Vers une solution de contrôle d'admission sécurisée dans les réseaux mesh sans fil

Juliette Dromard

Thèse
de doctorat
de l'UTT

Juliette DROMARD

Vers une solution de contrôle d'admission sécurisée dans les réseaux mesh sans fil

Spécialité :
Réseaux, Connaissances, Organisations

2013TROY0028 **Année 2013**

utt
université de technologie
Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : RESEAUX, CONNAISSANCES, ORGANISATIONS

présentée et soutenue par

Juliette DROMARD

le 6 décembre 2013

Vers une solution de contrôle d'admission sécurisée dans les réseaux mesh sans fil

JURY

M. B. COUSIN	PROFESSEUR DES UNIVERSITES	Président
M. M. Y. GHAMRI-DOUDANE	PROFESSEUR DES UNIVERSITES	Examinateur
M. R. KHATOUN	ENSEIGNANT CHERCHEUR UTT	Directeur de thèse
M. L. KHOUKHI	ENSEIGNANT CHERCHEUR UTT	Directeur de thèse
M. P. LORENZ	PROFESSEUR DES UNIVERSITES	Rapporteur
M. A. SERHROUCHNI	PROFESSEUR TELECOM PARISTECH	Rapporteur

Résumé

Les réseaux mesh sans fil (Wireless Mesh Networks-WMNs) sont des réseaux facilement déployables et à faible coût qui peuvent étendre l'Internet dans des zones où les autres réseaux peuvent difficilement accéder. Cependant, des problèmes de qualité de service (QoS) et de sécurité freinent le déploiement à grande échelle des WMNs. Dans cette thèse, nous proposons une solution de contrôle d'admission (CA) et un système de réputation afin d'améliorer les performances du réseau mesh et de le protéger des nœuds malveillants. Notre système de CA vise à assurer la qualité de service des flux admis dans le réseau en termes de bande passante et de délai tout en maximisant l'utilisation de la capacité du canal. L'idée de notre solution est d'associer au contrôle d'admission une planification de liens afin d'augmenter la bande passante disponible. Nous proposons également un système de confiance ayant pour but de détecter les nœuds malveillants et de limiter les fausses alertes induites par la perte de paquets sur les liens du réseau. L'idée de cette solution est d'utiliser des tests statistiques comparant la perte de paquets sur les liens avec un modèle de perte préétabli. De plus, notre proposition comprend un système de surveillance composé de plusieurs modules lui permettant de détecter un grand nombre d'attaques. Notre CA a été validé via de nombreuses simulations sur le simulateur ns-2. Les résultats montrent qu'il permet de respecter les exigences des flux en termes de bande passante et de délai et aussi d'améliorer la capacité totale du réseau. Notre système de réputation a été validé via le logiciel de simulation R: les résultats prouvent qu'il permet de diminuer les faux positifs par rapport aux solutions existantes avec d'autant plus d'efficacité que la variance de la perte de paquets des liens du réseau est faible. Ces deux solutions ont été conçues dans la perspective d'une fusion, pour par la suite proposer une solution de contrôle d'admission sécurisée dans un réseau mesh sans fil.

Abstract

Wireless mesh networks (WMNs) are a very attractive new field of research. They are low cost, easily deployed and high performance solutions to last mile broadband Internet access. However, they have to deal with security and quality of service issues which prevent them from being largely deployed. In order to overcome these problems, we propose in this thesis two solutions: an admission control with links scheduling and a reputation system detecting bad nodes. These solutions have been devised in order to further merge into a secure admission control. Our admission control schedules dynamically the network's links each time a new flow is accepted in the network. Its goal is to accept only flows whose constraints in terms of delay and bandwidth can be respected, increasing the network capacity and decreasing the packet loss. Our reputation system aims at assigning each node of the network a reputation which value reflects the real behavior of the node. To reach this goal this reputation system is made of a monitoring tool which can monitor many types of attacks and consider the packet loss in the network. The evaluations of our solutions show that they both meet their objectives in terms of quality of service and security.

A mes parents et ma grand-mère,

Remerciements

Mes remerciements vont d'abord à mes deux co-directeurs de thèse, Rida Khatoun et Lyes Khoukhi, tous les deux enseignants-chercheurs à l'UTT pour leur encadrement, leur disponibilité, leur patience et leurs précieux conseils.

Je remercie également l'ensemble de l'équipe d'environnement et de réseaux autonomes de l'UTT pour m'avoir accueillie parmi eux si chaleureusement.

Je remercie, mes collègues de bureau, Badis Hammi, Charles Perez, Nathalie Matta, et Karina Perez de m'avoir supportée aussi bien dans mes moments de stress que de perte de moral et pour les nombreux moments de partage et de discussions que l'on a pu avoir.

Je remercie Pascal Lorenz, Professeur des Universités à Nancy et Ahmed Serhrouchni, Professeur à Télécom ParisTech, pour avoir accepté d'être les rapporteurs de ma thèse.

Je remercie Bernard Cousin, Professeur des Universités à Rennes et Yacine Ghamri-Doudane Professeur des Universités à La Rochelle pour avoir accepté d'être les examinateurs de ma thèse.

Et je ne remercierai jamais assez mes parents, qui m'ont guidée, soutenue et encouragée pendant toutes ces nombreuses années qui n'ont pas toujours été très faciles. Et finalement mes pensées les plus tendres vont à Mamu, le magicien qui m'a permis de garder le sourire pendant cette difficile mais passionnante épopée qu'est la thèse.

Sommaire

RESUME 3

ABSTRACT 5

REMERCIEMENTS 9

LISTE DES FIGURES 15

LISTE DES TABLEAUX 19

CHAPITRE 1. INTRODUCTION GENERALE 1
1.1. Contexte 1
1.2. Problématique 2
1.3. Contribution 3
1.4. Organisation du document 5
CHAPITRE 2. LES RESEAUX MESH SANS FIL 9
2.1. Introduction 9
2.2. Présentation, objectifs des réseaux mesh sans fil 9
2.2.1. Les protocoles de routage dans les réseaux mesh 11
2.2.2. L'objectif des réseaux mesh 15
2.3. Applications des réseaux mesh sans fil 15
2.4. Limitations des réseaux mesh sans fil 18
2.5. Conclusion 19

PARTIE 1. UN SYSTEME DE CONTROLE D'ADMISSION AVEC PLANIFICATION DES LIENS DANS UN RESEAU MESH SANS FIL 21

PARTIE 1.INTRODUCTION 23
CHAPITRE 3. ETAT DE L'ART : CONTROLE D'ADMISSION ET PLANIFICATION DES LIENS 25
3.1. Introduction 25
3.2. Contrôle d'admission 26
3.2.1. Défis d'un contrôle d'admission 26
3.2.2. Solutions existantes de contrôle d'admission 31
3.2.3. Limites des solutions existantes de contrôle d'admission 41
3.3. La planification de liens 42
3.3.1. Présentation, objectifs et challenges de la planification de liens 43
3.3.2. Solutions existantes de planification des liens 47
3.3.3. Limitations des solutions existantes de planification des liens 53
3.4. Conclusion 54
CHAPITRE 4. CONTROLE D'ADMISSION AVEC PLANIFICATION DES LIENS 57
4.1. Introduction 57

4.2. Objectifs de notre contrôle d'admission avec planification des liens dans un réseau mesh 57
4.3. Modélisation du réseau et problématique 59
4.3.1. Découpage du temps et modélisation de l'interférence 61
4.3.2. Les flux admis dans le réseau 66
4.3.3. La bande passante et la planification d'un flux 67
4.3.4. Le délai d'un flux 70
4.3.5. Formulation du problème d'admission d'un flux 76
4.4. Notre solution de contrôle d'admission avec planification des liens 86
4.4.1. Algorithme d'admission et de planification d'un nouveau flux 86
4.4.2. Contrôle d'admission d'un flux 93
4.4.3. Envoi d'une requête d'admission d'un flux 93
4.4.4. Procédure d'admission ou de rejet du flux 94
4.4.5. Envoi de la réponse d'admission ou de rejet d'un flux 94
4.5. Evaluation du contrôle d'admission avec planification des liens 95
4.6. Conclusion 101
PARTIE 1. CONCLUSION 103

PARTIE 2. NOUVEAU SYSTEME DE CONFIANCE AVEC DETECTION DES MAUVAIS NŒUDS DANS UN RESEAU MESH SANS FIL 105

PARTIE 2. INTRODUCTION 107
CHAPITRE 5. LES MODELES DE CONFIANCE 109
5.1. Introduction 109
5.2. Objectifs et fonctionnement des modèles de confiance dans les réseaux mesh sans fil 110
5.3. Modèles de confiance existants dans les réseaux mesh 113
5.4. Conclusion sur l'état de l'art des systèmes de confiance dans les réseaux mesh 120
Conclusion 122
CHAPITRE 6. NOUVEAU SYSTEME DE CONFIANCE DE DETECTION DES MAUVAIS NŒUDS 125
6.1. Introduction 125
6.2. Objectifs 125
6.3. Fonctionnement du système de confiance 128
6.3.1. Modélisation de la perte de paquets 128
6.3.2. Système de détection multiple des mauvais nœuds 136
6.3.3. Système de calcul de la confiance 142
6.4. Evaluation du système de confiance 144
6.4.1. Evaluation du premier et du second module de l'IDS de notre système de confiance 145
6.4.2. Evaluation du troisième module de l'IDS de notre système de confiance 150
6.4.3. Evaluation comparative de notre système de confiance avec un système de confiance existant 153
6.5. Conclusion 158
PARTIE 2. CONCLUSION 161
CHAPITRE 7. CONCLUSION GENERALE ET PERSPECTIVES 163
7.1. Contribution 163

7.1.1. Contrôle d'admission avec planification des liens 163
7.1.2. Nouveau système de confiance avec détection des mauvais nœuds dans un réseau mesh 164
7.2. Perspectives 165
7.2.1. Etudes d'évaluation approfondie 165
7.2.2. Intégration de notre système de confiance au contrôle d'admission 165
7.2.3. Différenciation de la confiance 166

PUBLICATIONS 167

BIBLIOGRAPHIE 169

LISTE DES ABRÉVIATIONS 177

Liste des figures

Figure 1 : Fonction de répartition du débit de flux TCP selon la distance entre leur nœud source et la destination, sachant qu'une seule session TCP est active à la fois. Chaque flux est émis à 5Mbit/s (Das, Koutsonikolas, & Hu, 2008). 3

