le terrain.

Voici les leviers configurables sur chaque nœud :

- *RTSTH* (RTS Threshold) : Ce seuil détermine à partir de quelle taille le mécanisme de réservation du médium RTS/CTS est activé.
- *BUF* : Ce paramètre indique le nombre de paquets pouvant être mis en file d'attente sur chaque nœud.

De même, ces leviers sont paramétrables pour chaque nœud, mais aussi pour chaque classe de qualité de service :

- *SRC[AC]* (Short Retry Count) : Cet entier correspond au nombre de réémissions en cas d'échec de transmission si RTS n'est pas activé.
- *LRC[AC]* (Long Retry Count) : Celui-ci indique le nombre de réémissions en cas d'échec si RTS est activé.
- *AIFS[AC]* : Ceci indique le nombre de durées élémentaires pendant lesquelles le canal doit être libre avant qu'un nœud puisse tirer une période de contention.
- *CW<sub>min</sub>[AC]* : Cet entier indique la borne supérieure initiale lors du tir de la période de contention.
- *CW<sub>max</sub>[AC]* : Celui-ci correspond à la borne supérieure maximale lors du tir de la période de contention.

Pour les nœuds de type point d'accès :

- activation et période d'émission de beacon peuvent être configurées.

Ainsi, les scénarii des simulations mettront en œuvre des actions séparées ou combinées sur ces leviers.

### 4.1.2 Rappel des problèmes évoqués dans l'état de l'art

Dans un premier temps, nous voulons faire apparaître sur la simulation les problèmes évoqués dans l'état de l'art. Nous privilégions une topologie en ligne plutôt qu'en étoile pour couvrir de grandes surfaces tout en augmentant le nombre de répéteurs impliqués dans la transmission. En effet, chaque nœud répète les paquets et le canal radio est de plus en plus utilisé. Ainsi, le débit utile diminue et des phénomènes de collision et de pertes de paquets apparaissent. De plus, le problème du nœud caché entraîne aussi des pertes lorsque certains nœuds ne sont pas à portée.

Cependant, une topologie en étoile peut révéler d'autres problèmes qui pourraient être mis en évidence avec ce simulateur.

La qualité de service est impactée par tous ces problèmes, et ce simulateur nous permet dans un premier temps de les mettre en avant, puis ensuite de tester des actions sur les leviers.

## 4.2 Simulation dans une configuration de référence

### 4.2.1 Présentation de la simulation

Nous avons utilisé la même topologie que celle qui nous a permis de décrire et valider le simulateur. Elle est présentée sur la figure 4.1 et correspond à un camping équipé où tous les emplacements sont couverts en WiFi. La configuration de la simulation utilisera donc des paramètres mesurés dans un cas réel. Celle-ci présente 8 nœuds entre lesquels les niveaux de réception sont différents, ce qui entraîne des débits physiques différents. Les nœuds 1 à 6 sont des points d'accès alors que 7 et 8 sont des stations. Les points d'accès 1, 2, 3, 4 et 6 sont alignés alors que le point d'accès 5 communique avec le nœud 3. La station 7 est connectée au point d'accès 5, et la station 8 est connectée au point d'accès 6.

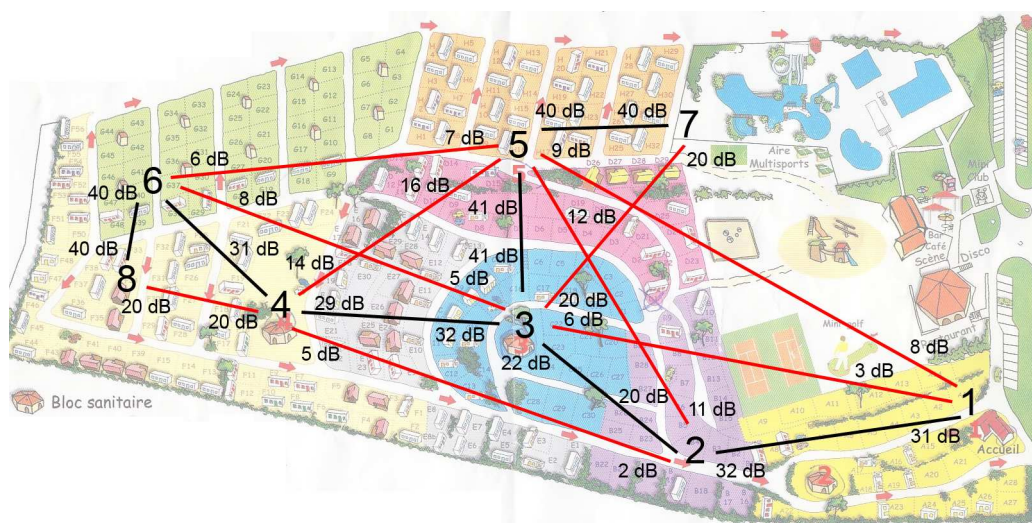


FIGURE 4.1 – Topologie de la simulation

L'altimétrie ainsi que les zones boisées ou non boisées ne sont pas représentées sur ce plan. Cependant, les rapports signal sur bruit et les débits physiques sont indiqués. C'est pourquoi le lien 2-3 est plus faible que le lien 1-2 alors qu'ils paraissent à la même distance.

Tous ces nœuds communiquent sur le même canal dans le but de faire apparaître les problèmes présentés précédemment. En effet, le nœud 1 ne capte pas le nœud 4. Les paquets émis par ceux-ci peuvent donc entrer en collision.

## 4.2.2 Résultats

Dans un premier temps, nous gardons les valeurs par défaut et nous injectons un flux classique dans le but d'observer la qualité de service.

Sur cette topologie est configurée un flux de 1000 paquets de 1500 octets allant du nœud 1 au nœud 4. Ce flux est de type UDP, et remplit le buffer d'émission du nœud 1. Lorsque ce nœud a l'opportunité d'émettre, les paquets sont relayés par les nœuds 2 et 3 jusqu'au nœud 4. Il n'y a pas d'accusé de réception au niveau Transport ou Application. Seul les accusés de réception de la couche Liaison seront émis comme le prévoit la norme 802.11.

### 4.2.2.1 Simulation d'un flux entre les nœuds 1 et 4

Nous pouvons donc tracer l'utilisation des files d'attente de chaque nœud, ainsi que les pertes de paquets dues aux files d'attente saturées.

Ainsi, les graphes 4.2 et 4.3 représentent le nombre de paquets présents dans les files d'attente d'émission de la couche liaison en fonction du temps, sur les nœuds 2 et 3 respectivement. La capacité de la file d'attente est de 100 paquets.

Le nombre de paquets dans le nœud 1 plafonne dès le début au maximum car la couche application remplit la file d'attente le plus possible. Les paquets qui ne sont pas dans cette file d'attente sont dans la mémoire de la couche UDP. Ils ne sont donc pas perdus.

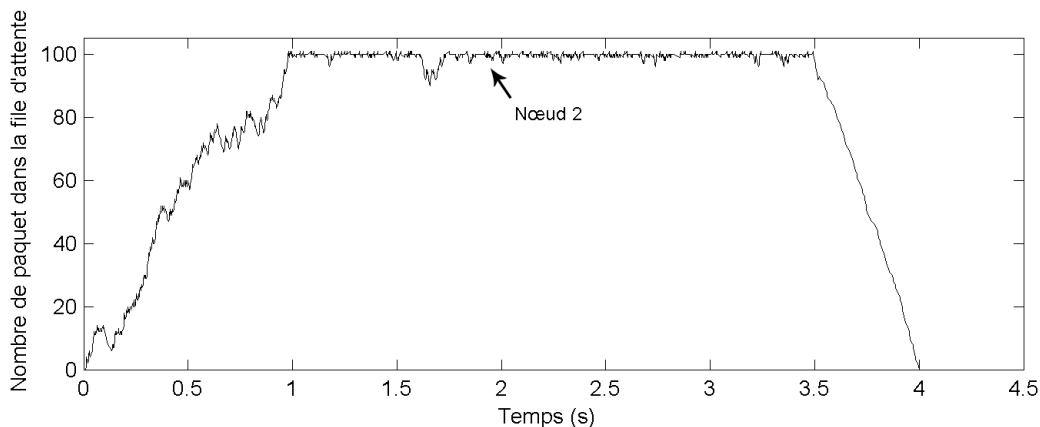


FIGURE 4.2 – Etat de la file d'attente du nœud 2 avec un flux 1-4



La courbe 4.2 correspond au nœud 2. Nous pouvons constater qu'elle croit jusqu'à atteindre le maximum. L'arrivée à ce seuil, qui correspond à la taille de la file d'attente, révèle des pertes de paquets par la couche LLC qui supprime les paquets quand la couche MAC ne peut pas les accueillir. Ceci est confirmé par le tableau 4.1 qui présente le nombre de paquets perdus à cause d'une file d'attente pleine.

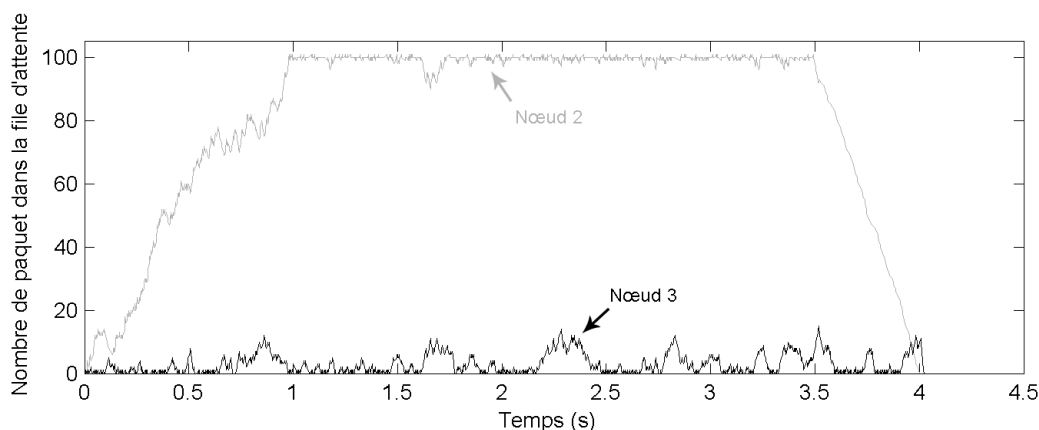


FIGURE 4.3 – Etat des files d'attente des nœuds 3 et 4 avec un flux 1-4

La courbe 4.3 représente le nombre de paquets dans le nœud 3 en noir, et rappelle la courbe de la file d'attente du nœud précédent en gris clair. Elle indique que les paquets sont acheminés au nœud suivant, sans attendre dans la file d'attente du nœud 3. Cela signifie que le nœud 3 retransmet les paquets à peu près au rythme où il les reçoit.

La file d'attente du nœud 4 n'est pas utilisée car ces paquets sont seulement reçus (non réémis) et les accusés de réception sont émis immédiatement sans passer par la file d'attente.

Ceci indique que le nœud 2, contrairement au nœud 3, reçoit plus de paquets que ce qu'il a l'opportunité d'émettre. En effet, le nœud 2 est un nœud intermédiaire qui ne génère pas de trafic, et qui est censé transférer tous les paquets qu'il reçoit. L'augmentation du nombre de paquets dans sa file d'attente révèle que le débit de paquets entrants est supérieur au débit de paquets sortants. Les pertes occasionnées par la saturation des files d'attente entraînent une dégradation de la qualité de service, ce qui n'est pas compatible avec des applications de voix sur ip.

Le tableau 4.1 nous montre les causes des pertes de paquets décrites ci-dessous :

- Pertes à cause de collision : Nombre de pertes de paquets à cause d'une collision
- Pertes à cause de récepteur en émission : Nombre de paquets perdus car le récepteur est en cours d'émission
- Pertes à cause d'erreur radio : Paquet volontairement altéré pour simuler des problèmes de transmission radio
- Pertes à cause de buffer plein : Paquet reçu et acquitté mais perdu car le récepteur ne peut pas placer le paquet dans la file d'attente d'émission parce qu'elle est pleine.

Rappelons que lors d'une perte dans la couche MAC (collision, récepteur en émission ou erreur radio), le paquet est réémis ; alors que lorsqu'il est perdu dans la couche LLC

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	636	293	1200	306
2-1	0	0	471	0
2-3	0	369	456	0
3-2	0	0	762	0
3-4	288	358	1022	0
4-3	348	0	459	0
Total	1272	1020	4370	306

TABLE 4.1 – Nombre et cause des pertes sur chaque lien pour la simulation 1-4

à cause d'un manque de place dans la file d'attente, il est immédiatement perdu.

Le flux injecté allant du nœud 1 au nœud 4, les liens 2-1, 3-2 et 4-3 ne transportent que les accusés de réception, c'est pourquoi les nombres de pertes comptabilisées sont plus faibles que sur les autres liens.

Suite à un problème de transmission de type collision, paquet perdu ou erreur radio, la couche MAC 802.11 prévoit des réémissions, alors qu'en cas de pertes dues à un dépassement de la capacité de la file d'attente, le paquet est perdu et aucune réémission n'est prévue.

Parmi les traces générées par le simulateur, le compteur de paquets reçus au niveau UDP indique que 307 paquets sur 1000, soit 31 %, n'ont pas été reçus. De plus, le débit au niveau de la couche UDP est de  $2067 \text{ Kbit.s}^{-1}$ , le délai moyen est de 838 ms avec un écart type de 242 ms. L'écart type correspond la gigue dans le cas d'une transmission de voix sur IP. Cette valeur reflète la qualité audio.

Ce fort taux de pertes est principalement dû à la saturation des files d'attentes. En effet, seulement 1 paquet a été perdu à cause de collisions, receveurs en émission et erreurs radio.

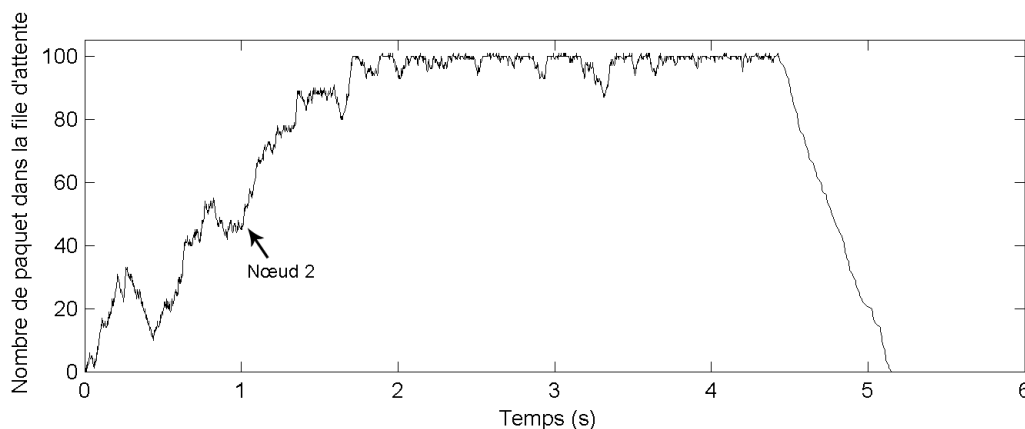


FIGURE 4.4 – Etat de file d'attente du nœud 2 avec un flux 1-6

#### 4.2.2.2 Simulation d'un flux entre les nœuds 1 et 6

Une seconde simulation est réalisée avec le nœud 6 comme destinataire. L'état des files d'attente est présenté sur les graphes 4.4, 4.5 et 4.6. Les autres paramètres de la simulation restent identiques à la simulation précédente.

De même que précédemment, le nombre de paquets dans la file d'attente du nœud 1 reste au maximum tant que des paquets restent à émettre dans la couche UDP.

La courbe de la figure 4.4, correspondant au nœud 2, affiche aussi un remplissage de la file d'attente jusqu'à des pertes de paquets dues à la taille limitée de la mémoire. Cependant, cette progression est plus lente que précédemment.

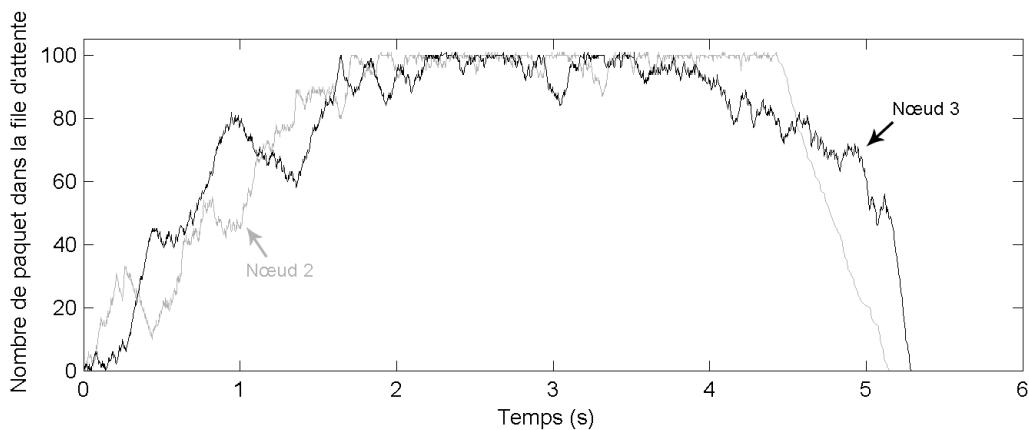


FIGURE 4.5 – Etat des files d'attente des nœuds 2 et 3 avec un flux 1-6

La courbe noire de la figure 4.5, correspondant au nœud 3, présente aussi un remplissage des files d'attente, contrairement à la simulation précédente. Cette figure rappelle en gris la courbe représentant le remplissage du nœud 2.

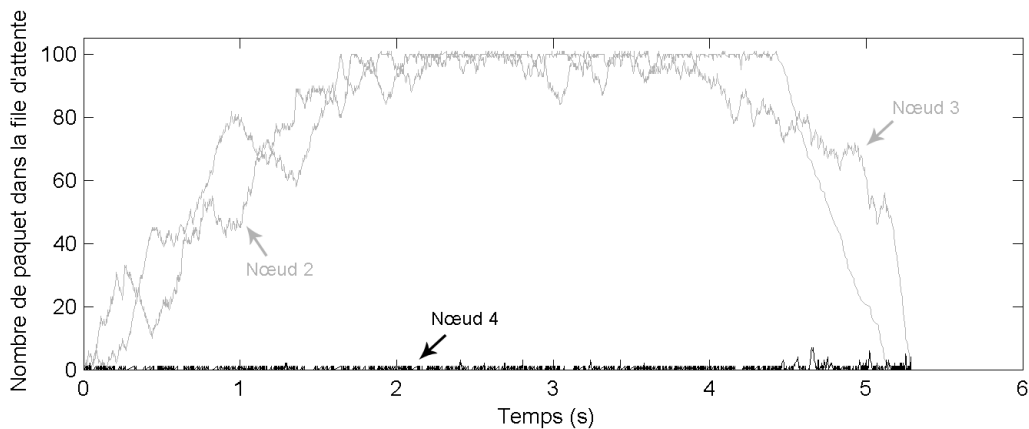


FIGURE 4.6 – Etat des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6

Enfin, l'état de la file d'attente du nœud 4, présent sur la courbe noire de la figure

4.6, retransmet ses paquets au rythme où il les reçoit vers le nœud 6 et l'état de sa file d'attente reste peu rempli. Les courbes grises rappellent l'état des files d'attente des nœud 2 et 3.

Le tableau 4.2 présente le nombre et les raisons des pertes de paquet.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	1378	299	1540	166
2-1	4	1	437	0
2-3	172	469	604	31
3-2	0	0	917	0
3-4	518	524	1345	0
4-3	317	0	557	0
4-6	1041	145	631	0
6-4	494	0	592	0
Total	3924	1438	6623	197

TABLE 4.2 – Nombre et cause des pertes sur chaque lien pour la simulation 1-6

Dans cette simulation, les traces indiquent que 201 paquets sur 1000 (20 %) n'ont pas été reçus au niveau de la couche UDP du destinataire. L'ajout d'un nœud supplémentaire dans la transmission a diminué le taux de pertes de paquets dû à la saturation des files d'attente. De plus, le débit UDP est de  $1813 \text{ Kbit.s}^{-1}$ . Le délai moyen est de 1399 ms et l'écart type de 435 ms. Le débit et le délai sont moins bon que lors de la simulation précédente, et peut être expliqué d'une part par l'ajout d'une étape à la transmission, ce qui allonge la durée du transfert, mais aussi parce que 10 % de paquets supplémentaires ont réussi à arriver à destination.

#### 4.2.2.3 Discussion sur les différences constatées entre les flux 1-4 et 1-6

Dans le cas d'une transmission du nœud 1 au nœud 4, cette accumulation de paquet peut être expliquée par les débits physiques entre les nœuds. En effet, le lien 1-2 est à  $54 \text{ Mbit.s}^{-1}$ , alors que le lien 2-3 n'est qu'à  $24 \text{ Mbit.s}^{-1}$ . Ainsi, au niveau du nœud 2, les paquets mettent donc plus de temps à être émis qu'à être reçu.

De même dans le cas d'une transmission du nœud 1 au nœud 6, les débits physiques entre les nœuds peuvent expliquer cette accumulation de paquets, mais un autre phénomène est présent. Le nœud 1 ne reçoit que les nœuds 2 et 3, alors que le nœud 2 reçoit les nœuds 1, 3, 4. Grossièrement, le nœud 1 a donc une probabilité de  $\frac{1}{3}$  de gagner la période de contention alors que le nœud 2 n'a qu'une probabilité de  $\frac{1}{4}$ .

Sur cette topologie, les simulations ont permis de constater que la saturation d'une file d'attente est une cause importante de pertes. Nous pouvons donc chercher si une ou plusieurs actions sur les leviers permettent de limiter l'impact de la saturation des files d'attentes. Les actions que nous allons mener sont orientées vers un ralentissement de l'injection de flux à la source.

### 4.3 Ralentissement de l'injection de flux au niveau de la source du trafic

Afin d'éviter aux files d'attente des nœuds intermédiaires d'être saturées, et ainsi de perdre des paquets, une piste d'amélioration serait de ralentir l'arrivée des paquets injectés par la source, et de laisser une opportunité de parole plus importante aux nœuds relais.

Nous pouvons commencer par étudier en détail l'influence de la plage de tir de la période de contention pour comprendre les différentes possibilités sur les paramètres  $IFS$ ,  $CW_{min}$  et  $CW_{max}$ .

#### 4.3.1 Rappel du rôle des paramètres $AIFS$ , $CW_{min}$ et $CW_{max}$

Nous considérons que les seules erreurs de transmission sont dues aux collisions uniquement. Ainsi, nous ne tiendrons pas compte des erreurs de transmission radio dans cette partie.

Supposons qu'une transmission correcte soit vue sur le médium par tous les nœuds, et que ceux-ci veulent émettre. Dès la fin de la transmission, ils comptent alors à partir du même moment le délai intertrame  $AIFS$ . Ensuite, ils tirent un entier compris entre  $[0; CW]$ . Cet entier est le nombre de durées élémentaires  $aSlotTime$  à attendre.

Pour un nœud, nous avons 4 possibilités :

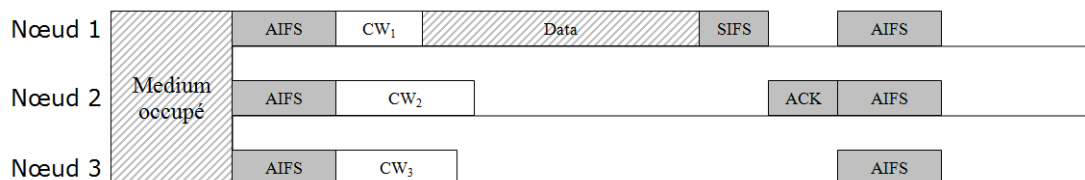


FIGURE 4.7 – Gain ou perte de période de contention

- Si le nœud tire une valeur strictement inférieure à celles tirées par les autres nœuds, il gagne la période de contention, et vu qu'il est le seul, le paquet est émis correctement. Par exemple, le nœud 1 de la figure 4.7 émet son paquet avec succès.
- Si le nœud a perdu car il n'a pas la valeur minimum et qu'il n'y a pas de collision, il ne va pas émettre. C'est le cas du nœud 3 sur la figure 4.7.

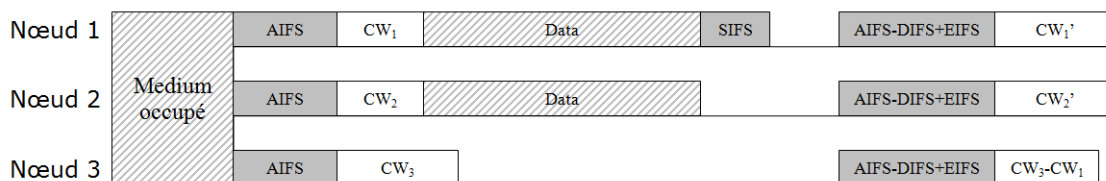


FIGURE 4.8 – Collision due au tir de période de contention

- Si le nœud tire le minimum, mais que d'autres nœuds l'ont tiré aussi, il va donc émettre en même temps que les autres nœuds ayant tirés la même valeur. Les émissions rentrent donc en collision et ce n'est que l'absence d'accusé de réception qui va indiquer cette collision. Le prochain délai intertrame est  $AIFS - DIFS + EIFS$ . Par exemple, sur la figure 4.8, le nœud 1 produit une collision avec le nœud 2.
- Si plusieurs autres nœuds tirent le minimum. Le nœud a perdu et ne va pas émettre, mais les autres nœuds rentrent en collision. Celle-ci est détectée, et le prochain délai intertrame est  $AIFS - DIFS + EIFS$ , comme par exemple le nœud 3 dans 4.8.

### 4.3.2 Calcul de probabilité des gains de période de contention

Nous avons mis en place un outil testant toutes les possibilités et comptant les gains de période de contention et les collisions avec les conditions spécifiées dans le paragraphe précédent. Les paramètres *AIFS* et *CW* sont spécifiés en entrée. Avec ceci, l'outil retourne les probabilités de gain, de perte et d'être un des acteurs d'une collision pour chaque nœud. De plus, il retourne la probabilité de collision par rapport à toutes les possibilités.

Le tableau 4.3 nous présente un exemple de calcul de probabilité de 3 nœuds avec le paramètre *AIFS* à 2, et *CW* à 3. Ce sont les valeurs par défaut de paquets de type *BE* à partir d'un point d'accès, qui ont été utilisées dans la simulation du cas réel précédent. Par rapport aux simulations précédentes, ceci est le cas au niveau du nœud 1 alors que les nœuds 1, 2 et 3 ont des paquets à émettre. Dans ce cas, lors de la première tentative, la probabilité de collision sera de 34%. Pour le nœud 1, il a une probabilité de 21% d'émettre correctement, 25% d'être impliqué dans la collision, et 53% de perdre la période de contention.

Nœud	<i>AIFS</i>	<i>CW</i>	Gain	Collision	Perdu
1	2	3	21 %	25 %	53 %
2	2	3	21 %	25 %	53 %
3	2	3	21 %	25 %	53 %
Collision 34 %					

TABLE 4.3 – Gain de période de contention de 3 nœuds avec la même configuration

En se plaçant au niveau du nœud 3, le canal radio est partagé entre les 5 nœuds 1, 2, 3, 4 et 6. Si ces 5 nœuds, ayant ces mêmes paramètres, ont des paquets à émettre, le tableau 4.4 montre que la probabilité qu'une collision se produise est de 52 %. De plus, la probabilité d'émettre correctement lors de la première tentative est de 9%.

Afin de ralentir l'injection de flux au niveau du nœud 1, nous pouvons augmenter son temps intertrame *AIFS* de 2 à 3. Le tableau 4.5 indique que les nœuds 2 et 3 ont une probabilité plus importante de réussir une émission alors que la probabilité de gain du nœud 1 passe donc de 21 % à 7 %. Donc, cette action sur les leviers permettent de favoriser le relais de paquets, ce qui devrait limiter dès lors le phénomène de saturation dans les files d'attente des nœuds concernés.

Nœud	<i>AIFS</i>	<i>CW</i>	Gain	Col	Perdu
1	2	3	9 %	25 %	65 %
2	2	3	9 %	25 %	65 %
3	2	3	9 %	25 %	65 %
4	2	3	9 %	25 %	65 %
5	2	3	9 %	25 %	65 %
Collision 52 %					

TABLE 4.4 – Gain de période de contention de 5 nœuds avec la même configuration

Nœud	<i>AIFS</i>	<i>CW</i>	Gain	Col	Perdu
1	3	3	7 %	14 %	78 %
2	2	3	31 %	25 %	43 %
3	2	3	31 %	25 %	43 %
Collision 29 %					

TABLE 4.5 – Gain de période de contention de 3 nœuds avec modification de *AIFS*

Un autre levier est le paramètre  $CW_{min}$ . Dans le tableau 4.6, ce paramètre qui avait la valeur 3 est modifié à la valeur 4 pour le nœud 1. De même que l'action sur le levier précédent, mais de manière moins importante, le nœud 1 a une priorité plus faible que les autres.

Nœud	<i>AIFS</i>	<i>CW</i>	Gain	Col	Perdu
1	2	4	17 %	20 %	62 %
2	2	3	25 %	25 %	50 %
3	2	3	25 %	25 %	50 %
Collision 32 %					

TABLE 4.6 – Gain de période de contention de 3 nœuds avec modification de  $CW_{min}$ 

Ces deux leviers diminuent la probabilité de gain de la période de contention du nœud 1, et donc les chances d'émettre un paquet si les autres nœuds veulent aussi émettre. Le levier *AIFS*, imposant un délai supplémentaire, est plus efficace que le levier  $CW_{min}$  qui agrandit la plage de tir de la période de contention. Cependant, bien qu'un paramètre soit plus efficace que l'autre, nous pouvons mesurer les influences respectives de ces deux paramètres via le simulateur afin de choisir celle qui est le plus appropriée.

### 4.3.3 Application de ces paramètres sur le simulateur

Après avoir observé que la probabilité de gain de la période de contention diminue en augmentant les paramètres *AIFS* ou  $CW_{min}$ , nous pouvons appliquer une action sur ces leviers dans le simulateur afin d'observer si l'occupation des files d'attente des deux nœuds intermédiaire diminue.

Dans un premier temps, le paramètre *AIFS* du nœud 1 est modifié seulement. En-

suite, le paramètre  $CW_{min}$  du nœud 1 sera modifié.

Pour chaque cas, deux simulations sont effectuées. Le flux de la première est destiné au 4, alors que celui de la seconde est destiné au nœud 6. Ainsi, les résultats seront comparables aux simulations réalisées dans la section précédente.

#### 4.3.3.1 Augmentation du paramètre *AIFS* sur le nœud 1 pour un flux du nœud 1 à 4

Dans cette simulation, le paramètre *AIFS* pour la classe de qualité de service BE du nœud 1 est modifié de la valeur 2 à la valeur 3. Cette classe de qualité de service correspond à celle au flux injecté. De plus, le flux est programmé du nœud 1 vers le nœud 4.

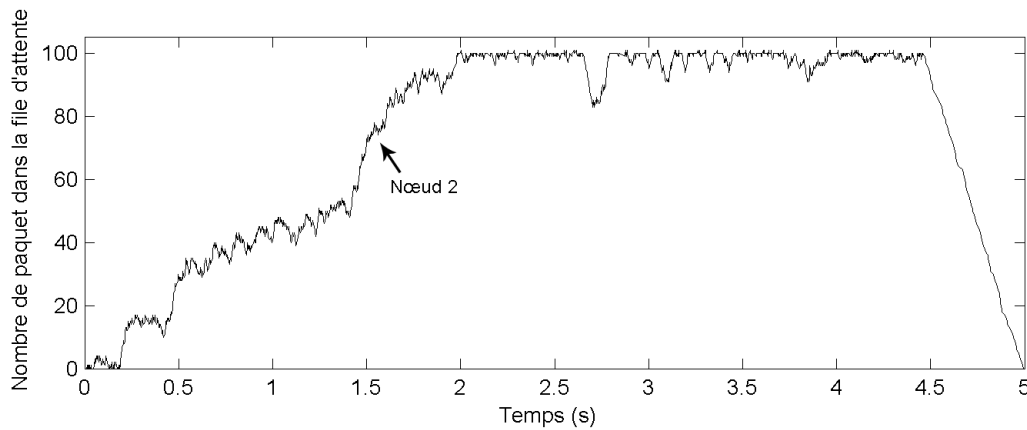


FIGURE 4.9 – Evolution de la file d’attente du nœud 2 avec un flux 1-4 et modification du paramètre *AIFS* sur le nœud source

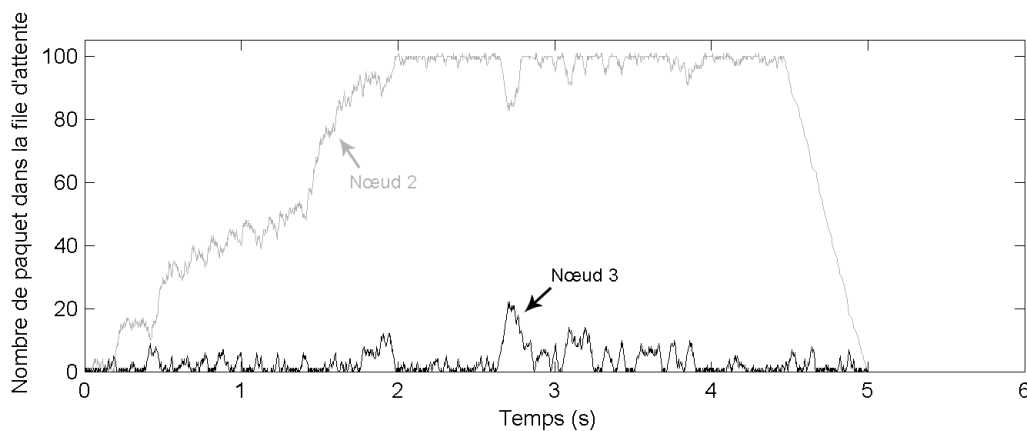


FIGURE 4.10 – Evolution des files d’attente des nœuds 2 et 3 avec un flux 1-4 et modification du paramètre *AIFS* sur le nœud source



La figure 4.9 présente l'évolution de la file d'attente du nœud 2 lors de la transmission. Nous pouvons constater que la saturation de la file d'attente a lieu au bout de 2 secondes, alors qu'elle avait déjà eu lieu au bout de 1 seconde lors de la simulation avec les paramètres par défaut présentée sur la figure 4.2. Le flux du nœud 1 est donc effectivement ralenti, mais le phénomène de saturation persiste.

La file d'attente du nœud 3, visible sur la courbe noire de la figure 4.10, est quasiment dans le même état qu'avec les valeurs par défaut présenté sur la figure 4.3. En effet, elle reste peu chargée. Le nœud transmet donc les paquets reçus au rythme où il les reçoit.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	576	194	1123	142
2-1	0	0	485	0
2-3	0	376	552	0
3-2	0	0	944	0
3-4	180	354	1228	0
4-3	396	0	597	0
Total	1152	924	4929	142

TABLE 4.7 – Nombre et cause des pertes sur chaque lien pour la simulation avec un flux 1-4 et modification du paramètre *AIFS* sur le nœud source

La colonne indiquant le nombre de pertes à cause de Buffer plein du tableau 4.7 confirme que le nombre de paquets perdus à cause d'une saturation de buffer a diminué par rapport au résultat précédent présent sur le tableau 4.1.

Le compteur de paquets UDP reçus indique que 142 paquets sur 1000 (14 %) ont été perdus. Ces pertes sont toutes dues à une saturation des files d'attente. Le nombre de perte est donc divisé par deux par rapport à la simulation ayant les leviers configurés avec les valeurs par défaut. Le débit UDP est de  $2058 \text{ Kbit.s}^{-1}$  alors qu'il était à  $2067 \text{ Kbit.s}^{-1}$  avec les valeurs par défaut. L'impact est donc très faible, voire négligeable sur le débit applicatif mais le nombre de pertes a diminué. Le délai moyen est de 878 ms avec un écart type de 262 ms, ce qui est sensiblement plus élevé qu'avec les valeurs par défaut.

#### 4.3.3.2 Augmentation du paramètre *AIFS* sur le nœud 1 pour un flux du nœud 1 à 6

Cette même modification du levier *AIFS* est appliquée sur un flux transitant du nœud 1 au nœud 6.