Figure 2: Exemple d'un réseau sans fil à un saut.....9

Figure 3: Exemple d'un réseau 11

Figure 4 : Broadcaste d'un RREQ par la source S, rediffusé par tous les nœuds qui le reçoive.....12

Figure 5 : envoi d'un RREP par la destination D le long du chemin emprunté par le RREQ.....13

Figure 6: I-STAMS permet de monter rapidement des réseaux mesh dans le domaine militaire (I-STAMS, 2013) 16

Figure 7 : Un réseau mesh sans fil déployé dans un train (Akyildiz, Wang, & Wang, 2005) 17

Figure 8 : Les différentes zones d'un nœud d'après la norme IEEE 802.11 27

Figure 9 : Protocole DCF 28

Figure 10 : Classification des protocoles de contrôle d'admission.....29

Figure 11 : C est un nœud caché de la communication entre A et B.....32

Figure 12 : Phénomène de contention intra-route; le flux est émis par cinq nœuds A, B, C, D, E appartenant à la zone d'écoute de C. L'émission du flux par les nœuds A, B, C, D, E diminuent la bande passante disponible du nœud C. 33

Figure 13 : Blocage du nœud E.....34

Figure 14 : Zone d'écoute et zone d'écoute élargie d'un nœud 38

Figure 15 : les protocoles d'interférences ont pour but de prédire si le nœud $r1$ reçoit avec succès le signal en provenance du nœud $u1$ sachant que l'ensemble des nœuds activés simultanément est $\Gamma = \{u1, u2, u3, u4, u5, u6\}$.....45

Figure 16 : Décomposition du temps en fenêtres de planification, chaque fenêtre est décomposée en N slots avec N=10 48

Figure 17 : Division de l'espace de l'ensemble Ri. Un lien par carré jaune est associé à un même slot. 51

Figure 18 : Graphe complet étiqueté et orienté $G(V, E, f)$ 60

Figure 19 : Division du temps en fenêtres de planification composées de 10 slots ($N = 10$) dont 2 sont réservés à l'accès à compétition au canal($Nc = 2$) et 8 à l'accès temporel au canal ($Np = 8$)...62

Figure 20: Trois situations où le lien (A,B) est en conflit primaire 63

Figure 21 : Trois nœuds envoient un paquet de données sur un même slot.....64

Figure 22 : Le nœud e_i a réservé trois slots pour le flux f par fenêtre de planification 67

Figure 23 : Le nœud B émet le paquet en provenance du nœud A au cours de la fenêtre de planification où il a reçu le paquet 71

Figure 24 : Le nœud B émet le paquet en provenance de A au cours de la fenêtre de planification suivant celle où il a reçu le paquet. 71

Figure 25 : Illustration du phénomène de *slots perdus*, lorsqu'un nœud ne peut émettre de paquet lors d'un slot réservé car il n'en possède aucun en attente 72

Figure 26 : Le problème A qui est NP-complet est transformable en un problème B via une fonction polynomiale f. Si B est également NP alors le problème B est NP-complet. 81

Figure 27 : Graphe biparti planaire extérieur. Chaque arête a un sommet carré et un sommet rond ; le graphe est donc bien biparti. 83

Figure 28 : Arborescence de la méthode de Dakin 89

Figure 29 : Algorithme de Dakin modifié, qui retourne une solution au PLVB d'entrée. Si l'algorithme retourne null, alors il n'a pas de solution. 91

Figure 30 : Approche itérative de l'algorithme d'admission et de planification d'un flux 92

Figure 31 : Topologie en croix 95

Figure 32 : Topologie maillée 95

Figure 33 : Topologie chainée 95

Figure 34: Débit des flux en kbit/s dans le temps (seconde) avec notre modèle sur la topologie chainée 96

Figure 35 : Débit des flux en kbit/s dans le temps (seconde) avec le modèle originel sur la topologie chainée 96

Figure 36 : Débit des flux en kbit/s dans le temps (seconde) avec notre modèle sur la topologie en croix 97

Figure 37: Débit des flux en kbit/s dans le temps (seconde) avec le modèle originel sur la topologie en croix 97

Figure 38 : Débit des flux en kbit/s dans le temps (seconde) avec notre modèle sur un réseau à topologie maillée 98

Figure 39 : Débit des flux en kbit/s dans le temps (seconde) avec le model originel sur un réseau à topologie maillée 98

Figure 40 : Comparaison du débit des flux du réseau à topologie chainée 99

Figure 41 : Comparaison du débit des flux pour la topologie en croix 99

Figure 42 : Comparaison du débit des flux pour la topologie maillée 99

Figure 43 : Délai des flux dans le réseau mesh à topologie maillée 100

Figure 44 : Délai des flux dans le réseau mesh à topologie en croix 100

Figure 45 : Délais des flux dans un réseau mesh à topologie chainé 100

Figure 46 : Architecture de base d'un nœud dans un système de réputation 112

Figure 47 : Illustration de *Watchdog* : A envoie un paquet à B et vérifie que B le fait suivre à C en écoutant les paquets que B envoie 114

Figure 48 : Extension de *Watchdog* 116

Figure 49 : S envoie une requête de route pour établir une route jusqu'au nœud D. Le nœud voisin A ajoute au RREQ la confiance αA qu'il a en S. Chaque noeud intermédiaire non voisin de la source récupère la valeur de confiance que A a en la source S lorsqu'il reçoit la requête de route. 117

Figure 50 : Perte de paquet sur le lien (A,B) 126

Figure 51 : Perte de paquet sur le lien (B,A) 126

Figure 52 : Distribution de la probabilité de perte de paquets. Chaque point correspond à une paire émetteur/récepteur à un débit particulier. Seuls les pairs émetteur/récepteur qui ont envoyé au moins un paquet avec succès pendant l'expérience, sont représentées sur la figure (Aguayo, Bicket, Biswas, Judd, & Morris, 2004). 127

Figure 53 : Probabilité de perte de paquets sur les liens 129

Figure 54: Le nœud i envoie m paquets à j, $\mathrm{K1}$ sont perdus sur le lien (i,j). Le nœud js'il est non mauvais envoie $\mathrm{m}-\mathrm{K1}$ paquets au nœud k et le nœud i entend seulement $m-\mathrm{K1}-\mathrm{K2}$ d'entre eux 130

Figure 55 : Valeur de p retournée par chacun des 100 KS-tests. Chaque KS-test compare un échantillon de n réalisation de la variable X_{ij}^{o} avec une loi normale 132

Figure 56: Carte du premier étage du bâtiment 133

Figure 57 : $p-value$ pour chaque KS-test effectué sur des échantillons de données de différentes tailles m récoltées sur le lien 1 135

Figure 58 : $p-value$ pour chaque KS-test effectué sur des échantillons de données de différentes tailles m récoltées sur le lien 2 135

Figure 59 : $p-value$ pour chaque KS-test effectué sur des échantillons de données de tailles m récoltées sur le lien 3 135

Figure 60: Carte de contrôle établie par la méthode CUSUM sur un échantillon de 40 données 142

Figure 61 : Système de confiance installé sur chaque nœud du réseau mesh afin d'établir la confiance qu'il a en chacun de ses voisins 144

Figure 62 : Pourcentage de faux négatifs et de faux positifs lorsque les échantillons sont de taille n et que l'écart-type de la loi normale tronquée en 0 et en 1 générant les échantillons est de 0.1 ($\sigma_{\mathrm{ij}}^{\mathrm{o}}=0.1$). 146

Figure 63 : Pourcentage de faux négatifs et de faux positifs lorsque les échantillons sont de taille n et que l'écart-type de la loi normale tronquée en 0 et en 1 générant les échantillons est de 0.5 ($\sigma_{ij}^{o}=.5$). 147

Figure 64 : Pourcentage de faux négatifs et de faux positifs lorsque les échantillons sont taille n et que l'écart-type de la loi normale tronquée en 0 et en 1 générant les échantillons est de 0.9 ($\sigma_{ij}^{o}=0{,}9$) 147

Figure 65 : Pourcentage d'acceptation de l'hypothèse $H0$ lorsque σ_{ij}^{o} ==0.1 et $n=50$ 148

Figure 66 : Taux de faux négatifs et faux positifs lorsque σ_{ij}^{o}=0.1 et $n = 50$ 148
Figure 67 : Pourcentage d'acceptation de l'hypothèse $H0$ lorsque σ_{ij}^{o}=0.5 et $n = 250$ 149
Figure 68 : Taux de faux négatifs et faux positifs lorsque σ_{ij}^{o} =0.5 et $n = 250$ 149
Figure 69 : Pourcentage d'acceptation de l'hypothèse $H0$ lorsque σ_{ij}^{o}=0.9 et $n = 300$ 150
Figure 70 : Taux de faux négatifs et faux positifs lorsque σ_{ij}^{o} =0.9 et $n = 300$ 150
Figure 71 : Pourcentage d'alertes ave CUSUM lorsque σ_{ij}^{o} =0.1 151
Figure 72 : Taux de faux négatifs et faux positifs du module 3 lorsque σ_{ij}^{o} =0.1 151
Figure 73 : Pourcentage d'alertes ave CUSUM lorsque σ_{ij}^{o} =0,5 152
Figure 74 : Taux de faux négatifs et faux positifs du module 3 lorsque σ_{ij}^{o}=0,5 152
Figure 75 : Pourcentage d'alertes avec CUSUM lorsque σ_{ij}^{o}=0.9 153
Figure 76 : Taux de faux négatifs et faux positifs du module 3 lorsque σ_{ij}^{o} =0.9 153
Figure 77 : Topologie en croix 154
Figure 78 : Topologie maillée 154
Figure 79 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=1 155
Figure 80 : Confiance des nœuds dans un réseau mesh à topologie en croix avec notre solution lorsque p =1 155
Figure 81 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=0.5 156
Figure 82 : Confiance des nœuds dans un réseau mesh à topologie en croix avec notre solution lorsque p =0.5 156
Figure 83 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=0.2 157
Figure 84 : Confiance des nœuds dans un réseau mesh à topologie en croix avec notre solution lorsque p =0.2 157
Figure 85 : Confiance des nœuds dans un réseau mesh à topologie maillé avec la solution de référence lorsque p=0.2 157
Figure 86 : Confiance des nœuds dans un réseau mesh à topologie maillée avec notre solution lorsque p =0.2 157

Liste des tableaux

Tableau 1 : Comparaison entre protocoles de routage réactif et proactif.....14

Tableau 2 : Sensibilité des applications à différentes métriques de QoS (Khoukhi, 2006) 25

Tableau 3: Comparaison de protocoles de contrôle d'admission existants 41

Tableau 4 : Tableau comparatif de systèmes de planification existants.....53

Tableau 5 : Description des paramètres nécessaires au calcul du temps d'émission d'un paquet de données dans un réseau mesh à accès temporel.....63

Tableau 6 : Tableau de description des paramètres.....78

Tableau 7 : Tableau des paramètres de simulation du contrôle d'admission avec planifications des liens.....96

Tableau 8 : Tableau de comparaison des solutions de système de confiance existantes 121

Tableau 9 : Paramètres de l'expérience 134

Tableau 10 : Performances des modules 1 et 2 selon l'écart-type de la perte de paquets 150

Tableau 11 : Table des résultats de validation du troisième module de l'IDS de notre système de confiance 153

Tableau 12 : Tableau des paramètres de simulation.....154

Chapitre 1. Introduction Générale

1.1. Contexte

Depuis le début des années 2000 et l'introduction des Smartphones et des tablettes, les utilisateurs de réseau deviennent de plus en plus mobiles et aspirent à pouvoir se connecter en permanence. Or, les réseaux filaires et sans fil avec infrastructure utilisés actuellement (WiFi, 3G, etc) possèdent une accessibilité et couverture limitée. En effet, les utilisateurs doivent avoir accès à un port pour se connecter à un réseau filaire et être dans la couverture du point d'accès pour accéder à un réseau sans fil à infrastructure non multi-sauts. De plus, tous ces réseaux nécessitent une infrastructure lourde et le déploiement de ces réseaux peut être très couteux: par exemple l'installation d'une ligne numérique d'abonnée (DSL-Digital Subscribe Line) requiert plusieurs mois (Sichitiu 2005).

Les réseaux ad-hoc multi-sauts permettent de résoudre certaines limites des réseaux existants, car ils ne possèdent pas ou peu d'infrastructures, ont un faible coût, sont rapides à déployer et peuvent étendre le réseau dans des zones encore non couvertes telles que des zones accidentées ou isolées. Ces réseaux sont appelés multi-sauts car ils transfèrent les données des utilisateurs de routeurs sans fil en routeurs sans fil. Ad-hoc est une locution latine signifiant « pour cela » car ces réseaux, à l'origine, étaient souvent créés de manière spontanée par des utilisateurs se connectant les uns aux autres temporairement et dans un but précis.

On distingue quatre principales familles de réseaux ad-hoc multi-sauts ; les réseaux mesh (Nandiraju, et al. 2007), les réseaux ad-hoc (Sivakumar, Suseela et Varadharajan 2012), les réseaux de capteurs et les VANETs (Vehicular Ad-hoc Networks) (Hartenstein et Laberteaux 2008) (Bakhouya, Gaber et Lorenz 2011). Les réseaux de capteurs sont des réseaux à faible énergie et faible puissance de calcul et ne permettent donc pas de répondre aux attentes des utilisateurs en termes de rapidité et performance. Les réseaux VANETs sont des réseaux véhiculaires qui permettent de guider et conseiller le conducteur, sa grande mobilité entraine des ruptures fréquentes de routes et de connexions. Les réseaux ad-hoc sont généralement mobiles et souvent désignés sous le terme de MANETs (Mobile Ad-hoc Networks). Ces derniers sont formés à partir des équipements de chaque utilisateur qui jouent à la fois le rôle de clients et de serveurs. Comme les réseaux VANETs, mais dans une moindre mesure, les utilisateurs des MANETs sont mobiles et se déconnectent régulièrement, les réseaux ad-hoc sont donc peu fiables, les routeurs composant les chemins pouvant disparaitre à tout moment coupant alors des connexions en cours.

Les réseaux mesh par contre, sont beaucoup plus fiables car ils sont composés d'entités fixes totalement dédiées à la tache de routage. De plus, ces réseaux permettent facilement d'interconnecter de nombreux réseaux d'accès sans fil existants comme la 3G, le Wifi et le WiMAX, le bluetooth (Iyer, Rosenberg et Karnik 2009). En effet, les routeurs mesh sans fil possèdent plusieurs cartes réseaux et peuvent permettre à différents types de réseaux d'accès de se connecter au WMN. Ainsi, les réseaux mesh permettent de répondre aux attentes des utilisateurs en apportant la connexion là où elle est inexistante, rapidement et à faible coût (Nandiraju, et al. 2007) tout en permettant l'interconnexion d'équipements aux technologies différentes.

1.2. Problématique

Les réseaux mesh permettent de connecter des zones encore blanches et de déployer l'Internet dans des domaines et des situations où les réseaux actuels font défaut. Cependant, ces réseaux souffrent actuellement d'une faible bande passante (Nandiraju, et al. 2007) et d'une qualité de service (QoS) médiocre qui limitent leur déploiement. En effet, les applications réseaux sont soumises à de nombreuses contraintes en termes de débit, délai, gigue, perte de paquets (voix sur IP, jeux en ligne, vidéo à la demande) et de sécurité (paiement en ligne, applications de stockage de données privées en ligne, gestion des comptes). En termes de qualité de service, de nombreuses études (Das, Koutsonikolas et Hu 2008) (Cheng, Prasant et Lee 2008) (Reis, et al. 2006) montrent que les réseaux mesh souffrent d'un taux de perte de paquets important (Das, Koutsonikolas et Hu 2008), d'une faible capacité (Nandiraju, et al. 2007), d'une iniquité entre les nœuds (Abouaissa, Brahmia et Lorenz 2013) et d'interférences. Par exemple, les résultats d'une expérience menée sur le campus de l'université de Purdue montrent que le débit d'un flux chute dés que son nœud source est situé à plus d'un saut de la destination (voir figure 1) et que la probabilité de perte d'un paquet est d'au moins 50% lorsque le nœud émetteur est situé à au moins trois sauts de sa destination (Das, Koutsonikolas et Hu 2008).

Les réseaux mesh sont également particulièrement vulnérables aux attaques internes et externes (Nandiraju, et al. 2007). Le canal d'un réseau mesh étant sans fil, un attaquant se situant sur la zone d'écoute peut facilement écouter ou brouiller tout ce qui passe sur le réseau. De plus, les routeurs mesh sont généralement physiquement mal protégés et situés dans des lieux public, il est alors aisé pour un attaquant de capturer le routeur, le modifier ou le remplacer. Les routeurs mesh sont également des équipements à bas coûts et les attaquants peuvent profiter de leurs failles logicielles pour les contrôler (Naouel Ben et Hubaux 2006). Il suffit qu'un seul routeur soit sous le contrôle d'un

attaquant pour que l'ensemble du réseau s'effondre, ce dernier peut alors envoyer de fausses annonces de routage, perturber tous les flux qu'il relaie, inonder le réseau de paquets, etc (Nandiraju, et al. 2007). Les réseaux mesh offrent donc de nombreuses voies d'accès pour les attaquants.

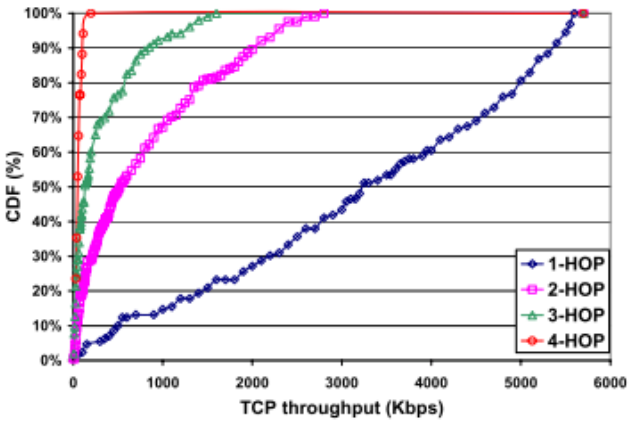


Figure 1 : Fonction de répartition du débit de flux TCP selon la distance entre leur nœud source et la destination, sachant qu'une seule session TCP est active à la fois. Chaque flux est émis à 5Mbit/s (Das, Koutsonikolas et Hu 2008).

Pour permettre le déploiement des réseaux mesh à large échelle, il est donc indispensable de réfléchir à ces réseaux en termes de qualité de service et sécurité afin de proposer des solutions permettant en termes de QoS de :

- respecter les contraintes des flux du réseau afin d'y utiliser des applications aux exigences strictes,
- améliorer la capacité utile du réseau et diminuer la perte de paquets.

Ces solutions doivent également avoir pour objectifs en termes de sécurité de :

- détecter les mauvais nœuds, qu'ils soient internes ou externes au réseau.
- protéger le réseau de l'action des mauvais nœuds.

1.3. Contribution

Afin de répondre à la problématique posée dans la section précédente, nous avons étudié au cours de cette thèse de nombreuses solutions existantes de qualité de service et de sécurité dans les

réseaux mesh. Suite à cette étude, nous avons proposé deux solutions : l'une de qualité de service et la seconde de sécurité. Elles ont été conçues et réfléchies dans l'optique d'une future fusion pour réaliser une architecture de qualité de service sécurisée pour les réseaux mesh sans fil. Dans cette thèse nous proposons donc deux solutions.

La première est un contrôle d'admission avec planifications des liens. Cette solution contribue à poser un cadre formel pour le contrôle d'admission avec planification de liens puisqu'elle présente le problème d'admission d'un flux sous la forme d'un programme linéaire à variables binaires et repose sur une nouvelle méthode de calcul du délai des flux. Notre proposition intègre un contrôle d'admission à la planification de liens, afin que la planification des liens évolue à l'admission de chaque nouveau flux sur le réseau. Les résultats de validation montrent que notre solution atteint ses objectifs puisqu'elle permet de respecter les exigences des flux admis sur le réseau, diminue la perte de paquets et augmente la capacité utile du réseau.