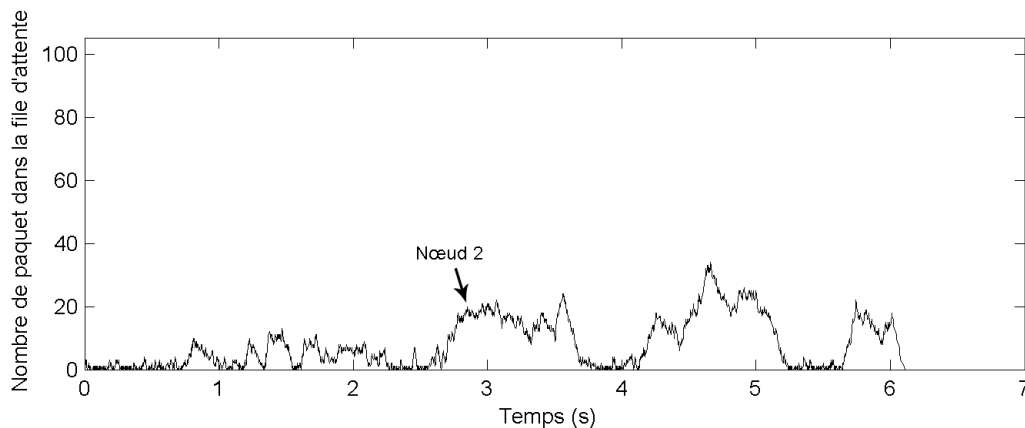


FIGURE 4.11 – Evolution de la file d'attente du nœud 2 avec un flux 1-6 et modification de *AIFS*

L'état de la file d'attente du nœud 2 est présenté sur la figure 4.11. Dans ce cas, elle reste faible avec un remplissage allant jusqu'à 20 paquets. La taille de la file n'est donc pas atteinte, et aucune perte de paquet n'est due à un dépassement de la capacité du buffer sur ce nœud.

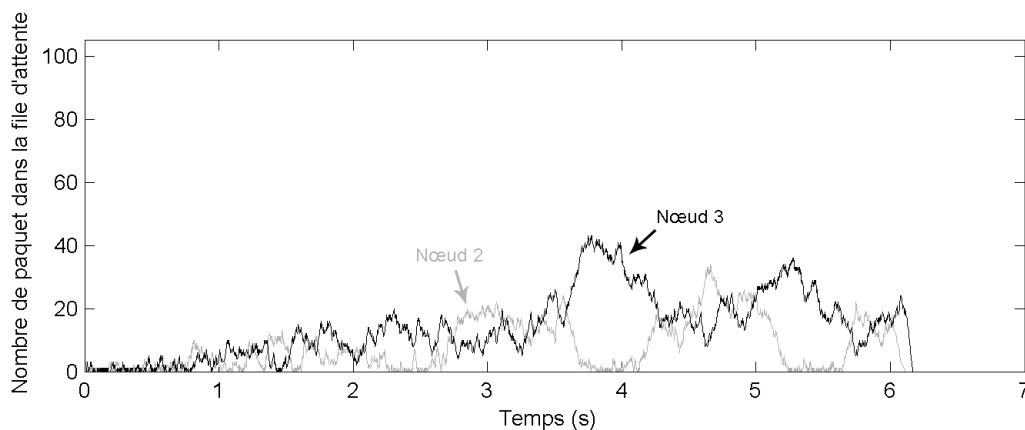


FIGURE 4.12 – Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 et modification du paramètre *AIFS* sur le nœud source

De même, la courbe noire de la figure 4.12 indiquant l'état de la file d'attente du nœud 3, montre que celle-ci ne se remplit pas.

Enfin, de même qu'avec le paramétrage de la simulation par défaut présenté sur la figure 4.6, la file d'attente du nœud 4, présentée sur la courbe noire de la figure 4.13, reste faible.

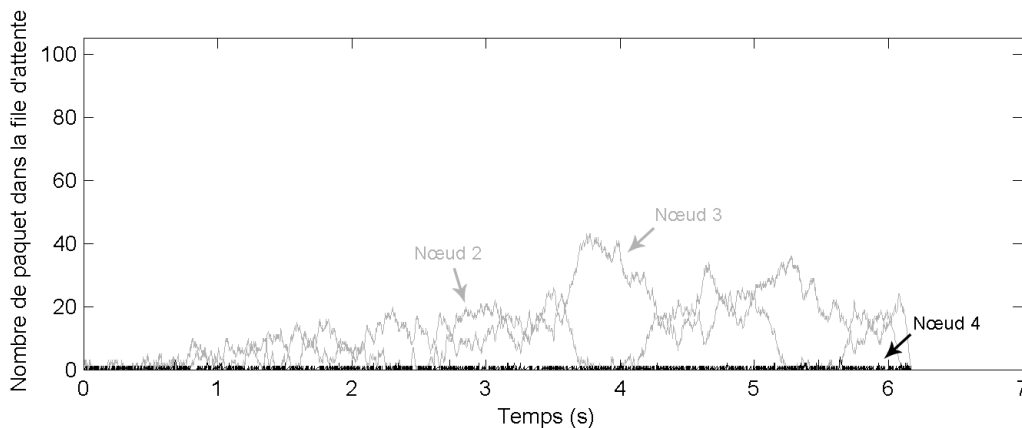


FIGURE 4.13 – Evolution des files d’attente des nœuds 2, 3 et 4 avec un flux 1-6 et modification du paramètre *AIFS* sur le nœud source

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	1724	216	1602	0
2-1	8	2	452	0
2-3	212	453	681	0
3-2	0	0	1101	0
3-4	438	499	1482	0
4-3	374	0	637	0
4-6	1426	156	740	0
6-4	754	0	766	0
Total	4936	1326	7461	0

TABLE 4.8 – Nombre et cause des pertes sur chaque lien pour la simulation avec un flux 1-6 et modification du paramètre *AIFS* sur le nœud source

Le tableau 4.8 indique bien qu’aucune perte de paquet n’est due à une saturation de file d’attente.

Dans cette simulation, seulement 5 paquets sur 1000 (5 %) n’ont pas été acheminés à la couche UDP du destinataire. Ces pertes ne sont pas dues à une saturation de files d’attente, mais à des problèmes radios. Le débit mesuré au niveau de la couche UDP est de  $1935 \text{ Kbit.s}^{-1}$ , ce qui est mieux que lors de la simulation ayant les leviers aux valeurs par défaut qui était de  $1813 \text{ Kbit.s}^{-1}$ .

Le délai moyen est de 735 ms avec un écart type de 175 ms. Ces résultats sont nettement meilleurs à ceux mesurés avec les valeurs par défaut (délai moyen de 1399 ms avec un écart type de 435 ms).

L’action sur ce levier est donc très positive dans ce cas, car le taux de perte est divisé par 4 sans impact conséquent sur le débit applicatif.

### 4.3.3.3 Augmentation du paramètre $CW_{min}$ sur le nœud 1 pour un flux du nœud 1 à 4

Dans cette simulation, nous modifions le paramètre  $CW_{min}$  pour la classe de qualité de service BE du nœud 1. Sa valeur est spécifiée à 4 alors qu'elle était par défaut à 3. Le paramètre  $AIFS$  est à nouveau configuré à sa valeur par défaut.

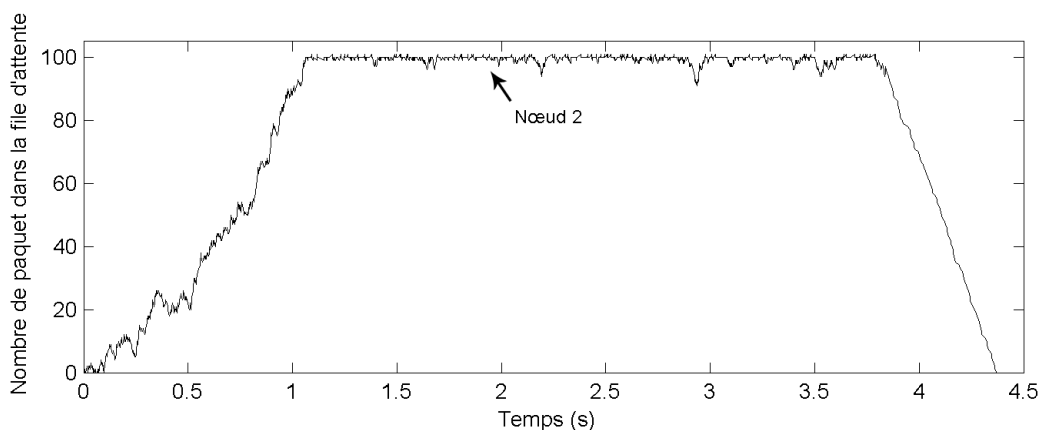


FIGURE 4.14 – Evolution de la file d'attente du nœud 2 avec un flux 1-4 et modification du paramètre  $CW_{min}$  sur le nœud source

La figure 4.14 présente l'état de la file d'attente du nœud 2. Dans ce cas, elle arrive à saturation au bout d'une seconde, ce qui entraîne des pertes de paquets.

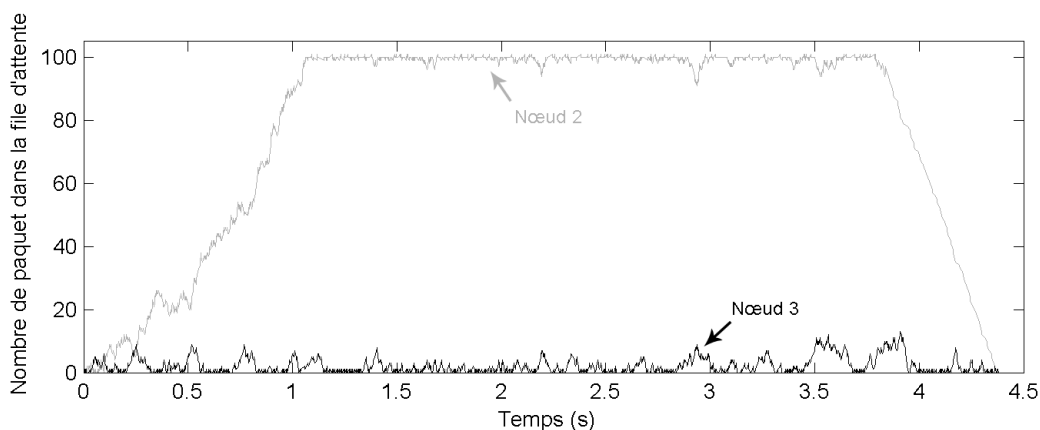


FIGURE 4.15 – Evolution des files d'attente des nœud 2 et 3 avec un flux 1-4 et modification du paramètre  $CW_{min}$  sur le nœud source

La file d'attente de la borne 3 reste peu remplie, comme nous pouvons le constater sur la courbe noire de la figure 4.15.

Le tableau 4.9 montre que l'action sur ce levier provoque moins de pertes dues à une file d'attente pleine qu'avec la valeur par défaut présenté sur le tableau 4.1.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	621	269	1207	249
2-1	0	0	488	0
2-3	0	376	528	0
3-2	0	0	816	0
3-4	282	374	1089	0
4-3	339	0	509	0
Total	1242	1019	4637	249

TABLE 4.9 – Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec modification du paramètre  $CW_{min}$  sur le nœud source

Sur cette simulation, 251 paquets sur 1000 (25 %) ont n'ont pas été reçu au niveau de la couche UDP. Nous pouvons constater une sensible amélioration par rapport à la simulation ayant les leviers aux valeurs par défaut, pour laquelle le taux de perte était de 31 %. De plus, le débit est de  $2052 \text{ Kbit.s}^{-1}$ , ce qui est proche du débit de la simulation initiale ( $2067 \text{ Kbit.s}^{-1}$ ). Enfin, le délai moyen est de 848 ms avec un écart type de 258 ms alors que nous avons un délai moyen de 838 ms avec un écart type de 242 ms. L'amélioration ne réside donc que dans la diminution des pertes de paquets.

#### 4.3.3.4 Augmentation du paramètre $CW_{min}$ sur le nœud 1 pour un flux du nœud 1 à 6

En conservant cette modification du levier  $CW_{min}$ , le flux est maintenant configuré pour aller du nœud 1 au nœud 6.

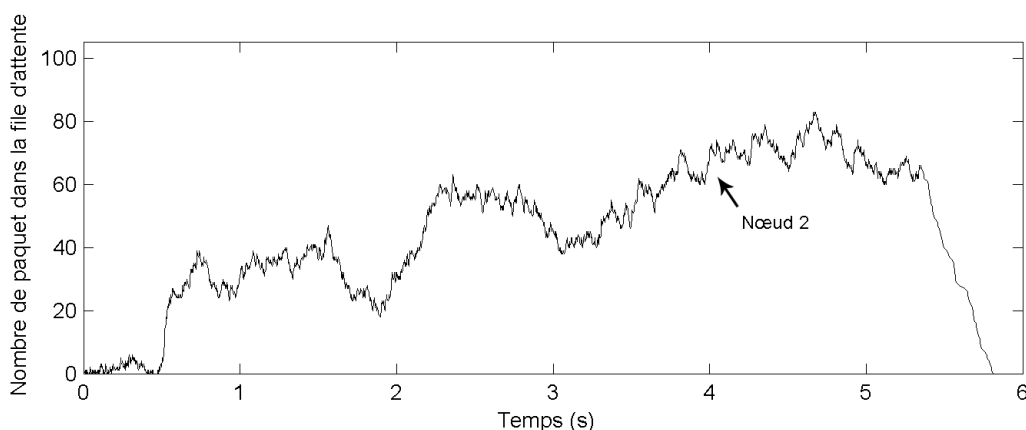


FIGURE 4.16 – Evolution de la file d'attente du nœud 2 avec un flux 1-6 et modification du paramètre  $CW_{min}$  sur le nœud source

La file d'attente du nœud 2, présentée sur la figure 4.16, semble augmenter très progressivement, mais n'atteint pas le plafond car tous les paquets ont été émis. Aucun

n'est perdu à cause d'un dépassement de capacité de cette file.

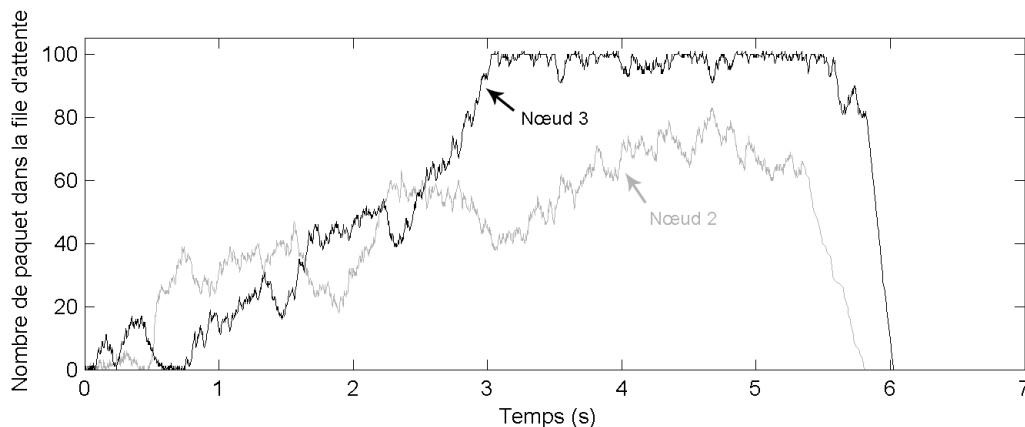


FIGURE 4.17 – Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 et modification du paramètre  $CW_{min}$  sur le nœud source

Cependant, la courbe noire de la figure 4.17 montre que la file d'attente du nœud 3 est saturée au bout de 3 secondes. Ceci arrive plus tard que lors de la simulation avec les paramètres par défaut présenté sur la figure 4.5, mais des pertes persistent.

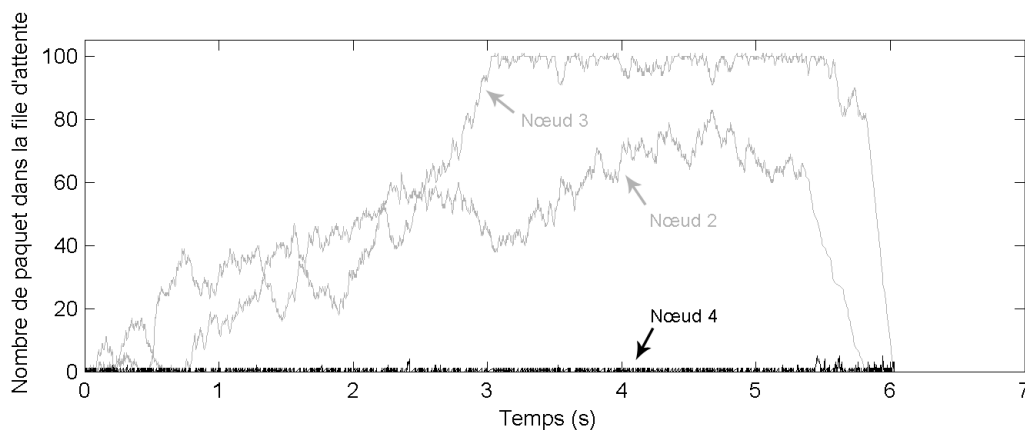


FIGURE 4.18 – Evolution des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6 et modification du paramètre  $CW_{min}$  sur le nœud source

Enfin, la file d'attente du nœud 4, présentée par la courbe noire de la figure 4.18, reste à un niveau très faible indiquant qu'elle émet les paquets au rythme où elle les reçoit.

Le tableau 4.10 confirme que moins de paquets sont perdus à cause de files d'attente pleines malgré une saturation sur le nœud 3.

Le nombre de paquets perdus au niveau de la couche UDP est de 93 sur 1000, soit 9 %. Ces pertes sont dues en majorité à un dépassement de file d'attente au niveau d'un

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	1508	300	1713	0
2-1	4	1	440	0
2-3	200	492	736	86
3-2	0	0	1077	0
3-4	508	528	1441	0
4-3	383	0	626	0
4-6	1197	158	700	0
6-4	554	0	759	0
Total	4354	1479	7492	86

TABLE 4.10 – Nombre et cause des pertes sur chaque lien pour la simulation 1-6 avec modification du paramètre  $CW_{min}$  sur le nœud source

nœud intermédiaire (nœud 3), mais aussi 7 paquets ont été perdus à cause de problèmes radio. Ceci est mieux qu'avec les valeurs par défaut configurées sur les leviers (20 %), mais moins performant que l'action sur le levier *AIFS* (1 %). De plus, le débit mesuré au niveau de la couche UDP est de  $1805 \text{ Kbit.s}^{-1}$ , ce qui est sensiblement identique au débit mesuré avec les valeurs par défaut, soit  $1813 \text{ Kbit.s}^{-1}$ .