La seconde proposition est un système de confiance qui a pour but, par la suite, de sécuriser le contrôle d'admission présenté ci-dessus. Ce système de confiance considère la perte de paquets sur les liens du réseau et plusieurs modes d'attaques. Il a pour objectif d'attribuer une valeur de confiance à chaque nœud du réseau reflétant son comportement réel. Il comporte un système de surveillance composé de trois modules, chaque module permet de détecter un mode d'attaque. Le premier module permet de détecter les nœuds qui s'inscrivent sur un maximum de routes mais ne font pas suivre les paquets de données afin de perturber les transmissions. Le second module surveille les nœuds qui envoient toujours des acquittements afin de ne pas attirer l'attention des nœuds avoisinants et ne font pas suivre correctement les paquets de données. Le troisième module permet de détecter le changement de comportement des nœuds, par exemple lorsque le nœud passe d'un état honnête à malveillant, ou de non défaillant à défaillant. De plus, afin que les informations collectées par les nœuds sur leurs voisins ne soient pas biaisées par la perte de paquets chaque module d'un nœud vérifie si les données collectées sur le voisin du nœud suivent une distribution « normale » de perte de paquets ou non via des tests statistiques. Si elles ne suivent pas une distribution « normale », le module détecte que le voisin du nœud abandonne certains paquets et est donc mauvais. Ainsi, un module est capable de différencier les mauvais des bons nœuds, même en présence de perte de paquets et pour divers modes d'attaques. Les résultats de la validation de notre solution montrent que nos modules peuvent détecter avec un faible taux de faux positifs et de faux négatifs divers types de mauvais nœuds en présence de perte de paquets sur les liens. Dans le pire des cas les pourcentages de faux négatifs et faux positifs peuvent atteindre 12% tandis que dans les cas les plus favorables ces pourcentages ne dépassent pas les 3%. Ainsi la valeur de confiance attribuée à un nœud est non biaisée par la perte de paquets sur ses liens et elle reflète

le comportement réel de ce dernier. Par la suite, nous souhaitons intégrer dans le contrôle d'admission d'un flux la valeur de confiance des nœuds, afin qu'un flux ne soit admis sur un chemin que si ce dernier répond à un ensemble de critères de qualité de service et de sécurité.

1.4. Organisation du document

Cette thèse est organisée en deux parties précédées d'un chapitre présentant le contexte de la thèse et les réseaux mesh sans fil. La première partie du document aborde la problématique de la qualité de service dans les réseaux mesh sans fil. Elle contient un chapitre sur l'état de l'art du contrôle d'admission et de la planification des liens, ainsi qu'un chapitre sur notre proposition de contrôle d'admission avec planification des liens dans un réseau mesh. La seconde partie du document aborde la problématique de la sécurité dans les réseaux mesh : elle contient un chapitre état de l'art sur les systèmes de confiance dans les réseaux mesh et un chapitre sur notre système de confiance et son évaluation.

Chapitre 2. Les réseaux mesh sans fil

Ce chapitre présente les réseaux mesh sans fil, leurs objectifs, leurs applications et leurs limites. Il permet d'introduire le réseau cible de la thèse ainsi que les motivations de ce choix (faible coût, facilité de déploiement, connexion de zones isolées, etc). Les limites actuelles des réseaux mesh sans fil sont également développées telles que le non-respect des contraintes des flux, la faible capacité du réseau (Nandiraju, et al. 2007), les problèmes de sécurité, des limites auxquelles cette thèse propose des solutions.

Partie 1. Un système de contrôle d'admission avec planification des liens dans un réseau mesh sans fil

Chapitre 3. Etat de l'art : Contrôle d'admission et planification des liens

Le chapitre précédent présente les réseaux mesh et souligne l'une de leurs principales limites, une bande passante restreinte. La faible capacité de ces réseaux entraine régulièrement le non-respect des contraintes d'applications aux exigences strictes telles que la VoIP, le streaming, la visioconférence, les jeux vidéo en réseau, et dès lors la non satisfaction des utilisateurs. Il est donc indispensable de doter ces réseaux de mécanismes de qualité de service afin de satisfaire les exigences des utilisateurs pour qu'ils puissent avoir un accès fluide à leurs différentes applications.

Ce chapitre présente un état de l'art sur deux solutions de QoS complémentaires: le contrôle d'admission, qui permet de garantir les contraintes des flux, et la planification des liens qui a pour principal objectif d'augmenter la bande passante du réseau. Ces mécanismes seront exploités au cours de cette thèse pour leurs nombreux atouts.

Chapitre 4. Contrôle d'admission avec planification des liens

Le chapitre précédent présente comment les solutions de contrôle d'admission et de planification de liens permettent d'améliorer la qualité de service d'un réseau mesh. Cependant, il souligne également leurs limites en mettant en évidence le nombre restreint de flux que les contrôles d'admission peuvent accepter à cause de la capacité limitée d'un réseau mesh. Il soulève également le problème des contrôles d'admission existants en termes de sous-estimation ou surestimation des ressources du réseau pouvant entrainer respectivement soit une perte des ressources soit une congestion du réseau. Il montre que les solutions de planifications de liens existantes ne s'adaptent généralement pas à la charge du réseau et limitent sa dynamique. Ce chapitre introduit notre contrôle d'admission avec planification des liens qui a pour objectifs principaux de respecter les contraintes des flux, d'augmenter la capacité du réseau et ainsi le nombre de flux admis. Notre solution propose de recalculer la planification des liens à chaque admission d'un nouveau flux. Ainsi, le système de planification évolue avec la charge du réseau, et le contrôle d'admission profite de l'augmentation de la bande passante utile et de l'estimation rigoureuse des ressources du réseau induite par la planification. Le chapitre se termine sur l'évaluation de notre solution.

Partie 2. Nouveau système de confiance avec détection des mauvais nœuds dans un réseau mesh sans fil

Chapitre 5. Les modèles de confiance

Le chapitre 5 présente un état de l'art sur les modèles de confiance dans les réseaux mesh, il liste leurs objectifs, décrit leurs structures, et introduit des modèles de confiance existants et leurs limites.

Chapitre 6. Nouveau système de confiance de détection des mauvais nœuds

Ce chapitre présente notre nouveau système de confiance dans un réseau mesh sans fil. Notre solution attribue à chaque nœud une valeur de confiance reflétant son comportement (mauvais ou bon). Notre système a pour objectif de pallier aux limites des solutions de confiance existantes qui

considèrent généralement un seul type de mauvais comportement et qui ne prennent pas en compte le bruit lors de la détection des mauvais nœuds. Notre système de confiance intègre un système de surveillance comprenant trois modules, chacun détectant un type de mauvais comportement et intégrant une méthode statistique afin de déterminer si les données observées proviennent d'un nœud mauvais ou d'un nœud qui subit une perte sur son réseau. Ce système de surveillance permet de différencier les mauvais des bons nœuds sur un réseau subissant une forte perte de paquets. Le chapitre se termine sur l'évaluation de notre solution qui montre qu'elle peut détecter trois types d'attaques différentes en présence de bruit sur le réseau, avec un faible taux de faux négatifs et de faux positifs.

Chapitre 7. Conclusions générale et perspectives

Dans ce chapitre, nous concluons la thèse suivant deux sections: la première section résume notre contribution et montre comment elle répond à notre problématique de départ et la seconde section introduit les axes d'amélioration et d'extension de cette thèse.

Chapitre 2. Les réseaux mesh sans fil

2.1. Introduction

Ce chapitre présente les réseaux mesh sans fil, leurs objectifs, leurs applications et leurs limites. Il permet d'introduire le réseau cible de la thèse ainsi que les motivations de ce choix (faible coût, facilité de déploiement, connexion de zones isolées, etc). Les limites actuelles des réseaux mesh sans fil sont également développées telles que le non-respect des contraintes des flux, la faible capacité du réseau (Nandiraju, et al. 2007), les problèmes de sécurité ; des limites auxquelles cette thèse propose des solutions.

2.2. Présentation, objectifs des réseaux mesh sans fil

Les réseaux sans fil font partie aujourd'hui de la vie quotidienne. Cependant, actuellement, l'accès au réseau sans fil s'effectue généralement à un saut : l'utilisateur étant directement connecté à un point d'accès qui transmet ensuite ses données en filaire (voir figure 2).

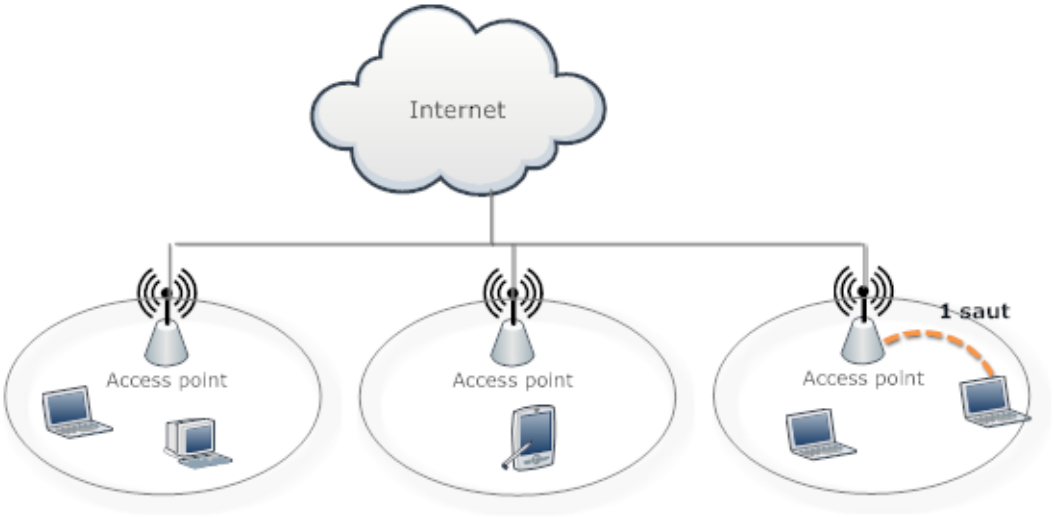


Figure 2: Exemple d'un réseau sans fil à un saut

Il existe de nombreux réseaux sans fil à un saut tels que le WiMAX (Worldwide Interoperability for Microwave Access), le bluetooth, et le plus utilisé: le Wifi. Les réseaux sans fil à un saut sont limités à certaines zones géographiques, ils ne possèdent qu'une portée limitée à la zone de couverture de

leur point d'accès et dépendent d'une maintenance et d'une infrastructure lourde et coûteuse. Pour pallier à ces limitations, les chercheurs et industriels se sont intéressés à la fin des années 90 aux réseaux sans fil à routage ad-hoc multi-sauts (multi-hop ad-hoc routing). Il existe deux principales familles de réseaux ad-hoc multi-sauts : les réseaux MANET (Mobile Ad-hoc Network) et les réseaux mesh sans fil. Les réseaux à routage ad-hoc multi-sauts sont composés d'un ensemble d'entités sans fil. Si le destinataire n'est pas directement à la portée de l'émetteur, les messages de l'émetteur sont alors relayés d'entité sans fil en entité sans fil jusqu'à destination. Les réseaux MANETs se sont avérés être des réseaux peu fiables car ils sont composés d'entités mobiles, la mobilité des nœuds pouvant entrainer à tout moment des ruptures de routes ainsi que la partition du réseau (Conti et Giordano 2007). Contrairement aux MANETs, les réseaux mesh sans fil sont composés d'un cœur de réseau à entités fixes, ce qui leur confère une certaine stabilité. Les réseaux mesh sans fil sont actuellement des réseaux complémentaires aux réseaux sans fil à un saut (Conti et Giordano 2007).