Le délai moyen est de 1198 ms avec un écart type de 430 ms. Ceci est aussi meilleur qu'avec les valeurs par défaut (1399 ms de moyenne et un écart type de 435 ms), et aussi moins bon que l'action sur *AIFS*.

#### 4.3.4 Bilan des actions sur l'injection de trafic

A partir de ces simulations, nous pouvons comparer les résultats afin de tirer des conclusions. Dans un premier temps, nous pouvons comparer les résultats des simulations entre les nœuds 1 et 4.

	Nœud 1		Pertes buffer nœud			UDP		Délai UDP	
	<i>AIFS</i>	$CW_{min}$	2	3	Total	Pertes	Débit $\text{Kbit.s}^{-1}$	Moyen	Ecart type
Défaut	2	3	306	0	306	31 %	2067	838 ms	242 ms
<i>AIFS</i>	3	3	142	0	142	14 %	2058	878 ms	262 ms
$CW_{min}$	2	4	249	0	249	25 %	2052	848 ms	258 ms

TABLE 4.11 – Récapitulatif des simulations 1-4

Le tableau 4.11 indique que les pertes de paquets au niveau des files d'attente n'ont lieu qu'au niveau du nœud 2. De plus, la file d'attente du nœud 3 n'arrive pas à saturation dans ces simulations.

Nous pouvons constater que les pertes de paquets au niveau UDP sont dues pour la majorité au dépassement de capacité des files d'attente.

Le levier  $CW_{min}$  améliore de 6 % ces pertes de paquets, et le levier  $AIFS$  l'améliore de 16 %. Cependant, le taux de perte est encore élevé, ce qui n'améliore pas suffisamment la qualité de service sur cette topologie.

	Nœud 1		Pertes buffer nœud				UDP		Délai UDP	
	$AIFS$	$CW_{min}$	2	3	4	Total	Pertes	Débit $Kbit.s^{-1}$	Moyen	Ecart type
Défaut	2	3	166	31	0	197	20 %	1813	1399 ms	435 ms
$AIFS$	3	3	0	0	0	0	1 %	1935	735 ms	175 ms
$CW_{min}$	2	4	0	86	0	86	9 %	1805	1198 ms	430 ms

TABLE 4.12 – Récapitulatif des simulations 1-6

Le tableau 4.12 résume les résultats des simulations pour lesquelles les flux sont configurés du nœud 1 au nœud 6. Il indique de même que le dernier nœud relais réémet les paquets au rythme où il les reçoit. De plus, les pertes de paquets au niveau UDP correspondent aux pertes dues au dépassement de file d'attente sur les autres nœuds.

Dans ce cas, le levier  $CW_{min}$  réduit de 11 % le taux de pertes, et le levier  $AIFS$  élimine presque totalement les pertes au niveau de la couche UDP (seulement moins de 1 %). Grâce aux actions sur ces leviers, la qualité de service est donc améliorée car le nombre de pertes, le délai et l'écart type (reflétant la gigue) sont plus faibles.

L'action menée sur le paramètre  $AIFS$  du premier nœud n'évite la saturation des files d'attente que dans le cas d'un flux 1-6. En effet, l'action sur le flux 1-4 est bénéfique mais n'est pas suffisante. Notre étude doit donc être menée sur l'intégralité du chemin, à savoir les nœuds relais entre source et destination. Dans ce but, nous allons étudier l'impact d'une action portant sur le nœud injecteur et sur le premier relais.

#### 4.4 Ralentissement de l'injection du flux au niveau de la source et du premier relais

Nous allons maintenant configurer  $AIFS$  avec la valeur 4 dans le nœud 1, et la valeur 3 dans le nœud 2. Les autres nœuds gardent la valeur par défaut 2. Le paramètre  $CW_{min}$  étant peu influant, garde la valeur par défaut 3. Afin de pouvoir comparer l'impact de cette action, les paramétrages des flux et de la topologie restent identiques aux simulations précédentes.

Le flux est configuré dans un premier temps du nœud 1 à 4, puis du nœud 1 à 6.



#### 4.4.1 Augmentation des paramètres *AIFS* sur les nœud 1 et 2 pour un flux du nœud 1 à 4

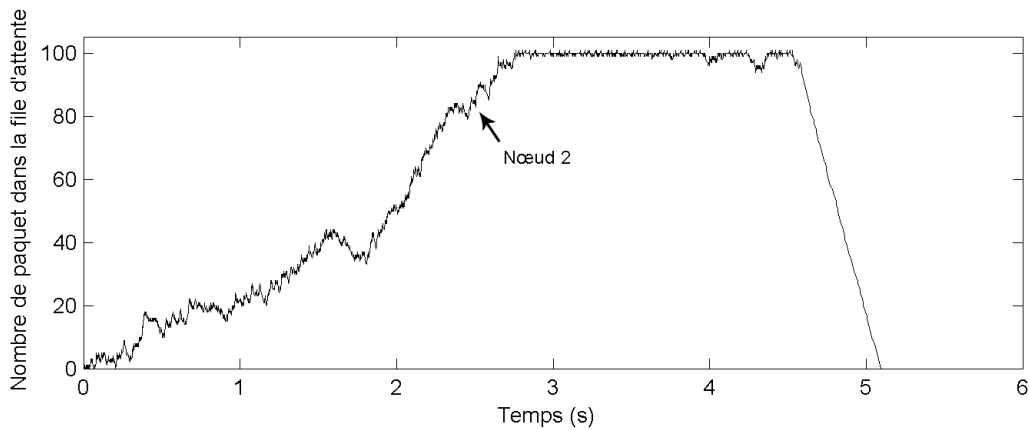


FIGURE 4.19 – Evolution de la file d’attente du nœud 2 avec un flux 1-4 et modification du paramètre *AIFS* sur les nœud 1 et 2

La courbe de la figure 4.19 révèle que la saturation sur le nœud 2 est toujours présente.

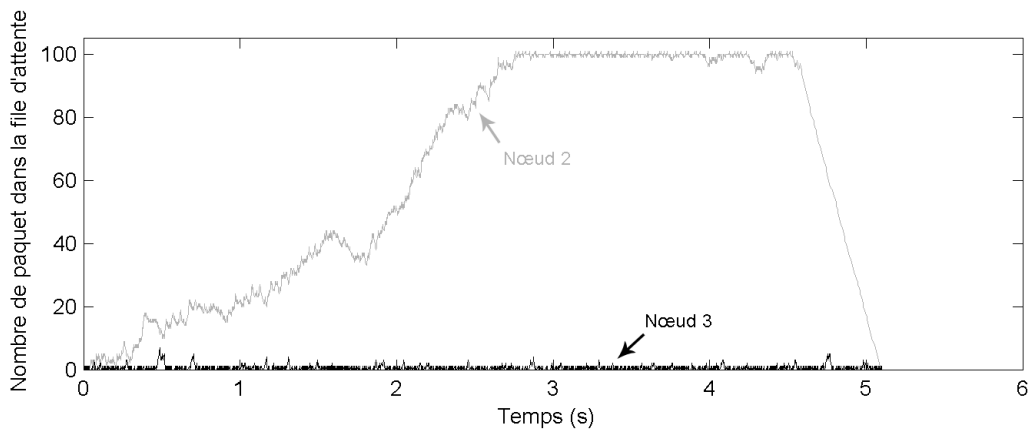


FIGURE 4.20 – Evolution des files d’attente des nœud 2 et 3 avec un flux 1-4 et modification du paramètre *AIFS* sur les nœud 1 et 2

L’état de la file d’attente du nœud 3 est présenté en noir sur la courbe de la figure 4.20 et celui du nœud 2 est rappelé en gris. Le nœud 3 n’accumule que peu de paquet dans sa file d’attente.

Les pertes de paquets au niveau des files d’attente n’ont lieu qu’au niveau du nœud 2, comme indiqué sur le tableau 4.14. Ceci est identique aux simulations précédentes avec un flux 1-4. Cependant, la saturation n’a lieu qu’à la date 2,5 secondes (figure 4.19), alors qu’elle n’a eu lieu qu’à 2 secondes dans la simulation où seul le paramètre

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	334	174	1085	92
2-1	0	0	460	0
2-3	0	318	637	0
3-2	0	0	1073	0
3-4	96	238	1219	0
4-3	238	0	601	0
Total	668	730	5075	92

TABLE 4.13 – Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec modification du paramètre *AIFS* sur les nœuds 1 et 2

*AIFS* du nœud 1 est modifié (figure 4.9). Le nombre de paquets perdus est donc moins important.

Nœud	<i>AIFS</i>		Pertes buffer			Pertes	UDP Débit	Délai UDP	
	1	2	2	3	Total			Moyen	Ecart type
Défaut	2	2	306	0	306	31 %	2067 $Kbit.s^{-1}$	838 ms	242 ms
<i>AIFS</i> 1	3	2	142	0	142	14 %	2058 $Kbit.s^{-1}$	878 ms	262 ms
<b><i>AIFS</i> 1-2</b>	<b>4</b>	<b>3</b>	<b>92</b>	<b>0</b>	<b>92</b>	<b>9 %</b>	<b>2137 <math>Kbit.s^{-1}</math></b>	<b>800 ms</b>	<b>253 ms</b>

TABLE 4.14 – Récapitulatif des simulations 1-4 en incluant la simulation modifiant le premier relais

Le tableau 4.14 permet de comparer les simulations réalisées avec un flux allant de 1 à 4. Nous pouvons constater que le nombre de pertes a diminué, mais est toujours à 9 %. De même, le débit a augmenté à 2137  $Kbit.s^{-1}$ , le délai moyen a diminué à 800 ms, et l'écart type a diminué à 253 ms. Les performances et la qualité de service sont donc meilleures, mais le nombre de pertes reste toujours trop important pour être acceptable.

#### 4.4.2 Augmentation des paramètres *AIFS* sur les nœud 1 et 2 pour un flux du nœud 1 à 6

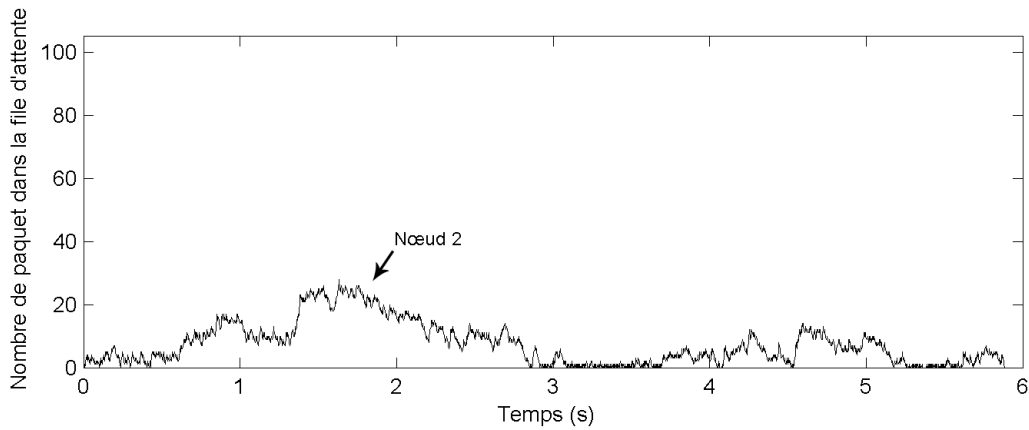


FIGURE 4.21 – Evolution de la file d'attente du nœud 2 avec un flux 1-6 et modification du paramètre *AIFS* sur les nœud 1 et 2

La figure 4.21 présente l'état de la file d'attente du nœud 2. Celle-ci reste peu remplie, et aucune perte n'est due à une saturation.

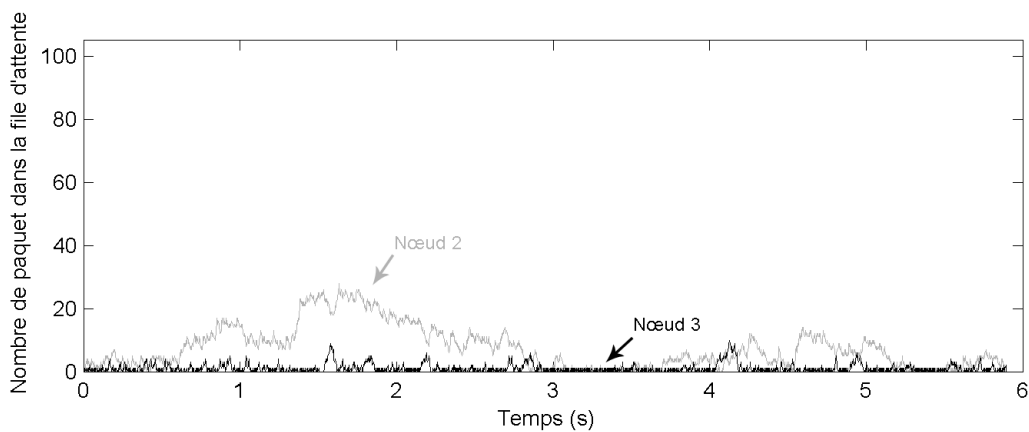


FIGURE 4.22 – Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 et modification du paramètre *AIFS* sur les nœud 1 et 2

La courbe noire de la figure 4.22 indique que la file d'attente du nœud 3 reste aussi très peu remplie.

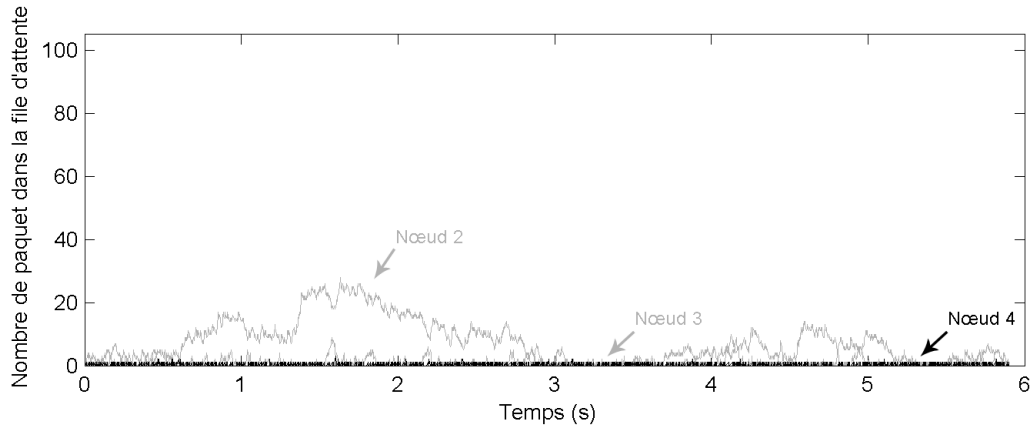


FIGURE 4.23 – Evolution des files d’attente des nœuds 2, 3 et 4 avec un flux 1-6 et modification du paramètre *AIFS* sur les nœuds 1 et 2

Enfin, la file d’attente du nœud 4, présentée sur la courbe noire de la figure 4.23, est toujours proche de 0. Cette simulation ne révèle aucune perte de paquet à cause d’un remplissage de file d’attente.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	1756	166	1587	0
2-1	4	1	489	0
2-3	120	329	629	0
3-2	0	0	1054	0
3-4	304	350	1447	0
4-3	227	0	678	0
4-6	1358	115	725	0
6-4	988	0	840	0
Total	4757	961	7449	0

TABLE 4.15 – Nombre et cause des pertes sur chaque lien pour la simulation 1-6 avec modification du paramètre *AIFS* sur les nœuds 1 et 2

Ceci est confirmé par le tableau 4.15 où aucune perte de paquet à cause d’un buffer plein n’est indiquée.

Cette simulation ne révèle aucune perte de paquet au niveau de la couche UDP. En effet, toutes les pertes de paquets au niveau radio n’ont pas entraîné de perte aux niveaux supérieurs grâce aux réémissions. De plus, le débit UDP est à  $2032 \text{ Kbit.s}^{-1}$ , ce qui est meilleur que lors des simulations précédentes avec ce type de flux. Le délai moyen est de 630 ms, et l’écart type à 127 ms. La qualité de service est donc meilleure.

Nœud	<i>AIFS</i>		Pertes buffer				Pertes	UDP	Délai UDP	
	1	2	2	3	4	Total		Débit	Moyen	Ecart type
Défaut	2	2	166	31	0	197	20 %	1813 $Kbit.s^{-1}$	1399 ms	435 ms
<i>AIFS</i> 1	3	2	0	0	0	0	1 %	1935 $Kbit.s^{-1}$	735 ms	175 ms
<b><i>AIFS</i> 1-2</b>	<b>4</b>	<b>3</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0 %</b>	<b>2032 <math>Kbit.s^{-1}</math></b>	<b>630 ms</b>	<b>127 ms</b>

TABLE 4.16 – Récapitulatif des simulations 1-6 en incluant la simulation modifiant le premier relais

#### 4.4.3 Bilan de l'impact de cet action

La modification du paramètre *AIFS* sur le nœud injecteur et sur le premier relais améliore le débit et la qualité de service dans les cas d'une transmission allant du nœud 1 à 4 mais aussi pour une transmission allant de 1 à 6. Ces indicateurs sont aussi meilleurs qu'une action seulement sur le premier nœud. Cependant, une saturation de la file d'attente est toujours présente avec un flux allant de 1 à 4, ce qui provoque des pertes toujours inadmissibles pour une transmission audio ou vidéo correcte.

Cependant, cette action positive peut être étendue à l'intégralité du chemin. Mais tous les nœuds ne peuvent pas avoir une valeur *AIFS* inférieure à celle des autres. L'adaptation de ce paramètre doit être dynamique. Une telle action relève donc d'une étude de "régulation de transfert", et donc être formulée selon une approche automatique. En effet, un nœud ayant une file d'attente vide ou se vidant pourrait augmenter son paramètre *AIFS*, et un nœud ayant une file d'attente pleine ou croissante pourrait diminuer son paramètre *AIFS*.

Avant d'envisager un tel comportement, nous nous sommes assurés que cette approche peut être applicable sur notre technologie. Ceci est le cas en allant lire dans le pilote WiFi le nombre de paquets en file d'attente et en agissant immédiatement sur le paramètre *AIFS*.