La lourdeur, le coût et le manque de flexibilité des réseaux sans fil à un saut ont entrainé l'émergence des réseaux mesh. Ces réseaux sont composés de deux entités différentes, les clients mesh et les routeurs mesh (voir figure 3) (Akyildiz, Wang et Wang 2005). Les routeurs mesh ont pour principale fonctionnalité de faire suivre les données des utilisateurs de routeurs en routeurs. Parmi les routeurs mesh, certains ont des fonctionnalités supplémentaires : ce sont les routeurs portail d'accès. Les routeurs portails d'accès permettent de faire l'interface entre les utilisateurs et le réseau mesh, ou entre le réseau mesh et un réseau généralement plus étendu tel qu'Internet. Les clients mesh sont les utilisateurs du réseau, ils peuvent être soit de simples terminaux ou des réseaux complets, tels que des réseaux wifi, des réseaux de capteurs, etc.

L'ensemble des routeurs mesh forme le cœur du réseau dans lequel réside la complexité. Un réseau mesh peut être considéré comme une architecture trois-tiers (Suli et Dipankar 2007) dont le premier niveau correspondrait à l'ensemble des clients mesh, le second niveau à l'ensemble des routeurs mesh qui font soit suivre les données et/ou qui jouent le rôle de portail d'accès entre le cœur de réseau et les clients mesh et, finalement, le troisième niveau à l'ensemble des routeurs mesh portail d'accès qui permettent de relier le réseau mesh à un réseau plus étendu, tel qu'Internet. Par la suite, nous utiliserons indifféremment le terme routeur mesh et nœud. Les réseaux mesh comme les MANETs sont basés sur un routage ad-hoc multi-sauts (contrairement aux réseaux à un saut). Cette particularité des réseaux mesh entraine qu'aucun protocole de routage issu des réseaux sans fil à un saut ou des réseaux filaires n'est facilement adaptable sur les WMNs.

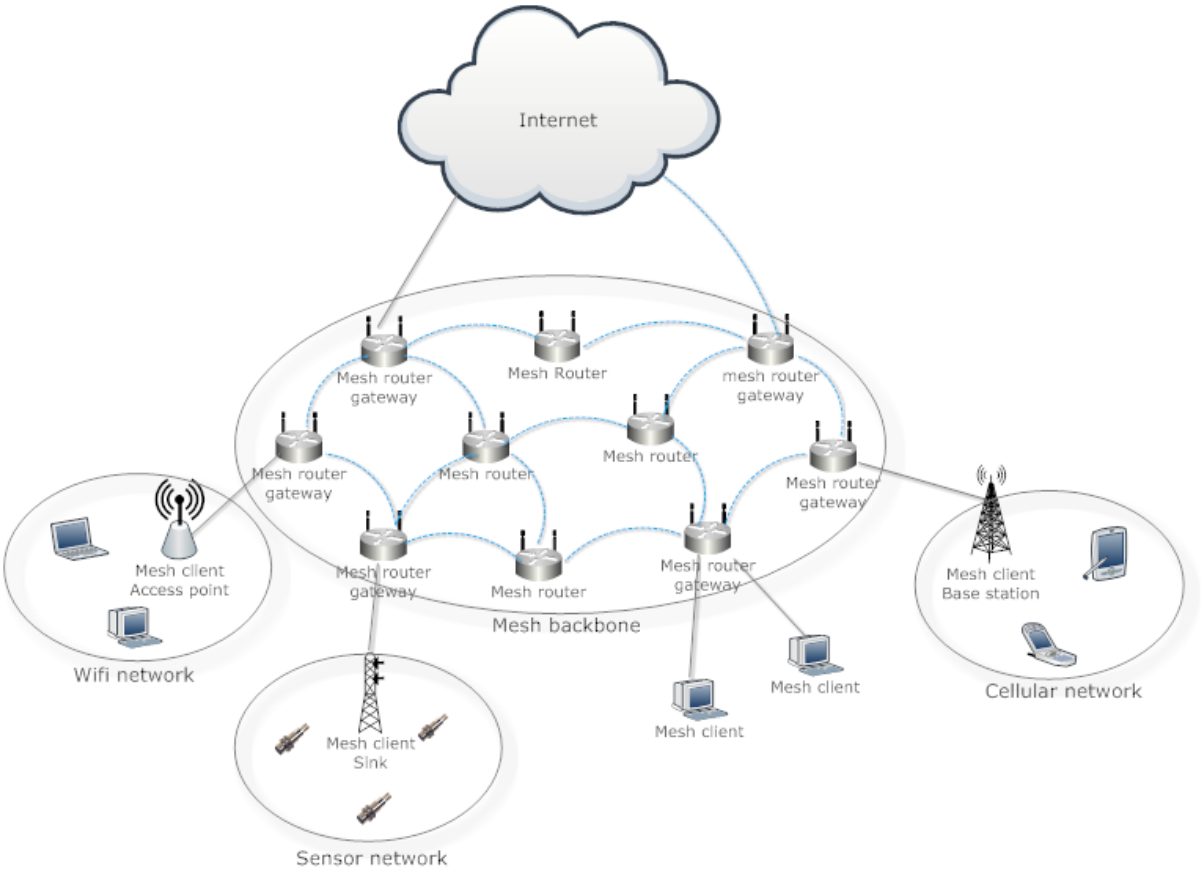


Figure 3: Exemple d'un réseau

2.2.1. Les protocoles de routage dans les réseaux mesh

Les réseaux ad-hoc multi-sauts tels que les réseaux mesh (WMNs) ont entrainé l'apparition de nouveaux protocoles de routage, les protocoles de routage ad-hoc multi-sauts. Ces protocoles sont pour la majorité adaptés à la fois pour les WMNs et pour les MANETs, même si les contraintes dans les MANETs sont plus importantes en termes de ressources (CPU – Central Processing Unit, mémoire, énergie) et de mobilité (Conti et Giordano 2007). Les protocoles de routage multi-sauts sont classés en deux catégories: les protocoles de routage réactifs et les protocoles de routage proactifs. Au sein de chacune de ces catégories, on différencie les protocoles de routage qui sont multi-chemins de ceux qui ne le sont pas.

Les protocoles de routage réactif ne sont pas basés sur des tables de routage car les nœuds découvrent le réseau à la demande, c'est à dire lorsqu'ils souhaitent émettre des données. Parmi les protocoles de routage réactif, il y a AODV (Ad-hoc On-demand Distance Vector) (Perkins, Royer et

Das 2003) et DSR (Dynamic Source Routing) (David et David 1996). AODV a d'ailleurs été intégré dans la norme IEEE 802.11s.

Dans AODV (Perkins, Royer et Das 2003), lorsqu'un nœud souhaite émettre il diffuse un paquet de demande de route RREQ (Route Request). Ce dernier contient, entre autres, un TTL (Time To Live), l'identifiant de la destination et de la source et un identifiant de diffusion. A la réception d'un RREQ, un nœud enregistre, pour une source et une destination donnée, l'identifiant du nœud qui vient de lui transmettre le paquet. Le nœud rediffuse ensuite le RREQ uniquement si le TTL n'a pas expiré et si c'est la première fois qu'il reçoit ce RREQ (voir figure 4).

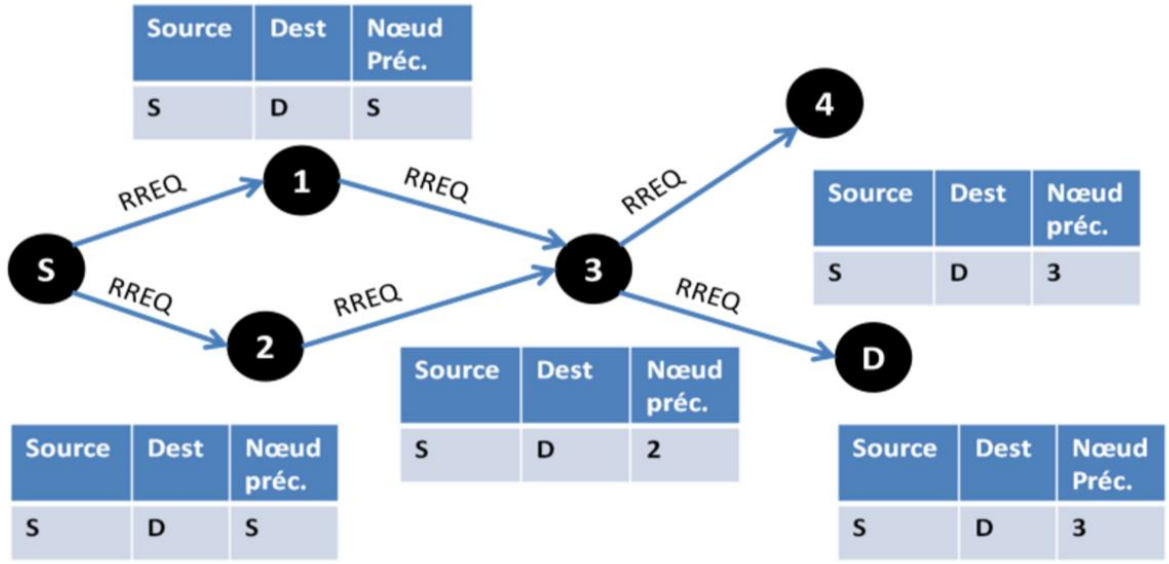


Figure 4 : Broadcaste d'un RREQ par la source S, rediffusé par tous les nœuds qui le reçoive

Lorsque la destination reçoit le premier paquet RREQ, elle envoie un paquet de réponse RREP (Route Reply) le long de la route traversée par le RREQ. Elle précise dans le RREP, le champ source et le champ destination. A la réception d'un RREP, un nœud enregistre pour une source et une destination donnée, l'identifiant du nœud qui vient de lui transmettre le paquet. Chaque nœud recevant le RREP fait suivre le paquet au nœud qui lui avait fait suivre le RREQ et dont il avait enregistré l'identifiant (voir figure 5).