### 4.5 Régulation du trafic par l'utilisation du levier *AIFS*

A la vue d'un nœud, la réception et l'émission de paquets remplissent et vident la file d'attente. Si une file d'attente se remplit jusqu'à être pleine, les nouveaux paquets reçus sont perdus. Les pertes de paquets nuisant à la qualité de service, la saturation de la file d'attente est à éviter en priorité. De plus, si un nœud émet rapidement tout le contenu de sa file d'attente, le nœud suivant subit la réception de paquets et peut donc arriver dans la même situation de saturation de sa file d'attente. Ceci est donc aussi à éviter. Enfin, un niveau de remplissage faible permet aux paquets de traverser plus rapidement le réseau, et donc d'obtenir un délai de transmission plus faible, ce qui améliore la qualité de service. Cependant, les variations de ce délai ne doivent pas être trop importantes, car elles correspondent à la gigue, indicateur de qualité de service qui doit être faible.

Nous avons montré qu'une action portée sur le paramètre *AIFS* permet de prendre la parole plus souvent dans le but d'émettre un paquet. Ainsi, la file d'attente peut

être vidée plus rapidement que les autres nœuds. Mais nous avons constaté les mauvais résultats si une valeur minimale est fixée à ce paramètre sur tous les nœuds.

C'est pourquoi, diminuer ce levier uniquement lorsque la file d'attente en a besoin, peut permettre à une file d'attente de rester à un niveau peu élevé, tout en laissant les autres nœuds faire de même quand leurs files d'attente sont dans le même état.

Ceci correspond à un système avec en entrée l'état de la file d'attente, et en sortie la commande à appliquer sur le paramètre AIFS. Ainsi le système bouclé peut observer l'efficacité de sa commande et réguler l'état de la file d'attente vers une valeur faible, mais pas nulle, de manière à laisser le paramètre AIFS augmenter pour laisser la parole aux autres nœuds.

Ce mécanisme peut être appelé régulation de trafic.

#### 4.5.1 Principe du mécanisme de régulation local

La littérature recense plusieurs utilisations de systèmes automatiques appliqués aux réseaux sans fils respectant la norme 802.11. Dans [BCG<sup>+</sup>07], les auteurs présentent une loi de commande destinée à borner le délai d'acheminement de certains flux tout en limitant les pertes de paquets. De même, [DQ07] et [DW09] utilisent un PID (Proportionnel Intégrateur Dérivé) pour asservir le délai dans un réseau sans fil comportant qu'un point d'accès. Utiliser ce type de système pour asservir le levier AIFS peut améliorer peut donc améliorer la qualité de service automatiquement.

La figure 4.24 présente la règle d'adaptation automatique des paramètres à partir de la connaissance sur la file d'attente du nœud courant. Ce schéma correspond à un Proportionnel Dérivé (PD) couramment utilisé en automatique.

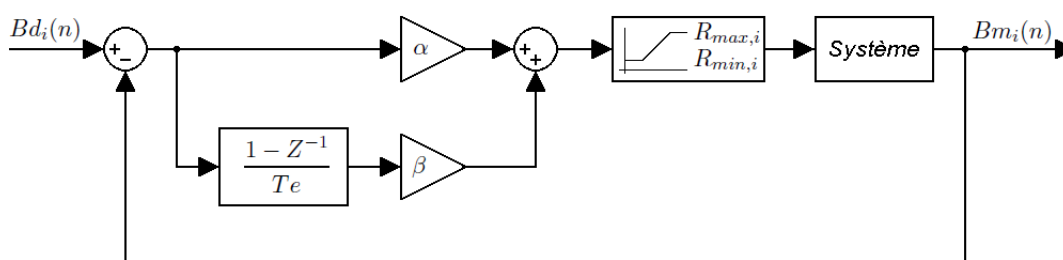


FIGURE 4.24 – Système présentant le régulateur de trafic

Ce système calculera et appliquera une nouvelle valeur à AIFS à la période  $Te$ . Cette durée ne doit pas être trop courte car le matériel doit pouvoir agir régulièrement à cette période sans être surchargé. Cependant, une trop grande durée rend la réaction de ce régulateur plus lente. Ainsi l'accumulation de paquets dans la file d'attente devient oscillante, ce qui augmente la gigue.

Les équations 4.1, 4.2, 4.3, 4.4 et 4.5 décrivent ce modèle sous forme algébrique.

$$R_i(n) = R_i(n-1) + \alpha (Bd_i(n) - Bm_i(n)) + \beta (\dot{Bd}_i(n) - \dot{Bm}_i(n)) \quad (4.1)$$

$$\dot{Bm}_i(n) = \frac{Bm_i(n) - Bm_i(n-1)}{Te} \quad (4.2)$$

$$\dot{B}d_i(n) = 0 \quad (4.3)$$

$$R_i(n) = \begin{cases} R_{min,i} & \text{si } R_i(n) < R_{min,i} \\ R_{max,i} & \text{si } R_i(n) > R_{max,i} \\ R_i(n) & \text{si } R_{min,i} \leq R_i(n) \leq R_{max,i} \end{cases} \quad (4.4)$$

$$AIFS_i(n) = int(R_i(n)) \quad (4.5)$$

Les variables suivantes sont utilisées dans les équations :

- $R_i(n)$  : valeur réelle bornée non arrondie de  $AIFS$  à l'instant  $n$ ,
- $R_{min,i}$  : borne inférieure de  $AIFS$  pour le nœud  $i$
- $R_{max,i}$  : borne supérieure de  $AIFS$  pour le nœud  $i$
- $AIFS_i(n)$  : nombre entier à affecter au paramètre  $AIFS$  à l'instant  $n$  au nœud  $i$ ,
- $Bd_i$  : nombre désiré d'élément dans la file d'attente,
- $Bm_i(n)$  : nombre mesuré d'éléments dans la file d'attente à l'instant  $n$ ,
- $\dot{B}m_i(n)$  dérivée discrète du nombre d'éléments dans la file d'attente, mesurée à l'instant  $n$ ,
- $Te$  : période de calcul et d'application d'un paramètre  $AIFS$ ,
- $\alpha$  : gain proportionnel,
- $\beta$  : gain dérivé.

#### 4.5.2 Identification des paramètres

Les paramètres  $\alpha$  et  $\beta$  de ce système doivent être configurés. Ceux-ci sont donc de nouveaux leviers à adapter pour améliorer les performances de la régulation du paramètre  $AIFS$ .

Les gains du système permettent d'accélérer ou de ralentir le temps de réponse sur le paramètre  $AIFS$  à chaque période :

- $\alpha$  joue sur l'écart entre le nombre désiré de paquets dans la file d'attente et celui qui est mesuré. Cette valeur est fixée à 0,05 dans un premier temps.
- $\beta$  influence la modification de  $AIFS$  à partir de la dérivée, qui correspond à la vitesse du changement du nombre d'éléments dans la file. Ce gain a la valeur 0,001 initialement.

De plus, les paramètres suivants permettent aussi de configurer le système :

- $R_{min,i}$  : Le système tend à diminuer  $AIFS$  jusqu'à cette valeur lorsque la file se remplit. Elle est configurée à la valeur 2.
- $R_{max,i}$  : Cette borne pourra être atteinte lorsque la file d'attente se vide. Cette valeur est fixée à 7.
- $R_i(0)$  :  $AIFS$  initial du nœud  $i$  est fixé à 4.
- $Bd_i$  : nombre de paquets désiré dans la file d'attente du nœud  $i$ . Le système modifie le paramètre  $AIFS$  pour permettre à la file d'attente d'atteindre ce niveau de remplissage. Cette valeur est fixée à 20 dans un premier temps.
- $Te$  : période de calcul et de modification de la valeur de  $AIFS$ . Ce temps indique la durée au bout de laquelle une nouvelle modification du paramètre  $AIFS$  pourra être faite. Ce délai a la valeur 50 ms initialement.

Une fois ce mécanisme implanté dans le simulateur, nous pouvons observer son effet avec ces valeurs initiales de paramètres.

Le nœud 1, injectant des paquets, a sa file d'attente pleine. Ce mécanisme n'est donc pas activé sur ce nœud et la valeur de *AIFS* est fixée à 5 dans un premier temps.

### 4.5.3 Résultats avec le régulateur pour un flux du nœud 1 à 4

Dans cette simulation, les valeurs suivantes sont configurées pour tous les nœuds à l'exception du nœud 1 :

- $\alpha = 0,05$
- $\beta = 0,001$
- $R(0) = 4$
- $Bd = 20$
- $Te = 50$  ms
- $R_{min} = 2$
- $R_{max} = 7$

Le paramètre *AIFS* du nœud 1 est fixé à 5.

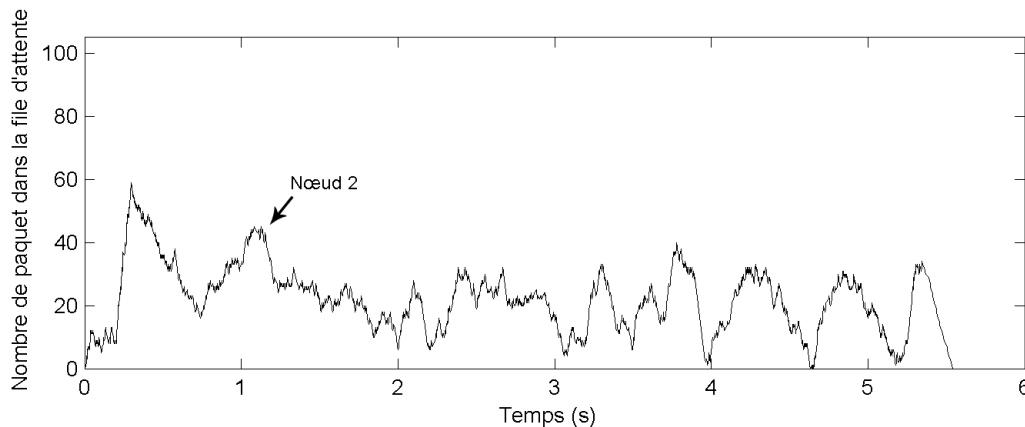


FIGURE 4.25 – Evolution de la file d'attente du nœud 2 avec un flux 1-4 et régulateur de trafic

La figure 4.25 présente l'état de la file d'attente du nœud 2. Nous pouvons constater que la courbe n'arrive pas à saturation, mais oscille autour de la valeur désirée *Bd* (20).

L'état de la file d'attente du nœud 3 est décrit sur la courbe noire de la figure 4.26. Celle-ci n'arrive pas non plus à saturation et tend aussi à osciller autour de la valeur 20, comme la courbe grise rappelant l'état de la file d'attente du nœud 2.



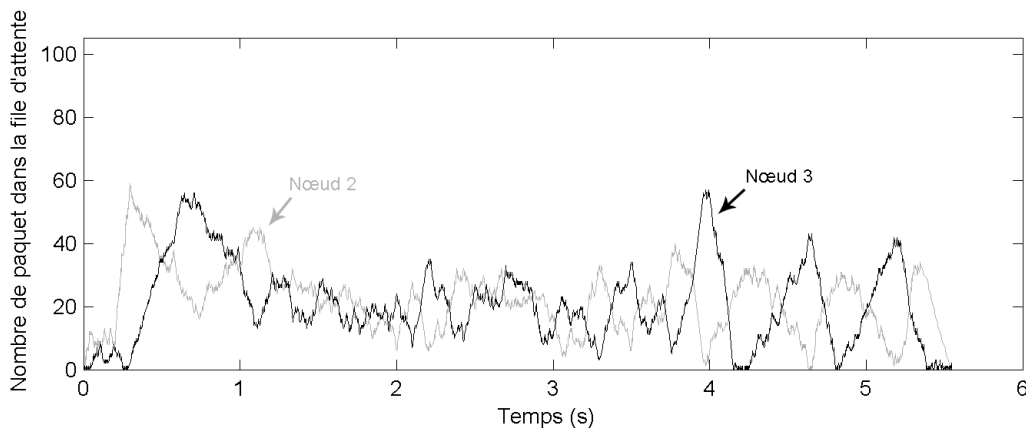


FIGURE 4.26 – Evolution des files d’attente des nœud 2 et 3 avec un flux 1-4 et régulateur de trafic

Le tableau 4.17 indique qu’aucun nœud n’a perdu de paquets à cause d’une saturation de la file d’attente conformément aux courbes précédentes.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	209	169	1090	0
2-1	0	0	508	0
2-3	0	310	652	0
3-2	0	0	1085	0
3-4	158	298	1352	0
4-3	51	0	651	0
Total	418	777	5338	0

TABLE 4.17 – Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec régulateur de trafic

Le débit est de  $2156 \text{ Kbit.s}^{-1}$ , ce qui correspond aux simulations précédentes. Cependant, aucune perte de paquet n’est due au buffer plein, et seulement 0,3 % des paquets ont été perdus au niveau UDP (contre 9 % dans la meilleure des simulations précédentes). Le délai moyen est de 775 ms avec un écart type de 169 ms. Ce reflet de la qualité de service est aussi amélioré.

Ce mécanisme supprime les pertes de paquets au niveau d’un dépassement de capacité des files d’attente dans le cas d’un flux allant du nœud 1 au nœud 4, tout en améliorant la qualité de service. Nous pouvons maintenant jouer sur les nouveaux leviers contrôlant ce système afin de choisir les meilleures valeurs.

Le tableau 4.18 récapitule les différentes étapes permettant d’affiner les valeurs de ces paramètres. La simulation nommée S1 correspond à la simulation précédente.

Les simulations S2 et S3 jouent sur la valeur *AIFS* du nœud 1. Nous retenons la valeur 4 utilisée dans la simulation S2 car les performances sont très sensiblement

supérieures au niveau du débit et de l'écart type. Une valeur trop faible présentée par la simulation S3 introduit à nouveau des pertes dues aux files d'attente pleines.

Le nombre de paquets présent dans le buffer désiré ( $Bd$ ) est affinée par les simulations S4 et S5. La valeur 5 est conservée car elle permet d'améliorer le délai et l'écart type.

Le gain  $\alpha$  est réglé à partir des simulations S6 et S7. Nous retenons la valeur 0,04 qui améliore le débit, le délai et l'écart type. La valeur plus faible utilisée dans la simulation S6 a un effet négatif sur la qualité de service.

Nom	gain		AIFS Nœud 1	$Bd$	Pertes buffer	UDP		Débit UDP	
	$\alpha$	$\beta$				Pertes	Débit ( $Kbit.s^{-1}$ )	Moyen	Ecart type
S1	0,05	0,001	5	20	0 %	0,3 %	2156	775 ms	169 ms
S2	0,05	0,001	4	20	0 %	0,3 %	2165	782 ms	167 ms
S3	0,05	0,001	3	20	6,4 %	6,9 %	2123	986 ms	247 ms
S4	0,05	0,001	4	10	0 %	0,5 %	2116	708 ms	138 ms
S5	0,05	0,001	4	5	0 %	0,2 %	2084	742 ms	164 ms
S6	0,03	0,001	4	5	0 %	0,6 %	2086	825 ms	189 ms
S7	0,04	0,001	4	5	0 %	0,2 %	2114	658 ms	137 ms
S8	0,04	0,002	4	5	0 %	0,2 %	2125	669 ms	135 ms
S9	0,04	0,0005	4	5	0 %	0,3 %	2112	664 ms	139 ms

TABLE 4.18 – Résultats de simulations avec flux 1-4 permettant d'optimiser le régulateur de trafic

Enfin, les simulations S8 et S9 jouent sur le paramètre  $\beta$ . La valeur 0,001 permet d'obtenir les meilleures performances.

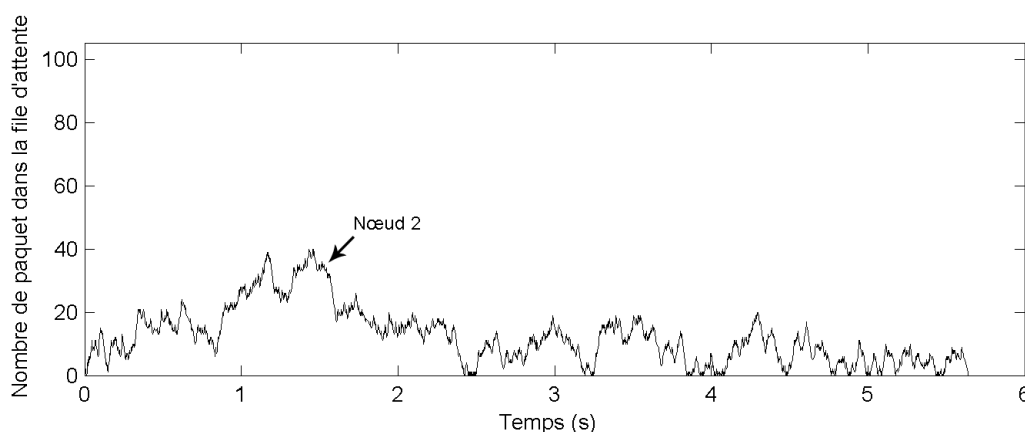


FIGURE 4.27 – Evolution de la file d'attente du nœud 2 avec un flux 1-4 et régulateur de trafic optimisé

La meilleure simulation (S7) permet d'obtenir un débit de  $2114 Kbit.s^{-1}$  et un délai moyen de 658 ms avec un écart type de 137 ms. Celle-ci a les paramètres suivants :

- $\alpha$  : 0,04
- $\beta$  : 0,001
- $R(0)$  : 4
- $Bd$  : 5
- $Te$  : 50 ms
- $R_{min} = 2$
- $R_{max} = 7$

La figure 4.27 représente l'état de la file d'attente du nœud 2. Celui-ci est régulé et reste à un niveau faible.

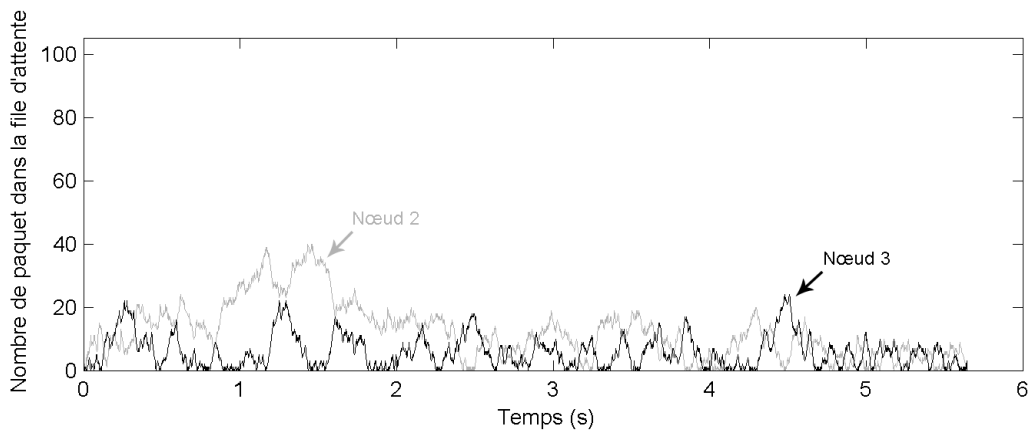


FIGURE 4.28 – Evolution des files d'attente des nœud 2 et 3 avec un flux 1-4 et régulateur de trafic optimisé

De même, l'état de la file d'attente du nœud 3 représenté par la courbe noire de la figure 4.28 reste peu élevé.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	484	165	1076	0
2-1	0	0	470	0
2-3	0	402	659	0
3-2	0	0	1058	0
3-4	186	411	1478	0
4-3	298	0	652	0
Total	968	978	5393	0

TABLE 4.19 – Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec régulateur de trafic optimisé

Nœud	AIFS		Pertes buffer			Pertes	UDP Débit	Délai UDP	
	1	2	2	3	Total			Moyen	Ecart type
Défaut	2	2	306	0	306	31 %	2067 $Kbit.s^{-1}$	838 ms	242 ms
AIFS 1	3	2	142	0	142	14 %	2058 $Kbit.s^{-1}$	878 ms	262 ms
AIFS 1-2	4	3	92	0	92	9 %	2137 $Kbit.s^{-1}$	800 ms	253 ms
<b>AIFS auto</b>	<b>4</b>	<b>auto</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>1 %</b>	<b>2114 <math>Kbit.s^{-1}</math></b>	<b>658 ms</b>	<b>137 ms</b>

TABLE 4.20 – Récapitulatif des simulations 1-4 sans puis avec régulateur de trafic optimisé

Ceci est confirmé par le tableau 4.19 qui indique qu'aucune perte n'a lieu à cause d'un dépassement de capacité des files d'attente.

Le tableau 4.20 rappelle les valeurs obtenues sans et avec ce mécanisme de régulation, et montre le gain de qualité de service apporté.

#### 4.5.4 Résultats avec le régulateur pour un flux du nœud 1 à 6

Il est important de vérifier que ce mécanisme n'affecte pas les performances obtenues précédemment avec un flux allant du nœud 1 au nœud 6. C'est pourquoi nous allons utiliser les paramètres du système déterminés à partir d'un flux 1-4 sur une nouvelle simulation utilisant un flux 1-6.

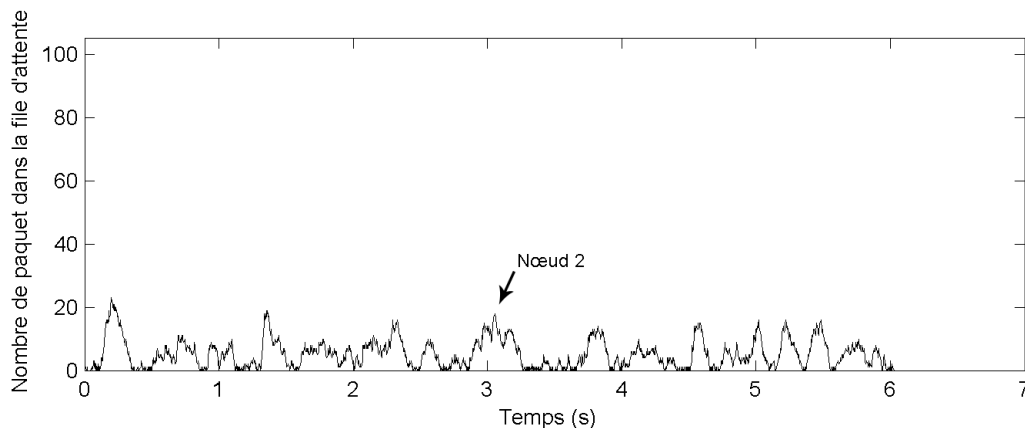


FIGURE 4.29 – Evolution de la file d'attente du nœud 2 avec un flux 1-6 avec régulateur de trafic optimisé

La figure 4.29 présente l'état de la file d'attente du nœud 2. Celui-ci reste à un niveau faible.

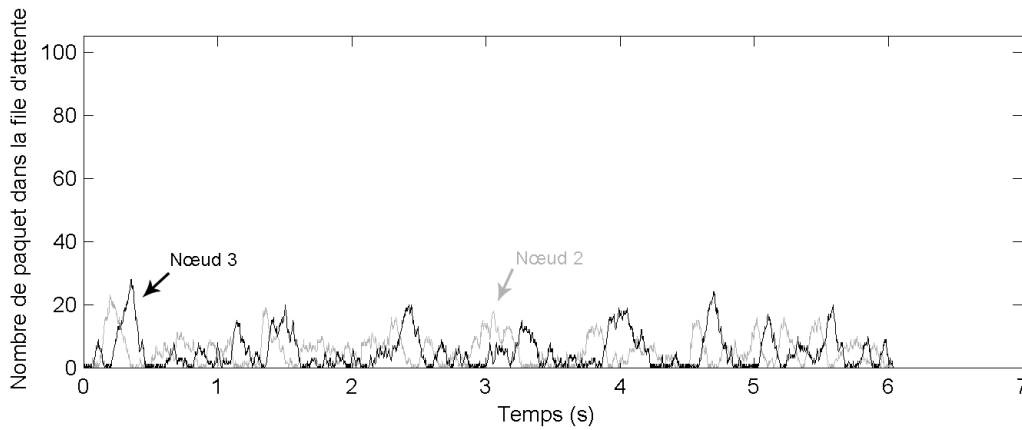


FIGURE 4.30 – Evolution des files d’attente des nœuds 2 et 3 avec un flux 1-6 avec régulateur de trafic optimisé

L’état de la file d’attente du nœud 3 est présenté par la courbe noire de la figure 4.30. De même que dans le nœud 2 rappelé en gris, cette courbe reste à des valeurs faibles.

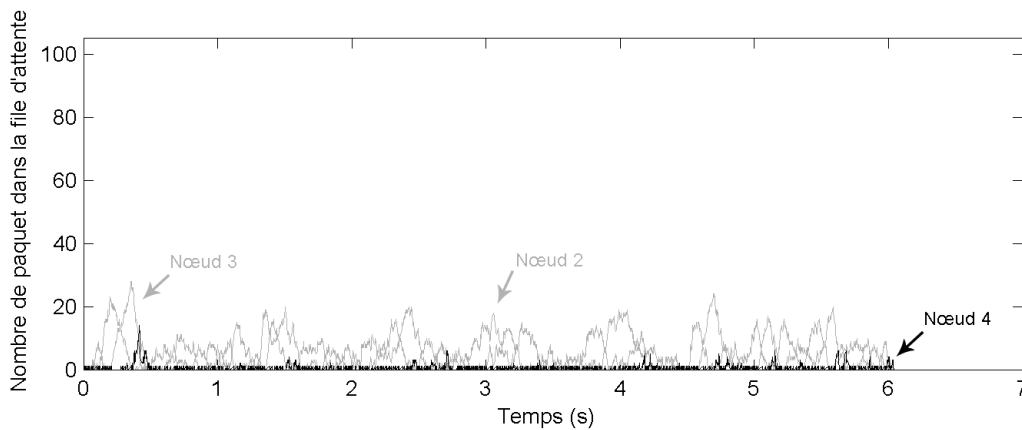


FIGURE 4.31 – Evolution des files d’attente des nœuds 2, 3 et 4 avec un flux 1-6 avec régulateur de trafic optimisé

Enfin, la courbe noire de la figure 4.31 présente l’état de la file d’attente du nœud 4. Tout comme les autres simulations impliquant le nœud 6, cette file d’attente reste presque vide.

Lien	Pertes causées par			
	collision	récepteur en émission	erreur radio	buffer plein
1-2	1812	239	1835	0
2-1	0	0	443	0
2-3	8	365	627	0
3-2	0	0	1038	0
3-4	246	348	1384	0
4-3	222	0	646	0
4-6	1118	15	687	0
6-4	722	0	796	0
Total	4128	967	7456	0

TABLE 4.21 – Nombre et cause des pertes sur chaque lien pour la simulation 1-6 avec régulateur de trafic optimisé

Le tableau 4.21 présente le bilan des pertes lors de simulation et ne fait apparaître aucune perte due à une file d'attente pleine.

Lors de cette simulation, le débit est de  $1979 \text{ Kbit.s}^{-1}$  et le délai moyen de 656 ms avec un écart type de 124 ms.

Nœud	AIFS		Pertes buffer				UDP		Délai UDP	
	1	2	2	3	4	Total	Pertes	Débit $\text{Kbit.s}^{-1}$	Moyen	Ecart type
Défaut	2	2	166	31	0	197	20 %	1813	1399 ms	435 ms
AIFS 1	3	2	0	0	0	0	1 %	1935	735 ms	175 ms
AIFS 1-2	4	3	0	0	0	0	0 %	2032	630 ms	127 ms
<b>AIFS auto</b>	<b>4</b>	<b>auto</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0 %</b>	<b>1979</b>	<b>656 ms</b>	<b>124 ms</b>

TABLE 4.22 – Récapitulatif des simulations 1-6 sans puis avec régulateur de trafic

Le tableau 4.22 rappelle les résultats obtenus sans et avec régulateur. Ceux-ci sont bien meilleurs que sans action sur ces nœuds, mais très sensiblement inférieur à une action sur le nœud injecteur et sur le premier relais. Cependant, la qualité de service apportée est tout à fait suffisante pour du transport de voix sur ip ou de vidéo.

#### 4.5.5 Proposition d'un mécanisme de régulation global

Ce mécanisme peut être étendu en permettant aux nœuds d'avoir une connaissance de l'état des files d'attente des nœuds voisins, voire de tous les nœuds du réseau. Ainsi, un nœud peut jouer différemment sur le levier AIFS, en sachant que le successeur a sa file vide ou pleine (ou plus généralement des informations qualitatives et/ou quantitatives sur l'état de ces files). De plus, l'évolution de l'état de la file d'attente du nœud précédent, qu'elle soit croissante ou décroissante, peut être un autre facteur permettant d'anticiper la transmission de paquets. Enfin, la notion de nœud précédent et de suc-

cesseur dépend de la direction du flux. En effet, dans nos premières simulations, nous n'avons pris en compte qu'un seul flux. Pour généraliser en considérant plusieurs flux de classes différentes, il est important de compter les paquets dans une file d'attente en fonction du nœud vers lequel ils doivent être émis, et de la classe de qualité de service associée.

Ceci nécessite la création de nouveaux paquets de contrôle, permettant à chaque nœud d'informer ses voisins de l'état de sa file d'attente, et de propager l'état des files d'attente de ses voisins. La fréquence de transmission de ces nouveaux paquets doit être peu élevée pour que ces paquets n'altèrent pas les performances du réseau. De plus, ce paquet doit être priorisé par rapport à la file d'attente, car s'il est injecté en queue, l'information qu'il contient peut être obsolète.

La transmission du nombre de paquets dans la file d'attente peut être périodique et donc systématique quel que soit l'état de la pile comme nous l'avons testé. Dans ce cas, cette information peut être incluse dans un nouveau champ d'un paquet de contrôle Beacon. En effet, la norme 802.11 permet d'ajouter des informations supplémentaires dans ce paquet mais aussi d'en modifier la fréquence de transmission. Cette transmission peut aussi être événementielle et ne se déclencher que lors du franchissement de certains seuils sur l'état de la file d'attente. Plusieurs seuils modifiant l'action du régulateur sont envisageables. Ainsi, ceci peut nous amener à une régulation floue.

Tout ceci pourra être simulé dans une étude ultérieure grâce à ce simulateur dans le but de qualifier la qualité de service apportée par ce mécanisme.

## 4.6 Conclusion

Ce chapitre a décrit les premières exploitations de ce simulateur sur une topologie de référence représentant un cas réel. Ces simulations sont axées sur le remplissage et la saturation des files d'attente sur les nœuds relayant les paquets lorsqu'un nœud source doit transmettre un grand nombre de paquets. Sur certains relais, la file d'attente se remplit car le nombre de paquets reçus est plus important que le nombre de paquets émis. Ainsi la file d'attente peut arriver à saturation et les nouveaux paquets arrivant sont perdus. D'une part, les pertes de paquets affectent énormément la qualité de service, mais de plus, le délai de transmission varie et augmente si un paquet attend au niveau d'un nœud. Ainsi, une communication audio ou vidéo traversant ce réseau WiFi maillé est de mauvaise qualité.

Ceci est le premier problème nuisant à la qualité de service, observé lors des simulations. En effet, la maîtrise et l'évolutivité de l'outil de simulation permettent d'approfondir un problème en analysant des compteurs et des événements, dans ce cas le niveau de remplissage des files d'attente. Des pistes d'actions ont d'abord été trouvées à partir de l'identification des leviers  $CW_{min}$  et  $AIFS$  ainsi que la qualification de leurs influences. Nous avons ensuite agi sur le paramètre le plus performant,  $AIFS$ , de manière statique sur le nœud source, puis sur les deux premiers nœuds, et observé quelques améliorations.

Une solution plus performante a été finalement apportée à ce problème à partir d'une étude du domaine de l'automatique. Une loi de commande, de type Proportionnel Dérivé, a été implémentée dans le modèle. Celle-ci contrôle dynamiquement le levier

*AIFS* à partir du niveau de remplissage des files d'attente  $Bm$ , agissant localement sur chaque nœud. Elle a permis d'apporter de fortes améliorations en terme de qualité de service.

Une évolution de cette loi de commande pourrait intégrer l'état des files d'attente des voisins, voire de l'ensemble des éléments du réseau. De nouveaux moyens de transmettre ces informations seraient à mettre en place entre chaque nœud.

Ce problème a été résolu sur cette topologie donnée, avec un flux unidirectionnel, mais il serait nécessaire d'analyser l'impact de ce mécanisme sur des topologies plus importantes et plus compliquées, avec des flux différents. De plus, ce mécanisme devra être implémenté sur le matériel pour apprécier les améliorations sur le terrain.

De nombreux axes de recherche sont ouverts maintenant que nous avons créé ce simulateur. Une exploitation plus approfondie est nécessaire, et pourrait faire apparaître d'autres situations où la qualité de service est à améliorer.





# Conclusion

Les communications audio et vidéo sont de plus en plus fréquentes au travers d'Internet et elles nécessitent une forte qualité de service, permettant une qualité sonore et une image correcte. Le développement de ce nouveau moyen de transport qui est le WiFi maillé permet d'apporter de l'accès Internet dans de grandes surfaces telles que les campings et résidences de vacances. Cependant, la répétition de paquets, sur un même canal radio par chacun des nœuds présents sur le chemin utilisé, altère la qualité de service des communications transportées. Ceci a été observé lors d'expérimentations.

Un état de l'art a présenté les problèmes et les solutions proposées dans la littérature. Ceux-ci ont révélé des problèmes de partage du canal radio ainsi que le problème du nœud caché. Ils entraînent une baisse du débit utile et de la qualité de service des communications. Malheureusement, les solutions proposées ne peuvent pas être implémentées sur notre architecture car elles nécessitent des modifications du protocole WiFi alors qu'une contrainte que nous avons est de rester compatible avec le matériel des clients.

L'étude expérimentale ne permet pas d'observer de manière microscopique les problèmes de transports altérant la qualité de service. En effet, afin de tout observer, il faudrait avoir connaissance de ce que chaque nœud émet et reçoit de manière synchronisée temporellement, tout en maîtrisant l'environnement radio. Ceci étant difficilement réalisable, nous nous sommes orientés vers une étude par simulation.

Ce mémoire de thèse présente un simulateur, créé dans le but de comprendre et d'agir sur les problèmes liés à cette méthode de transport. Celui-ci est basé sur les réseaux de Petri colorés temporisés stochastiques et hiérarchisés, formalisme de modélisation permettant une forte évolutivité et un raffinement successif du modèle en fonction des problèmes à étudier. Il permet de simuler différents flux réseaux au travers d'un réseau WiFi maillé dont la topologie est configurable. Cette modélisation a été validée à partir de la description et des mesures de liens effectués sur un cas réel.

Une première exploitation du simulateur sur cette topologie de référence a permis d'observer un remplissage des files d'attente d'émission de certains nœuds présents sur le chemin de communication. Ainsi la saturation de celles-ci entraîne des pertes de paquets, et la gigue mesurée sur un tel réseau est détériorée. Ces deux impacts provoquent une mauvaise qualité de service. Après avoir identifié les leviers permettant de contrôler le remplissage de ces files d'attente, nous avons pu jouer sur ceux-ci et observer une amélioration. Enfin, nous avons mis en place une loi de commande du domaine de l'automatique permettant de jouer automatiquement et dynamiquement sur ces leviers en fonction du remplissage des files d'attente. Cette loi, implémentée dans le modèle,

a apporté de meilleures performances par rapport aux simulations précédentes pour lesquelles soit la configuration était par défaut, soit l'action était figée sur un levier.

Notre contribution se situe donc à deux niveaux : d'une part la création d'un simulateur évolutif adapté aux réseaux WiFi maillés permettant la recherche de problèmes altérant la qualité de service d'une communication, et d'autre part la mise en place d'une loi de commande adaptant les leviers en fonction de l'état de remplissage des files d'attente.

## Perspectives

Une première perspective serait d'implémenter cette amélioration sur le matériel que nous utilisons. Nous pourrions alors mesurer son impact de manière quantitative directement sur le terrain, mais aussi tester des communications audio et vidéo et de fait apprécier l'amélioration apportée.

Cet outil pourrait être utilisé pour déterminer le meilleur déploiement des nœuds sur le terrain. Lors de notre étude, nous avons pris une installation existante pour l'étudier. À l'inverse, le choix entre les différentes possibilités de positionnement des nœuds sur une nouvelle installation pourrait être fait après simulation. Ainsi, la qualité du réseau pourrait être optimisée, et connue à l'avance.

Ensuite, cette amélioration devra être testée sur d'autres exemples d'implantation de nœuds sur le terrain, afin d'observer l'impact qu'elle aura. Cette recherche s'étant limitée à une topologie donnée, les améliorations pourraient être plus ou moins importantes.