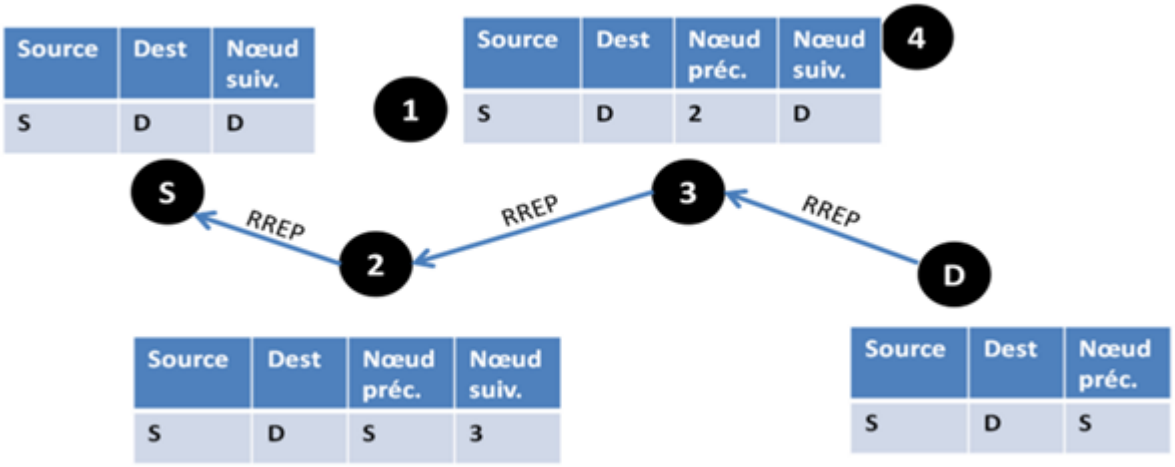


Figure 5 : envoi d'un RREP par la destination D le long du chemin emprunté par le RREQ

Lorsque la source reçoit le RREP, elle commence à envoyer les données via la route empruntée par le RREP. Chaque nœud sur le chemin connait le nœud suivant de la route puisqu'il l'a enregistré lors de la réception du RREP et peut donc faire suivre le message. Les informations concernant une route sont conservées le temps de l'envoi des données, c'est-à-dire que toutes les informations stockées au niveau des nœuds de la route sont effacées lorsque la source arrête d'envoyer des données.

Le protocole DSR (David et David 1996) fonctionne similairement à AODV et permet d'alléger la surcharge induite par la diffusion des paquets de RREQ. Dans DSR, un nœud n'enregistre pas uniquement le nœud précédent et le nœud suivant le long de la route découverte mais l'ensemble des nœuds de cette route. Pour ce faire, chaque nœud recevant le RREQ enregistre son identifiant dans le paquet sans effacer l'identifiant des nœuds précédents déjà présents. Lorsque la destination reçoit le RREQ, ce dernier contient l'ensemble des nœuds de la route. La destination va reporter l'ensemble des nœuds de la route dans le RREP. Chaque nœud recevant le RREP enregistre dans un cache, pour une destination et une source donnée, la route découverte. Les nœuds recevant une demande de découverte de route pourront répondre directement à cette demande via un RREP s'ils possèdent un chemin pour cette destination dans leur cache.

Les protocoles de routage réactif entrainent beaucoup de surcharge via la diffusion de messages, cependant, cette surcharge peut être diminuée via l'utilisation de caches au niveau des nœuds qui permettent de mémoriser les routes. Lorsqu'il n'y a pas de cache, ces protocoles nécessitent qu'à chaque émission d'un nouveau flux, les nœuds découvrent leur route (Boukerche, et al. 2011). Via cette découverte de route, les nœuds peuvent mener en parallèle une découverte de ressources. Cette propriété des protocoles de routage réactifs explique leur large exploitation dans les

protocoles de contrôle d'admission (CA) des WMNs où la découverte de ressources est une étape importante.

Les protocoles de routage proactif tels que OLSR (Optimized Link State Routing Protocol) (Clausen et Jacquet 2003) et DSDV (Destination-Sequenced Distance-Vector Routing) (Perkins et Bhagwat 1994) sont basés sur des tables de routage qui vont être mis à jour périodiquement. Ainsi lorsqu'un nœud cherche une route vers une destination, la source établit localement la route grâce aux informations qu'elle possède dans sa table. Les protocoles de routage proactif permettent de réduire le temps de découverte d'une route puisque chaque nœud peut la calculer localement et n'a plus besoin d'envoyer des messages de découverte de route. Cependant, ils engendrent une surcharge pour tenir à jour les tables et, dans le cas où elles seraient mal tenues à jour, une incapacité pour les nœuds à trouver une route (Boukerche, et al. 2011). Le tableau suivant présente une comparaison entre les protocoles de routage réactif et les protocoles de routage proactif.

Tableau 1 : Comparaison entre protocoles de routage réactif et proactif

	Protocole de routage Réactif	Protocole de routage proactif
Délai de découverte d'une route	Long	Court
Surcharge de maintenance	Faible car pas de table	Important car utilisation de tables
Surcharge de découverte de routes	Importante car découverte par inondation	Faible car découverte à partir de données locales
Possibilité de découvrir les caractéristiques d'une route avant son utilisation	Oui, lors de l'envoi du RREQ et du RREP	Non

Dans nos travaux, nous avons privilégié les protocoles de routage réactif aux protocoles de routage proactif car ils répondent mieux à nos besoins. En effet, nous proposons dans cette thèse un nouveau contrôle d'admission (CA), or un CA nécessite généralement de connaitre les caractéristiques, principalement en termes de bande passante, de délai et de perte de paquets d'une route avant d'y admettre un flux.

2.2.2. L'objectif des réseaux mesh

Les réseaux mesh sans fil ont pour objectif d'étendre rapidement, pour un bas coût et une faible maintenance les réseaux dans de nouvelles zones et de faciliter l'interconnexion entre différents réseaux (Akyildiz, Wang et Wang 2005) (Nandiraju, et al. 2007).

Les réseaux mesh sans fil possèdent un cœur de réseau totalement sans fil, leur installation ne nécessite donc pas de câble, peu de travaux d'installation et leur maintenance s'en trouve facilitée. Ainsi, ils permettent de réduire les coûts et le temps d'installation par rapport aux réseaux sans fil à un saut (Nandiraju, et al. 2007). Les WMNs permettent également d'étendre les réseaux dans des zones isolées et/ou économiquement non viables. Ils sont également des réseaux privilégiés suite à un sinistre, lorsque l'infrastructure préexistante est indisponible, car les réseaux mesh sont rapides d'installation et permettent de faciliter l'intervention des secours. Les routeurs mesh sans fil possèdent plusieurs interfaces réseaux permettant à différents réseaux d'accès (réseau de capteurs, réseau wifi, réseau WiMAX, réseau cellulaire, etc) d'être connectés à un même réseau mesh sans fil et de pouvoir communiquer. De plus, par rapport à d'autres réseaux ad-hoc multi-sauts, les réseaux mesh apportent une certaine fiabilité dans les transmissions car les nœuds sont fixes contrairement aux réseaux MANETs et car les routeurs ne sont pas limités en termes d'énergie et de CPU contrairement aux réseaux de capteurs, puisqu'ils peuvent être branchés au secteur (Akyildiz, Wang et Wang 2005). De par leurs nombreux avantages, les réseaux mesh sont appliqués à de nombreux environnements et domaines.

2.3. Applications des réseaux mesh sans fil

Les réseaux mesh sans fil permettent le déploiement du réseau dans des zones ou/et dans des domaines mal ou peu connectés. De nombreuses entreprises proposent à l'heure actuelle des équipements mesh telles que CISCO, StrixSystems, Mesh City, etc. De plus, une norme IEEE a été publiée récemment sur les réseaux mesh: l'IEEE 802.11s (Wifi-alliance 2011). Cette norme définit un ensemble de protocoles de niveau MAC et physique pour le fonctionnement d'un réseau mesh. Elle intègre notamment un protocole de routage de niveau 2, HWMP (Hybrid Wireless Mesh Protocol) et

un protocole d'authentification basé sur un mot de passe sécurisé et d'établissement de clés, SAE (Simultaneous Authentication of Equals).

Les atouts en termes de coût, de fiabilité, de facilité et de souplesse de déploiement font des réseaux mesh sans fil un réseau privilégié dans de nombreux environnements et domaines. Le domaine militaire a été l'un des premiers domaines à déployer des réseaux mesh (Mobile Mesh Networks for Military, Defense and Public Safety 2013). Le secteur militaire a dans les réseaux mesh comme dans beaucoup de technologies un temps d'avance sur le Civil. Dans le domaine militaire, les réseaux mesh permettent d'apporter le réseau à l'armée dans des zones reculées et hostiles où l'infrastructure est quasi-inexistante et/ou détruite (Peppas et Turgut 2007) (Jung, Ryu et Roh 2010). Ces réseaux pouvant être facilement étendus ou déplacés, ils peuvent donc suivre facilement le mouvement des militaires lors de leur progression. La compagnie TELOS (I-STAMS 2013) fournit à l'armée américaine des équipements comme l' I-STAMS (Scalable, modular tactical mesh wireless network for classified and unclassified voice, video, and data) (voir figure 6) permettant de monter rapidement des réseaux mesh dans des zones hostiles comme l'Afghanistan.

Figure 6: I-STAMS permet de monter rapidement des réseaux mesh dans le domaine militaire (I-STAMS 2013)

Les réseaux mesh sont également déployés dans d'autres domaines où les réseaux classiques atteignent leur limite, telles que :

- domaine des secours (Jia, et al. 2008) (Portmann et Pirzada 2008). Lors d'opérations de secours, les unités de terrain peuvent avoir besoin de communiquer. S'il n'existe pas d'infrastructure ou si celle-ci a été détruite, suite par exemple à un tremblement de terre, un ouragan ou une inondation, les réseaux mesh peuvent permettre de mettre en place une communication, entre équipes, rapidement afin de coordonner les efforts.
- domaine domestique (Akyildiz, Wang et Wang 2005). Dans une maison ou un appartement, il existe des zones blanches qu'un unique point d'accès est incapable de couvrir à cause de sa

limitation de portée. Ainsi, un réseau mesh peut permettre de résoudre ce problème. Il peut également permettre de relier les équipements ménagers entre eux.

- domaine des transports (Akyildiz, Wang et Wang 2005). Les réseaux mesh peuvent étendre la portée d'un point d'accès dans un bus, un ferry ou un train (voir figure 7) afin que tous les passagers puissent bénéficier d'une connexion. Le réseau mesh d'un ferry pourrait ensuite être connecté à Internet via un lien satellitaire.