Une autre perspective serait la mise en place d'un mécanisme de régulation global collaboratif entre chaque nœud. En effet, chacun d'eux pourrait transmettre à ses voisins l'état de sa file d'attente, ils pourraient alors anticiper les paquets qui leur arriveraient ainsi que l'impact que leurs émissions auraient sur les autres nœuds. En fonction de ces facteurs, les nœuds pourraient automatiquement et dynamiquement adapter leurs leviers dans le but de limiter l'impact négatif sur la qualité de service. La transmission de ces informations peut être périodique à une fréquence à déterminer. Si celle-ci est trop élevée, le canal radio est beaucoup utilisé avec ces paquets de contrôle. Si elle est trop faible, la réaction sera lente. La transmission périodique de ces informations peut être transportée dans les Beacons. Cette communication peut aussi être événementielle. Ainsi, l'état de la file d'attente sera connu des voisins si un seuil est dépassé. Plusieurs seuils peuvent aussi être déterminés et associés à différents niveaux d'action sur le levier AIFS. L'utilisation de notre simulateur pourra permettre d'évaluer les améliorations de qualité de service avec ces mécanismes.

De plus, maintenant qu'un outil est créé, de nouvelles simulations pourraient être faites sur d'autres topologies, avec des flux différents avec pour objectif de trouver d'autres causes d'altération de la qualité de service. Ceci permettrait à terme d'avoir un ensemble de lois de commande et d'adaptations des leviers qui garantiraient de bonnes performances quelle que soit la topologie du terrain sur lequel serait installé un réseau de WiFi maillé.

Enfin, de nouveaux modèles pourront être implémentés dans cet outil évolutif. Par exemple, la couche TCP permettrait d'évaluer les conséquences de pertes de paquets au niveau liaison de donnée, ce qui entraînerait des réémissions au niveau TCP. L'utilisation

de plusieurs canaux pourrait être implémentée dans ce simulateur. Ceci permettrait d'évaluer les performances d'une couverture WiFi plus étendue par l'interconnexion de plusieurs sous réseaux de WiFi maillé.



# Bibliographie

- [All11] Wi-Fi Alliance. Wi-fi alliance. <http://www.wi-fi.org>, 2011.
- [BCG05] R. Bruno, M. Conti, and Gregori. Mesh networks : commodity multihop ad hoc networks. *Communications Magazine IEEE*, 43 :123–131, Mars 2005.
- [BCG<sup>+</sup>07] G. Boggia, P. Camarda, L.A. Grieco, S. Mascolo, and A. Stefanelli. Performance evaluation of a feedback based dynamic scheduler for 802.11e mac. *Vehicular Technology Conference*, pages 835–839, Avril 2007.
- [BMG<sup>+</sup>10] N. Baldo, M. Miozzo, F. Guerra, M. Rossi, and Michele Z. Miracle : The multi-interface cross-layer extension of ns2. *EURASIP Journal on Wireless Communications and Networking*, Avril 2010.
- [DQ07] F. Du and Q.Q. Qian. Fuzzy immune self-regulating pid control for wireless networked control system. *Wireless, Mobile and Sensor Networks*, pages 1081–1084, Fév 2007.
- [DRFM08] F. De Rango, M. Fotino, and S. Marano. Ee-olsr : Energy efficient olsr routing protocol for mobile ad-hoc networks. *Military Communications Conference 2008. MILCOM 2008. IEEE*, page 1, Nov 2008.
- [DW09] F. Du and Du W. Novel smith predictor and cmac-pid control for wireless networked control systems. *Information Processing*, pages 59–62, Juil 2009.
- [GGFK10] N.A. Gdoura, S.M. Gammar, F. Filali, and F. Kamoun. Meshcast : A multi-channel multi-interface multicast protocol for mesh networks, Sept 2010.
- [Gre11] GreatSPN. Greatspn 2.0. <http://www.di.unito.it/~greatspn/index.html>, 2011.
- [HDM<sup>+</sup>10] G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke. Ieee 802.11s : The wlan mesh standard. *Wireless Communications IEEE*, 17 :104–111, Fév 2010.
- [IEE99a] IEEE. 802.11 wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, page 1, 1999.
- [IEE99b] IEEE. 802.11a high-speed physical layer in the 5 ghz band. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, 1999.

- [IEE99c] IEEE. 802.11b higher-speed physical layer extension in the 2.4 ghz band. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, 1999.
- [IEE03] IEEE. 802.11g further higher data rate extension in the 2.4 ghz band. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, 2003.
- [IEE05] IEEE. 802.11e medium access control (mac) quality of service enhancements. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, 2005.
- [IEE09] IEEE. 802.11n enhancements for higher throughput. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, 2009.
- [IEE11] IEEE. 802.11s wireless lan (wlan) mesh basic service set. *IEEE Computer Society LAN MAN Standards Committee. 802.11*, 2011.
- [JAB08] Issam JABRI. *Gestion dynamique des topologies sans fils*. PhD thesis, Université Henri Poincaré, Nancy I, 2008.
- [JR05] Zhu Jing and S. Roy. 802.11 mesh networks with two-radio access points. *Communications ICC IEEE International Conference on*, 5 :3609–3615, Mai 2005.
- [JYM07] Zhang Jian, Chen Yuanzhu, and I. Marsic. Persistent pseudo-clearance problem in ieee802.11 mesh networks and its multicast based solutions. *Local and Metropolitan Area Networks LANMAN 2007. 15th IEEE Workshop on*, pages 1–6, Juin 2007.
- [KJCV06] P. Kyasanur, So Jungmin, C. Chereddi, and N.H. Vaidya. Multichannel mesh networks : challenges and protocols. *Wireless Communications IEEE*, 13 :30–36, Avril 2006.
- [KRR<sup>+</sup>09] A. Kandhalu, A. Rowe, R. Rajkumar, Chingchun Huang, and Chao-Chun Yeh. Real-time video surveillance over ieee 802.11 mesh networks. *Real-Time and Embedded Technology and Applications Symposium 15th IEEE*, Avril 2009.
- [LKC05] X. Li, P.-Y. Kong, and K.-C. Chua. Analysis of tcp throughput in ieee 802.11 based multi-hop ad hoc networks. *Computer Communications and Networks ICCCN 2005 Proceedings. 14th International Conference on*, pages 297–302, Oct 2005.
- [LPSY09] A. Lyakhov, I. Pustogarov, A. Safonov, and M. Yakimov. Starvation effect study in ieee 802.11 mesh networks. *Mobile Adhoc and Sensor Systems MASS '09. IEEE 6th International Conference on*, pages 651–656, Oct 2009.
- [MPV06] R. Moraes, P. Portugal, and F. Vasques. A stochastic petri net model for the simulation analysis of the ieee 802.11e edca communication protocol. *Emerging Technologies and Factory Automation*, pages 38–45, Sept 2006.
- [MSK<sup>+</sup>10] S. Mehta, N. Sulatan, H. Kabir, N. Ullah, and Kwak K. S. Network and system simulation tools for next generation networks : A case study. *Modelling, Simulation and Identification*, Aout 2010.

- [NHF07] D. Niyato, E. Hossain, and A. Fallahi. Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks : Performance analysis and optimization. *Mobile Computing IEEE Transactions on*, 6 :221, Fév 2007.
- [NNCA06] N.S. Nandiraju, D.S. Nandiraju, D. Cavalcanti, and D.P. Agrawal. A novel queue management mechanism for improving performance of multi-hop flows in ieee 802.11s based mesh networks. *Performance Computing and Communications Conference IPCCC 2006. 25th IEEE International*, pages 161–168, Avril 2006.
- [NS-11a] NS-2. The network simulator - ns-2. <http://www.isi.edu/nsnam/ns>, 2011.
- [NS-11b] NS-3. The ns-3 network simulator. <http://www.nsnam.org>, 2011.
- [nt11] Scalable network technologies. Qualnet. <http://www.scalable-networks.com/products/qualnet>, 2011.
- [OPN11] OPNET. Application and network performance with opnet. <http://www.opnet.com/>, 2011.
- [OQDS06] W.D. Obal, Akber Quereshi, D.D. Deavours, and W.H. Sanders. Overview of ultrasn, Sept 2006.
- [RAGK08] S. Razak, N.B. Abu-Ghazaleh, and V. Kolar. Modeling of two-flow interactions under sir model in multi-hop wireless networks. *Local Computer Networks LCN 2008. 33rd IEEE Conference on*, page 297, Oct 2008.
- [RD09] M. Ramadhan and M. Davis. Performance study of a cross-layer optimization to the dsr routing protocol in wireless mesh networks. *Network and Service Security 2009. N2S '09. International Conference on*, page 1, Juin 2009.
- [SKGR11] Bill Sanders, Ken Keefe, Shravan Gaonkar, and Eric Rozier. The môbius tool. <https://www.mobius.illinois.edu>, 2011.
- [SL09] H. Somnuk and M. Lerwatechakul. Multi-hop aodv-2t. *Intelligent Ubiquitous Computing and Education 2009 International Symposium on*, page 214, Mai 2009.
- [SPK<sup>+</sup>07] Singh Sumit, Aravinda Prashanth, Acharya Kumar, Madhowa Upamanyu, and M. Belding-Royer Elizabeth. Sticky csma/ca : Implicit synchronization and real-time qos in mesh networks. *Elsevier Science*, Janv 2007.
- [TCCYMC06] Hou Ting-Chao, Wang Chien-Yi, and Chan Ming-Chieh. A token-based distributed scheduling for mesh networks with chain topologies. *Advanced Information Networking and Applications AINA 2006. 20th International Conference on*, page 6, Avril 2006.
- [URKRZ<sup>+</sup>08] K. Ur Rahman Khan, A.V. Reddy, R.U. Zaman, K.A. Reddy, and T.S. Harsha. An efficient dsdv routing protocol for wireless mobile ad hoc networks and its performance comparison. *Computer Modeling and Simulation 2008. EMS '08. Second UKSIM European Symposium on*, page 506, Sept 2008.



- [Var01] A. Varga. The omnet++ discrete event simulation system. *Proceedings of the European Simulation Multiconference (ESM'2001)*, 2001.
- [Var11] A. Varga. Omnet++ network simulation framework. <http://www.omnetpp.org/>, 2011.
- [VH08] A. Varga and R. Hornig. An overview of the omnet++ simulation environment. *Simutools '08 : Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems workshops*, pages 1–10, 2008.
- [WV11] Michael Westergaard and Eric Verbeek. Cpn tools. <http://cpntools.org>, 2011.
- [XG06] Zhu Xiaoqing and B. Girod. Media-aware multi-user rate allocation over wireless mesh network. *Operator-Assisted (Wireless Mesh) Community Networks 2006 1st Workshop on*, pages 1–8, Sept 2006.
- [YJSHYB<sup>+</sup>11] Son Yu-Jin, Lee Sung-Hee, Ko Young-Bae, Park Sang-Joon, and Song Yoo-Seung. Session-oriented adaptive routing for improving path stability in wlan based mesh networks, Fév 2011.

# Table des figures

1.1	Système de distribution WiFi proposé par IEEE 802.11 . . . . .	6
1.2	Mode infrastructure . . . . .	7
1.3	Mode ad-hoc . . . . .	7
1.4	Exemple de réseau maillé . . . . .	8
1.5	Exemple de zone de réception . . . . .	8
1.6	Désignation des nœuds . . . . .	9
1.7	Différents délais intertrames présentés dans la norme [IEE99a] . . . . .	11
1.8	Exemple de gain de période de contention . . . . .	11
1.9	Exemple de collision par tir du même backoff . . . . .	12
1.10	Paquet de contrôle RTS/CTS . . . . .	12
1.11	Topologie du réseau expérimental . . . . .	16
1.12	RTT sans WMM avec 1 RTP . . . . .	17
1.13	RTT sans WMM avec 1 RTP et 1 HTTP . . . . .	17
1.14	RTT sans WMM avec 1 RTP et 2 HTTP . . . . .	18
1.15	RTT avec WMM avec 1 RTP . . . . .	18
1.16	RTT avec WMM avec 1 RTP et 1 HTTP . . . . .	18
1.17	RTT avec WMM avec 1 RTP et 2 HTTP . . . . .	19
1.18	Gigue sans WMM avec 1 RTP . . . . .	20
1.19	Gigue sans WMM avec 1 RTP et 1 HTTP . . . . .	20
1.20	Gigue sans WMM avec 1 RTP et 2 HTTP . . . . .	20
1.21	Gigue avec WMM avec 1 RTP . . . . .	21
1.22	Gigue avec WMM avec 1 RTP et 1 HTTP . . . . .	21
1.23	Gigue avec WMM avec 1 RTP et 2 HTTP . . . . .	21
2.1	Topologie présentant quatres noeuds alignés . . . . .	26
2.2	Topologie et zone d'influence . . . . .	28
2.3	Topologie et influence des nœuds de cette étude . . . . .	28
2.4	Planification des émissions . . . . .	29
2.5	Topologie des nœuds X Y B A . . . . .	29
2.6	Description du problème . . . . .	30
2.7	Différents scénarios d'influence . . . . .	30
2.8	Topologie présentant différents scénarios d'influence . . . . .	31
3.1	Exemple de place . . . . .	45
3.2	Exemple de transition . . . . .	45

3.3	Exemple d'arc . . . . .	45
3.4	Exemple de macro transition . . . . .	46
3.5	Exemple de place d'entrée . . . . .	46
3.6	Exemple de place de sortie . . . . .	46
3.7	Exemple de place d'entrée sortie . . . . .	47
3.8	Exemple de transmission sur un réseau maillé . . . . .	47
3.9	Exemple de transmission sur le modèle Internet . . . . .	48
3.10	Transmission sur chaque nœud . . . . .	49
3.11	Repliage du modèle de la transmission . . . . .	49
3.12	Présentation des éléments du modèle . . . . .	50
3.13	Disposition des nœuds avec les rapports signal sur bruit . . . . .	54
3.14	Modèle <i>Network</i> . . . . .	57
3.15	Modèle Radio - 1 <sup>ère</sup> étape - initialisation . . . . .	59
3.16	Modèle Radio - 2 <sup>ème</sup> étape - duplication . . . . .	60
3.17	Modèle Radio - 3 <sup>ème</sup> étape - temporisation et occupation du médium . . . . .	61
3.18	Modèle Radio - 4 <sup>ème</sup> étape - gestion des interactions entre paquets . . . . .	62
3.19	Modèle Radio - 5 <sup>ème</sup> étape - libération du canal radio . . . . .	63
3.20	Modèle Radio - 6 <sup>ème</sup> étape - tir de probabilité de réussite . . . . .	64
3.21	Modèle <i>Radio</i> complet . . . . .	65
3.22	Modèle <i>MAC</i> décomposé en sous parties . . . . .	66
3.23	Modèle <i>MAC</i> - 1 <sup>ère</sup> étape - initialisation du modèle . . . . .	67
3.24	Modèle <i>MAC</i> - 2 <sup>ème</sup> étape - classification et priorisation . . . . .	68
3.25	Modèle <i>AccessCategories</i> - 1 <sup>ère</sup> étape - initialisation du modèle . . . . .	69
3.26	Modèle <i>AccessCategories</i> - 2 <sup>ème</sup> étape - ajout d'un paquet . . . . .	69
3.27	Modèle <i>AccessCategories</i> - 3 <sup>ème</sup> étape - traitement d'un paquet . . . . .	70
3.28	Modèle <i>AccessCategories</i> complet . . . . .	72
3.29	Modèle <i>MAC</i> - 3 <sup>ème</sup> étape - décompte du temps intertrame . . . . .	73
3.30	Modèle <i>Iifs</i> - 1 <sup>ère</sup> étape - entrée du paquet et vérification du canal radio . . . . .	74
3.31	Modèle <i>Iifs</i> - 2 <sup>ème</sup> étape - temporisation . . . . .	75
3.32	Modèle <i>Iifs</i> - 3 <sup>ème</sup> étape - sortie du paquet . . . . .	75
3.33	Modèle <i>Iifs</i> complet . . . . .	76
3.34	Modèle <i>MAC</i> - 4 <sup>ème</sup> étape - Période de contention . . . . .	77
3.35	Modèle <i>Contention</i> - 1 <sup>ère</sup> étape - Entrée d'un paquet . . . . .	78
3.36	Modèle <i>Contention</i> - 2 <sup>ème</sup> étape - Temporisation . . . . .	79
3.37	Modèle <i>Contention</i> complet . . . . .	80
3.38	Modèle <i>MAC</i> - 5 <sup>ème</sup> étape - Emission de paquet . . . . .	81
3.39	Modèle <i>ACKRTSCTS</i> - 1 <sup>ère</sup> étape - Initialisation . . . . .	82
3.40	Modèle <i>ACKRTSCTS</i> - 2 <sup>ème</sup> étape - Emission de paquet ou de RTS . . . . .	82
3.41	Modèle <i>ACKRTSCTS</i> - 3 <sup>ème</sup> étape - Attente de CTS . . . . .	83
3.42	Modèle <i>ACKRTSCTS</i> - 4 <sup>ème</sup> étape - Réception de RTS et de CTS . . . . .	84
3.43	Modèle <i>ACKRTSCTS</i> complet . . . . .	85
3.44	Modèle <i>SendWithSIFS</i> . . . . .	86
3.45	Modèle <i>MAC</i> - 6 <sup>ème</sup> étape - Réception de paquet . . . . .	87
3.46	Modèle <i>MACReceive</i> - 1 <sup>ère</sup> étape - Initialisation et introduction de paquet . . . . .	88
3.47	Modèle <i>MACReceive</i> - 2 <sup>ème</sup> étape - Aiguillage du paquet . . . . .	88
3.48	Modèle <i>MACReceive</i> - 3 <sup>ème</sup> étape - Suppression des paquets reçus en double . . . . .	89