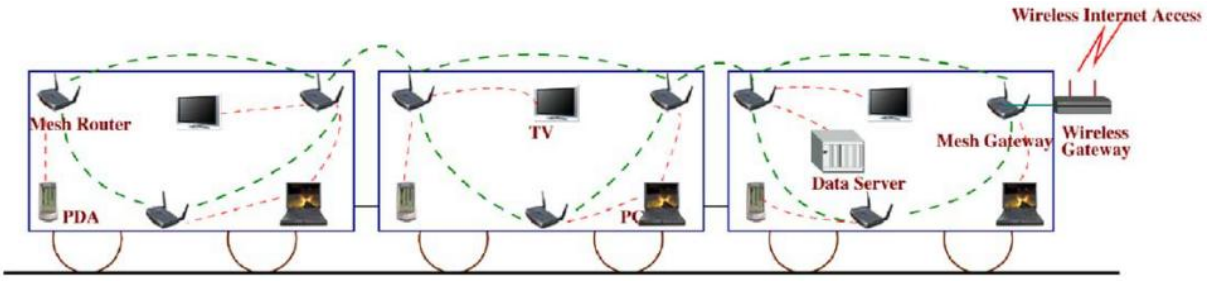


Figure 7 : Un réseau mesh sans fil déployé dans un train (Akyildiz, Wang et Wang 2005)

Les réseaux mesh sont donc des réseaux privilégiés dans certains domaines, mais également dans certaines zones dépourvues actuellement d'infrastructures telles que :

- les zones rurales (Sichitiu 2005) (Akyildiz, Wang et Wang 2005). Dans certaines zones rurales, la pose de câbles peut s'avérer économiquement non viable, le nombre de clients potentiels n'étant pas suffisant. De par leur faible coût, les réseaux mesh peuvent assurer la connexion dans ces zones.
- les zones accidentées (Portmann et Pirzada 2008) (Akyildiz, Wang et Wang 2005). Dans certaines zones, la pose de câbles peut se révéler impossible car le terrain peut être trop abrupt. Les réseaux mesh sont encore une fois la solution pour permettre la connexion de ces zones.
- les zones en voie de développement (Sichitiu 2005). En effet, le bas coût de ces réseaux permet leur installation dans des zones où le déploiement d'un réseau sans fil classique ne serait pas économiquement viable, telles que les zones en voie de développement. Par exemple, le projet « One laptop per children » (One Laptop per children association 2005) ayant pour but de fournir un ordinateur par enfant pauvre, intègre le protocole IEE 802.11s ainsi qu'une transmission sans fil particulièrement puissante dans tous ses ordinateurs afin d'offrir l'accès à Internet aux enfants de pays en voie de développement.

En conclusion, les réseaux mesh peuvent être utilisés dans de nombreux domaines et environnements où les réseaux sans fil à un saut sont inadaptés pour des raisons de coût, de contraintes d'installations, etc. Cependant, le déploiement des WMNs est limité à cause de leur

faible bande passante, d'un mauvais passage à l'échelle, de problèmes de sécurité et de l'iniquité qu'ils induisent entre les nœuds. Ce sujet est développé dans la section suivante.

2.4. Limitations des réseaux mesh sans fil

Comme développé dans la section précédente, les réseaux mesh commencent à être déployés dans différents domaines et environnements où les réseaux sans fil à un saut s'avèrent inadaptés. Cependant, leur déploiement est freiné par certaines de leurs caractéristiques qui posent problème en termes de :

- qualité de service (QoS) (Mogre, Hollick et Steinmetz 2007) (Nandiraju, et al. 2007). Les réseaux mesh peinent à satisfaire la QoS des utilisateurs car ces réseaux subissent des problèmes d'interférence dans les transmissions, d'iniquité entre les nœuds, de perte de paquets importantes sur les liens et de limitation de la bande passante (Aguayo, et al. 2004). En effet, la nature sans fil du canal des réseaux mesh les rend vulnérables aux interférences et à la perte de paquets. De plus, le routage multi-sauts accentue ces phénomènes puisque chaque saut d'un paquet augmente sa probabilité d'être endommagé. La nature sans fil induit également une limitation en bande passante, car seules certaines bandes de fréquences sont utilisables et chaque nœud sans fil doit à la fois faire suivre les données de ses utilisateurs ainsi que celles des utilisateurs d'autres routeurs mesh. Des études (Aoun et Boutaba 2006) (Subramanian et Leith 2010) ont montré que les réseaux mesh souffrent d'iniquité (Abouaissa, Brahmia et Lorenz 2013) (Subramanian et Leith 2010) et que, en cas de congestion du réseau, les nœuds situés à plus de deux sauts des portails d'accès se retrouvent en famine (Aoun et Boutaba 2006) (Ouni, Rivano et Valois 2010). Les limites en termes de QoS des WMNs représentent un obstacle au déploiement d'applications à fortes contraintes (VoIP, streaming, visioconférence, jeux vidéo en réseau) (Cheng, Prasant et Lee 2008).
- passage à l'échelle (Conti et Giordano 2007). La capacité des réseaux mesh est restreinte par le phénomène de goulot d'étranglement au niveau des portails d'accès. Une étude (Jun et Sichitiu 2003) a montré analytiquement que dans un réseau mesh sans fil, lorsque le nombre de nœuds n augmente, le débit de chaque nœud décroit en $(1/n)$.
- sécurité. La nature sans fil des WMNs facilite l'écoute des messages, ainsi que leur modification et leur brouillage. Ils sont donc particulièrement vulnérables. De plus, les

routeurs mesh n'étant souvent pas protégés physiquement (ils ne sont pas enterrés et se trouvent souvent à l'extérieur), ils peuvent être facilement modifiés ou remplacés par des attaquants (Ben Salem et Hubaux 2006). La gestion d'un réseau mesh étant basée sur la collaboration entre les nœuds, il suffit alors qu'un nœud soit capturé par un attaquant pour que l'ensemble du réseau soit compromis (Ben Salem et Hubaux 2006).

Les nombreux avantages des réseaux mesh permettent de concevoir de nombreuses applications difficilement réalisables avec des réseaux sans fil à un saut et d'apporter l'Internet dans des zones qui ne peuvent pas être couvertes avec les solutions actuelles. Cependant, les limites actuelles des réseaux mesh sans fil, en termes de sécurité et de QoS freinent leur déploiement.

2.5. Conclusion

Les réseaux mesh sans fil présentent de nombreux atouts, tels que le coût, la facilité et la rapidité d'installation et sont des solutions complémentaires aux réseaux sans fil à un saut. Ces réseaux sont utilisés dans le domaine militaire et pourraient être plus largement étendus dans le domaine civil, par exemple dans des zones rurales, accidentés ou économiquement non viables. Ainsi, à terme, ces réseaux pourraient permettre d'établir des connexions «Anywhere, Anytime ». Cependant, ils présentent actuellement des limites en termes de qualité de service (faible capacité, perte de paquets, délai des flux) et de sécurité (faible protection des routeurs mesh, large zone d'écoute) freinant leur déploiement ; les exigences d'applications à fortes contraintes (jeux en ligne, vidéo à la demande, certaines applications de paiement en ligne) ne peuvent être respectées. En proposant et en intégrant des solutions de qualité de service et de sécurité aux réseaux mesh sans fil, ces derniers pourraient être déployés à plus large échelle. Ainsi, de nombreux utilisateurs pourraient profiter des nombreux avantages de ces réseaux : tel est l'un des principaux objectifs du contrôle d'admission, de la planification de liens et des systèmes de confiance.

Partie 1. Un système de contrôle d'admission avec planification des liens dans un réseau mesh sans fil

Partie 1.Introduction

Les réseaux mesh suscitent un grand intérêt dans le domaine de la recherche car ils permettent de connecter des zones encore blanches. Cependant, comme l'a souligné le chapitre précédent, le déploiement de ces réseaux est freiné par leur actuelle médiocre qualité de service induite principalement par leur faible capacité (Nandiraju, et al. 2007), une importante perte de paquets et un non respect des contraintes de nombreux flux. Cette partie introduit dans un premier temps un état de l'art sur deux mécanismes de qualité de service qui permettent de solutionner partiellement les problèmes de qualité de service des réseaux mesh : le contrôle d'admission et la planification des liens sur le réseau. Le contrôle de planification permet d'accepter sur le réseau seulement les flux dont il peut respecter les contraintes ; cependant, les solutions existantes sont limitées dans le nombre de flux qu'elles peuvent admettre à cause de la faible capacité du réseau. La planification des liens permet d'attribuer, à chaque lien du réseau, un ensemble de slots sur lesquels il peut émettre sans risque de collision, elle diminue ainsi la perte de paquets sur le réseau et augmente sa capacité utile. Cependant, la planification de liens ne s'adapte pas à la charge du réseau, ce qui peut entrainer une perte en bande passante utile et des congestions. Dans un second temps, cette partie présente notre solution de contrôle de planification intégrant un système de planification de liens. Cette solution re-planifie les émissions des liens à chaque fois qu'un nouveau flux est admis dans le réseau. L'objectif de cette proposition est de limiter la perte de paquets et d'augmenter la capacité utile du réseau tout en assurant les contraintes en termes de délai et de bande passante de l'ensemble des flux admis.

Chapitre 3. Etat de l'art : Contrôle d'admission et planification des liens

3.1. Introduction

Le chapitre précédent présente les réseaux mesh et souligne l'une de leurs principales limites, une bande passante restreinte. La faible capacité (Nandiraju, et al. 2007)de ces réseaux entraine régulièrement, le non-respect des contraintes d'applications aux exigences strictes telles que la VoIP, le streaming, la visioconférence, les jeux vidéo en réseau, et dès lors la non satisfaction des utilisateurs. Le tableau 2 présente la sensibilité des principales applications réseaux aux quatre plus importantes métriques de QoS que sont la bande passante (BP - bandwidth), le délai (delay), la gigue (jitter) et la perte de paquets (Loss) (Chen, Farley et Ye 2004).