3.49	Modèle <i>MACReceive</i> complet . . . . .	90
3.50	Modèle <i>MAC</i> - 7 <sup>ème</sup> étape - Réception d'accusé de réception . . . . .	90
3.51	Modèle <i>RecvAck</i> complet . . . . .	91
3.52	Modèle <i>MAC</i> - 8 <sup>ème</sup> étape - Emission périodique de beacon . . . . .	92
3.53	Modèle <i>Beacon</i> complet . . . . .	93
3.54	Modèle <i>MAC</i> complet . . . . .	94
3.55	Modèle <i>LLC</i> - 1 <sup>ère</sup> étape - Emission de paquet . . . . .	95
3.56	Modèle <i>LLC</i> - 2 <sup>ème</sup> étape - Répétition de paquet . . . . .	96
3.57	Modèle <i>LLC</i> - 3 <sup>ème</sup> étape - Réception de paquet pour le nœud courant . . . . .	97
3.58	Modèle <i>LLC</i> complet . . . . .	97
3.59	Modèle <i>UDP</i> - 1 <sup>ère</sup> étape - Emission de paquet . . . . .	98
3.60	Modèle <i>UDP</i> - 2 <sup>ème</sup> étape - Réception de paquet . . . . .	99
3.61	Modèle <i>UDP</i> complet . . . . .	100
3.62	Modèle <i>ICMP</i> - 1 <sup>ère</sup> étape - Emission d'une requête . . . . .	101
3.63	Modèle <i>ICMP</i> - 2 <sup>ème</sup> étape - Réponse à une requête . . . . .	101
3.64	Modèle <i>ICMP</i> - 3 <sup>ème</sup> étape - Réception de réponses . . . . .	102
3.65	Modèle <i>ICMP</i> complet . . . . .	103
3.66	Plan du cas d'étude . . . . .	107
3.67	Légende des graphes . . . . .	109
3.68	Graphe généré pour la transmission d'un échange ICMP . . . . .	110
3.69	Première partie du graphe généré lors d'un échange ICMP . . . . .	111
3.70	Seconde partie du graphe généré lors d'un échange ICMP . . . . .	113
3.71	Graphe généré pour la transmission d'échanges ICMP et UDP . . . . .	114
3.72	Echanges ICMP et UDP zoomée entre 10000 $\mu$ s et 13000 $\mu$ s . . . . .	115
3.73	Echanges ICMP et UDP zoomée peu après la date 30000 $\mu$ s . . . . .	116
3.74	Graphe généré par une simulation de RTS CTS . . . . .	117
3.75	Graphe généré lors d'une simulation de beacon . . . . .	119
4.1	Topologie de la simulation . . . . .	122
4.2	Etat de la file d'attente du nœud 2 avec un flux 1-4 . . . . .	123
4.3	Etat des files d'attente des nœuds 3 et 4 avec un flux 1-4 . . . . .	124
4.4	Etat de file d'attente du nœud 2 avec un flux 1-6 . . . . .	125
4.5	Etat des files d'attente des nœuds 2 et 3 avec un flux 1-6 . . . . .	126
4.6	Etat des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6 . . . . .	126
4.7	Gain ou perte de période de contention . . . . .	128
4.8	Collision due au tir de période de contention . . . . .	128
4.9	Evolution de la file d'attente du nœud 2 avec un flux 1-4 et modification du paramètre <i>AIFS</i> sur le nœud source . . . . .	131
4.10	Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-4 et modification du paramètre <i>AIFS</i> sur le nœud source . . . . .	131
4.11	Evolution de la file d'attente du nœud 2 avec un flux 1-6 et modification de <i>AIFS</i> . . . . .	133
4.12	Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 et modification du paramètre <i>AIFS</i> sur le nœud source . . . . .	133
4.13	Evolution des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6 et modification du paramètre <i>AIFS</i> sur le nœud source . . . . .	134

4.14	Evolution de la file d'attente du nœud 2 avec un flux 1-4 et modification du paramètre $CW_{min}$ sur le nœud source . . . . .	135
4.15	Evolution des files d'attente des nœud 2 et 3 avec un flux 1-4 et modification du paramètre $CW_{min}$ sur le nœud source . . . . .	135
4.16	Evolution de la file d'attente du nœud 2 avec un flux 1-6 et modification du paramètre $CW_{min}$ sur le nœud source . . . . .	136
4.17	Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 et modification du paramètre $CW_{min}$ sur le nœud source . . . . .	137
4.18	Evolution des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6 et modification du paramètre $CW_{min}$ sur le nœud source . . . . .	137
4.19	Evolution de la file d'attente du nœud 2 avec un flux 1-4 et modification du paramètre $AIFS$ sur les nœud 1 et 2 . . . . .	140
4.20	Evolution des files d'attente des nœud 2 et 3 avec un flux 1-4 et modification du paramètre $AIFS$ sur les nœud 1 et 2 . . . . .	140
4.21	Evolution de la file d'attente du nœud 2 avec un flux 1-6 et modification du paramètre $AIFS$ sur les nœud 1 et 2 . . . . .	142
4.22	Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 et modification du paramètre $AIFS$ sur les nœud 1 et 2 . . . . .	142
4.23	Evolution des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6 et modification du paramètre $AIFS$ sur les nœud 1 et 2 . . . . .	143
4.24	Système présentant le régulateur de trafic . . . . .	145
4.25	Evolution de la file d'attente du nœud 2 avec un flux 1-4 et régulateur de trafic . . . . .	147
4.26	Evolution des files d'attente des nœud 2 et 3 avec un flux 1-4 et régulateur de trafic . . . . .	148
4.27	Evolution de la file d'attente du nœud 2 avec un flux 1-4 et régulateur de trafic optimisé . . . . .	149
4.28	Evolution des files d'attente des nœud 2 et 3 avec un flux 1-4 et régulateur de trafic optimisé . . . . .	150
4.29	Evolution de la file d'attente du nœud 2 avec un flux 1-6 avec régulateur de trafic optimisé . . . . .	151
4.30	Evolution des files d'attente des nœuds 2 et 3 avec un flux 1-6 avec régulateur de trafic optimisé . . . . .	152
4.31	Evolution des files d'attente des nœuds 2, 3 et 4 avec un flux 1-6 avec régulateur de trafic optimisé . . . . .	152

# Liste des tableaux

1.1	Listes des expérimentations . . . . .	16
1.2	Liste des RTT Mesurés . . . . .	19
1.3	Liste des Giges Mesurés . . . . .	22
1.4	Etat du canal . . . . .	22
3.1	Exemple de configuration des nœuds . . . . .	53
3.2	Exemple de configuration des nœuds (suite) . . . . .	53
3.3	Exemple de matrice de rapport signal sur bruit . . . . .	55
3.4	Exemple de matrice d'atteignabilité . . . . .	55
3.5	Exemple de matrice de chemin . . . . .	56
3.6	Configuration des nœuds du cas d'étude . . . . .	107
3.7	Matrice de rapport signal sur bruit du cas d'étude . . . . .	108
3.8	Matrice de chemins du cas d'étude . . . . .	108
3.9	RTT des requêtes ICMP . . . . .	116
4.1	Nombre et cause des pertes sur chaque lien pour la simulation 1-4 . . . . .	125
4.2	Nombre et cause des pertes sur chaque lien pour la simulation 1-6 . . . . .	127
4.3	Gain de période de contention de 3 nœuds avec la même configuration . . . . .	129
4.4	Gain de période de contention de 5 nœuds avec la même configuration . . . . .	130
4.5	Gain de période de contention de 3 nœuds avec modification de <i>AIFS</i> . . . . .	130
4.6	Gain de période de contention de 3 nœuds avec modification de $CW_{min}$ . . . . .	130
4.7	Nombre et cause des pertes sur chaque lien pour la simulation avec un flux 1-4 et modification du paramètre <i>AIFS</i> sur le nœud source . . . . .	132
4.8	Nombre et cause des pertes sur chaque lien pour la simulation avec un flux 1-6 et modification du paramètre <i>AIFS</i> sur le nœud source . . . . .	134
4.9	Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec modification du paramètre $CW_{min}$ sur le nœud source . . . . .	136
4.10	Nombre et cause des pertes sur chaque lien pour la simulation 1-6 avec modification du paramètre $CW_{min}$ sur le nœud source . . . . .	138
4.11	Récapitulatif des simulations 1-4 . . . . .	138
4.12	Récapitulatif des simulations 1-6 . . . . .	139
4.13	Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec modification du paramètre <i>AIFS</i> sur les nœuds 1 et 2 . . . . .	141
4.14	Récapitulatif des simulations 1-4 en incluant la simulation modifiant le premier relais . . . . .	141

4.15	Nombre et cause des pertes sur chaque lien pour la simulation 1-6 avec modification du paramètre <i>AIFS</i> sur les nœuds 1 et 2 . . . . .	143
4.16	Récapitulatif des simulations 1-6 en incluant la simulation modifiant le premier relais . . . . .	144
4.17	Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec régulateur de trafic . . . . .	148
4.18	Résultats de simulations avec flux 1-4 permettant d'optimiser le régulateur de trafic . . . . .	149
4.19	Nombre et cause des pertes sur chaque lien pour la simulation 1-4 avec régulateur de trafic optimisé . . . . .	150
4.20	Récapitulatif des simulations 1-4 sans puis avec régulateur de trafic optimisé	151
4.21	Nombre et cause des pertes sur chaque lien pour la simulation 1-6 avec régulateur de trafic optimisé . . . . .	153
4.22	Récapitulatif des simulations 1-6 sans puis avec régulateur de trafic . . .	153

# Glossaire

ACK	Acknowledge, Trame qui indique que la trame précédente est reçue correctement
ADSL	Asymmetric Digital Subscriber Line, Technologie de communication utilisant la ligne téléphonique
AIFS	Arbitration IFS, Temps intertrame présenté dans la norme de qualité de service IEEE 802.11e
AIS	Asymmetric Incomplete State, Catégorie d'interaction entre nœud proposé par [RAGK08]
AODV	Ad Hoc On-Demand Vector, Protocole de routage adapté au réseaux mobiles
AP	Access Point, Point d'accès dans un réseau IEEE 802.11 de type infrastructure
BE	Best Effort, Classe de qualité de service proposée dans la norme IEEE 802.11e meilleure que BG
BK	Background, Classe de qualité de service proposée dans la norme IEEE 802.11e de plus faible priorité
BSS	Basic Service Set, Ensemble de stations associées aux points d'accès d'un même réseau
BSSID	Basic Service Set Identifier, Identifiant d'un BSS
CMT	Clustered Multi-channel Two-radio, Approche multicanal proposée par [JR05]
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance, Méthode d'accès basée sur l'accusé de réception
CSMA/CD	Carrier Sense Multiple Access with Collision Detection, Méthode d'accès basée sur la détection de collision
CTS	Clear To Send, Paquet de contrôle permettant de réserver le médium pendant un temps donné
CW	Contention Window, Période de contention tiré aléatoirement dans une fenêtre donnée
DCF	Distributed Coordination Function, Méthode de transmission où un nœud attend que le médium soit libre pour parler
DIFS	DCF IFS, Temps intertrame lors de transmission DCF
DS	Distribution System, Système de distribution sans fil permettant d'interconnecter un BSS



DSR	Dynamic Source Routing, Protocole de routage adapté au réseaux mobiles maillé
DSDV	Destination Sequenced Distance Vector, Protocole de routage adapté au réseaux mobiles
EDCA	Enhanced Distributed Channel Access, Méthode d'accès présenté dans la norme IEEE 802.11e
EIFS	Extended IFS, Temps intertrame succédant un problème de transmission
ESS	Extended Service Set, Ensemble de un ou plusieurs BSS
FSMC	Finite State Markoc Chain, Système mathématique qui décrit des transitions d'un état à un autre
HCMP	Hybrid MultiChannel Protocol, Protocole de gestion multicanal proposé dans [KJCV06]
HTTP	Hypertext Transfer Protocol Secure, Protocole permettant la diffusion de contenu multimedia
IBSS	Independent BSS, Ensemble de stations associés en mode Ad Hoc
ICMP	Internet Control Message Protocol, Protocole de supervision sur IP
IDIS	Interfering Destination Incomplete State, Catégorie d'interaction entre nœud proposé par [RAGK08]
IEEE	Institute of Electrical and Electronics Engineers, Organisme de normalisation
IFS	Inter Frame Space, Temps intertrame
MAC	Media Access Control, Sous couche inférieur de la couche liaison dans le modèle OSI
MAP	Mesh Access Point, MP ayant les fonctionnalités de point d'accès
MCCA	Mesh Coordinated Channel Access, Nouvelle méthode d'accès proposé dans IEEE 802.11s
MCF	Medium Access Coordination Function, Ensemble de fonctions décrivant l'accès au médium
MMAC	Multichannel MAC, Protocole de gestion multicanal proposé par [KJCV06]
MP	Mesh Point, Entité élémentaire d'un réseau de WiFi maillé
MPP	Mesh Portal, MP faisant le lien avec un autre réseau
MRTS	Multicast RTS, Nouveau paquet de contrôle proposé par [JYM07]
NAM	Network Animator, Outils animant les résultats obtenus avec NS2
NS2	Network Simulator 2, Simulateur réseau
OLSR	Optimized Link State Routing, Protocole de routage adapté au réseaux mobiles maillés
OSI	Open Systems Interconnection, Modèle décrivant l'interconnexion de réseaux
PC	Point Coordinator, Élément central en mode PCF
PCF	Point Coordination Function, Méthode de coordination orchestré par un élément central nommé PC
PD	Proportionnel Dérivé, Type de loi de commande automatique
PIFS	PCF IFS, Temps intertrame lors de transmission PCF

PPC	Persistent Pseudo-Clearance, Nom donné dans [JYM07] à un problème d'utilisation des paquets RTS/CTS
QAM	Quadrature Amplitude Modulation, Forme de modulation par modification de l'amplitude en quadrature
QMMN	Queue Management in Multihop Networks, Nouvelle gestion de file d'attente proposé dans [BCG05]
QoS	Quality of Service, Qualité de service
R-CTS	Real-time CTS, Nouveau paquet de contrôle proposé par [SPK <sup>+</sup> 07]
R-RTS	Real-time RTS, Nouveau paquet de contrôle proposé par [SPK <sup>+</sup> 07]
RTP	Real Time Protocol, Protocole utilisé pour transporter des flux audio ou vidéo
RTS	Request To Send, Paquet de contrôle demandant une réservation de médium
RTT	Round Trip Time, Temps aller retour
SCAI	Senders Connected Asymmetric Incomplete State, Catégorie d'interaction entre nœud proposé par [RAGK08]
SCSI	Senders Connected Symmetric Interference, Catégorie d'interaction entre nœud proposé par [RAGK08]
SINR	Signal to Interference Noise Ratio, Rapport signal sur bruit prenant en compte les interférences dans le bruit
SIFS	Short IFS, Temps intertrame très court permettant aux accusé de réception d'être prioritaire
SIS	Symmetric Incomplete State, Catégorie d'interaction entre nœud proposé par [RAGK08]
SIP	Session Initiation Protocol, Protocole de mise en relation initialisant des flux audio ou vidéo
SNR	Signal to Noise Ratio, Rapport signal sur bruit
SPN	Stochastic Petri Nets, Réseaux de Petri stochastiques
SSID	Service Set Identifier, Identifiant d'un réseau WiFi
STA	Station, Entité pouvant se connecter en tant que client sur un point d'accès WiFi en mode infrastructure
TCL	Tool Command Language, Langage informatique permettant de décrire des réseaux dans NS2
TCP	Transmission Control Protocol, Protocole de transport orienté mode connecté
TSAM	Time-Synchronised Application-level MAC protocol, Protocole basé sur TDMA proposé par [KRR <sup>+</sup> 09]
TOS	Type Of Service, Champs indiquant la qualité de service nécessaire dans l'entête IP
UDP	User Datagram Protocol, Protocole de datagramme utilisateur
VI	Video, Classe de qualité de service proposée dans la norme IEEE 802.11e correspondante à la vidéo
VO	Vo-Ip, Classe de qualité de service proposée dans la norme IEEE 802.11e correspondante à la voix sur IP

WDS	Wireless Distribution System, Système de diffusion sans fil interconnecté aussi sans fil
WiFi	Wireless Fidelity, Ensemble de protocoles sans fil régis par la norme IEEE 802.11
WME	Wireless Multimedia Extention, Nom donné à la norme de qualité de service IEEE 802.11e
WMM	Wireless MultiMedia, Nom donné à la norme de qualité de service IEEE 802.11e
WMN	Wireless Mesh Network, Réseau maillé WiFi



# Contribution à l'étude et à l'amélioration de la qualité de service sur un réseau maillé WiFi

Si l'utilisation de voix et vidéo sur ip fonctionne correctement sur un réseau WiFi avec un seul point d'accès, les performances en terme de qualité de service sont altérées sur un réseau WiFi maillé où chaque point d'accès sert de relais aux suivants. En effet, le partage d'un même canal radio sur de grandes surfaces par plusieurs nœuds qui ne se captent pas toujours permet difficilement d'établir des communications nécessitant de la qualité de service.

La difficulté d'une analyse par expérimentation, due à la difficulté d'observation des communications, a orienté l'étude vers la simulation. En l'absence de simulateurs existants adéquats au vu d'un réseau maillé qui est un concept récent, l'objet de cette thèse est la création, l'étude et l'exploitation d'un simulateur, basé sur des réseaux de Petri, permettant la simulation de communications sur un réseau maillé WiFi. Les améliorations apportées par cette étude ont pour but d'être implémentées sur un matériel existant afin d'offrir un service de transport de voix et de vidéo sur un réseau WiFi maillé.

Les premières simulations réalisées à l'aide de cet outil ont fait apparaître des pertes de paquets et des altérations de la gigue due à l'état des files d'attente d'émission de certains nœuds. Après avoir identifié des leviers accessibles sur le matériel existant sur lequel peuvent être effectivement apportées des améliorations tout en restant dans le cadre de la norme, une approche automatique et réseaux a mis en évidence des stratégies d'améliorations possibles de la qualité de service nécessaire pour des communications audio et vidéo.

**Mots clés :** 802.11, WMN, WiFi maillé, Qualité de Service, simulation, réseau de Petri

## Towards Quality of Service Enhancement for Wireless Mesh Network

If the use of voice and video over IP is working properly on a wireless network with only one access point, the QoS (Quality of Service) is altered on a wireless mesh network where each access point relays packets to the next one. Indeed, sharing the same radio channel over a large number of nodes that do not always detect the others makes it difficult to establish communications with good enough QoS.

Because an experimental analysis with communication monitoring is very difficult, we proceeded with a simulation. In lack of adequate simulators because mesh network is a recent concept, the purpose of this thesis is to create, design and use a simulator based on Petri nets, to simulate the wireless mesh network communications. The improvements resulting from this study are intended to be implemented on an existing hardware to provide a transport service for voice and video over a wireless mesh network.

The first simulations performed using this tool pointed out packet loss and jitter due to the lack of management of packet queues within intermediate access points. A modification of a parameter on the existing hardware resulted in a good improvement : an automation and network approach highlighted possible strategies for improvement of QoS required for voice and video communications.

**Keywords :** 802.11, WMN, Wireless Mesh Network, Quality of Service, simulation, Petri Nets