Tableau 2 : Sensibilité des applications à différentes métriques de QoS (Khoukhi 2006)

Performance dimensions				
Application	Bandwidth	Sensitivity to:		
		Delay	Jitter	Loss
VoIP	Low	High	High	Med
Video Conferencing	High	High	High	Med
Streaming Video on Demand	High	Med	Med	Med
Streaming Audio	Low	Med	Med	Med
Client/Server Transactions	Med	Med	Low	High
E-mail	Low	Low	Low	High
File transfer	Med	Low	Low	High

Il est donc indispensable de doter ces réseaux de mécanismes de qualité de service afin de satisfaire les exigences des utilisateurs pour qu'ils puissent avoir un accès fluide à leurs différentes applications. Ce chapitre présente un état de l'art sur deux solutions de QoS complémentaires: le contrôle d'admission, qui permet de garantir les contraintes des flux (Hanzo et Tafazolli 2009), et la planification des liens qui a pour principal objectif d'augmenter la bande passante du réseau (Gore et Karandikar, Link Scheduling Algorithms for Wireless Mesh Networks 2011). Ces mécanismes seront exploités au cours de cette thèse pour leurs nombreux atouts.

3.2. Contrôle d'admission

De nombreuses solutions ont été proposées afin d'augmenter la capacité des réseaux mesh comme des mécanismes d'assignation de canal, de planification de liens et de routage efficace (Pathak et Dutta 2011). Cependant, la plupart de ces solutions, bien qu'elles augmentent la capacité du réseau ne peuvent plus, lorsqu'elles font face à une congestion du réseau, garantir les contraintes des flux. Ainsi, le contrôle d'admission apparait comme indispensable puisqu'il est l'un des uniques mécanismes apte à garantir les contraintes des flux dans les réseaux mesh (Rezgui, Hafid et Gendreau 2008). Le contrôle d'admission a pour but d'accepter ou de rejeter un nouveau flux selon si le réseau est capable de garantir ses contraintes ainsi que celles des flux préalablement admis. Cependant, bien que la définition d'un contrôle d'admission soit assez simple, sa réalisation pose de nombreux défis.

3.2.1. Défis d'un contrôle d'admission

Lors du contrôle d'admission d'un flux, ce dernier est accepté si ses contraintes en termes de qualité de service peuvent être respectées par le réseau ainsi que celles des flux préalablement admis.

3.2.1.1. Les métriques d'un contrôle d'admission : des métriques difficiles à estimer

Avant d'émettre un flux, la source doit préciser les caractéristiques requises par le flux, telles que, la bande passante (BP), le délai, la perte de paquet, la gigue ainsi que sa destination (Hanzo et Tafazolli 2009). Le réseau doit ensuite évaluer s'il existe une route pour ce flux possédant suffisamment de ressources pour satisfaire ses contraintes tout en respectant celles des flux préalablement admis sur le réseau. Le réseau doit donc être capable d'évaluer ses ressources le long d'une route. Les ressources en bande passante au niveau d'une route dépendent du nœud formant le goulot d'étranglement sur la route, c.à.d. du nœud ayant le minimum de bande passante disponible sur la route.

Les ressources d'un nœud sur une route donnée sont définies selon plusieurs métriques, telles que le CPU, l'énergie, la bande passante du lien du nœud, la perte de paquets du lien du nœud, l'occupation

et la taille de son buffer, etc (Hanzo et Tafazolli 2009). L'énergie et le CPU n'étant pas des ressources limitées dans les réseaux mesh, elles ne sont généralement pas considérées. La bande passante disponible d'un routeur est une ressource très difficile à estimer (Lohier, Ghamri-Doudane et Pujolle 37-48) (Belbachir, et al. 2012), contrairement à la taille et l'occupation du buffer ainsi que dans une moindre mesure la perte de paquet sur un lien. En effet, chaque nœud partage la capacité de son canal avec les nœuds environnants (Hanzo et Tafazolli 2009). L'accès au canal dans un réseau mesh est généralement à compétition, ce qui n'assure à aucun nœud une BP minimum.

Les nœuds d'un réseau mesh ont généralement un accès à compétition au canal basé sur le protocole DCF (Distributed Coordination Function) proposé par la norme IEEE 802.11 (IEEE 1997). Ce protocole considère que chaque nœud possède une zone de transmission ainsi qu'une zone d'écoute. Dans la zone de transmission d'un routeur est située l'ensemble des nœuds avec lequel il peut directement communiquer. La zone d'écoute d'un routeur contient l'ensemble des nœuds qu'il peut entendre, même s'il n'est pas toujours capable de comprendre leurs messages (voir figure 8).

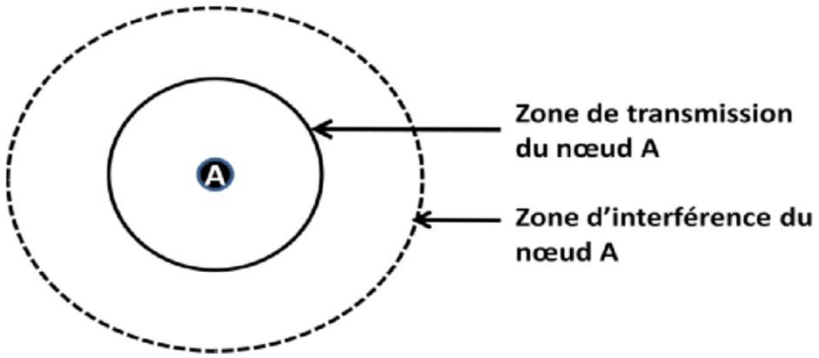


Figure 8 : Les différentes zones d'un nœud d'après la norme IEEE 802.11

Afin d'éviter les interférences, un nœud, selon la norme IEEE 802.11, peut émettre si sa zone d'écoute est silencieuse et donc si aucun autre nœud de sa zone d'écoute n'émet déjà. Ainsi, d'après la norme IEEE 802.11, un nœud partage son canal avec l'ensemble des nœuds situés dans sa zone d'écoute.

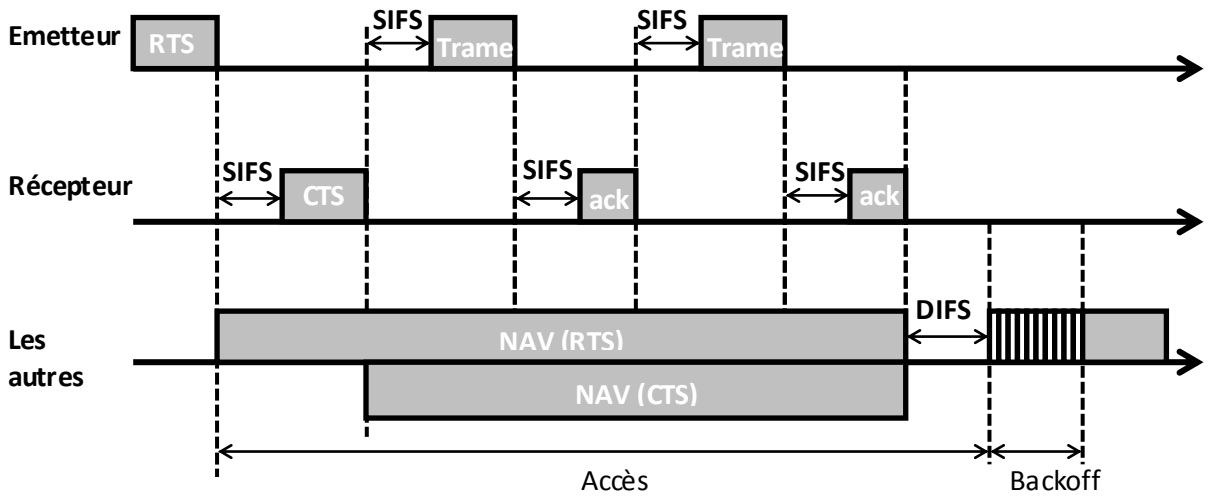


Figure 9 : Protocole DCF

Afin d'éviter les collisions, DCF utilise des espaces inter-trames entre l'envoi des paquets, un algorithme exponentiel de *backoff* et une méthode optionnelle le RTS/CTS (Xu, Gerla et Bae 2002) (Ready To Send-RTS et Clear To Send-CTS) qui est très largement utilisée dès qu'un paquet dépasse une certaine taille. Si un nœud souhaite émettre, il doit attendre que sa zone d'écoute soit silencieuse. Dés que sa zone d'écoute est silencieuse depuis au moins un temps d'espace inter-trames appelée DIFS (DCF Interframe Space), alors le nœud lance l'algorithme de *backoff* exponentiel. Cet algorithme permet d'éviter que l'ensemble des nœuds d'une zone émettent simultanément lorsqu'ils détectent leur zone d'écoute libre et qu'il y ait des collisions. Un nœud exécutant l'algorithme de *backoff* met en place un minuteur ; il choisit aléatoirement dans un intervalle appelé fenêtre de contention [0 ; CW] une valeur. Cette valeur représente l'ensemble des slots que le nœud doit attendre avant de pouvoir émettre (voir figure 9). Si un nœud de sa zone d'écoute émet avant que son minuteur n'expire, alors le nœud suspend ce dernier, il le reprendra lorsque sa zone d'écoute sera à nouveau libérée depuis un temps DIFS. Si le minuteur expire et que le canal est toujours libre, le nœud peut commencer à émettre. Si la méthode RTS/CTS est déclenchée alors le nœud envoie un paquet RTS au destinataire où il précise la durée pendant laquelle il va utiliser le canal. Le destinataire, après avoir reçu le RTS et si sa zone d'écoute est libre, envoie un paquet CTS contenant également la durée de l'échange. Tous les nœuds recevant le RTS ou le CTS et n'étant ni la source ou la destination vont alors retarder leur transmission de la durée indiquée dans les paquets en fixant leur NAV (Network Allocation Vector), qui est une minuterie, de la durée pendant laquelle le médium est réservé (voir figure 9). A la réception du CTS, le nœud émetteur commence à envoyer des paquets de données. A chaque paquet reçu le destinataire envoie un acquittement. A partir de l'envoi du RTS, l'émetteur et le récepteur attendent une période SIFS (Short Interframe Space) avant d'envoyer une trame (voir figure 9), la durée de SIFS est bien plus

courte que celle de DIFS. La taille de la fenêtre de contention définie par la valeur de CW va s'adapter aux caractéristiques du réseau. Si la transmission d'un nœud subit une collision, alors la taille de la fenêtre de contention du nœud est doublée ainsi $CW_{new} = 2 * CW$.e Cependant, la taille de la fenêtre de contention ne peut excéder la valeur $CWmax$ fixée par le standard. Si, un nœud transmet avec succès alors sa fenêtre de contention est remise à sa taille minimum $CW = CWmin$.

Le protocole DCF a été conçu pour des réseaux sans fil à un saut, et n'est pas approprié aux réseaux mesh (Nandiraju, et al. 2007). Son utilisation dans les réseaux mesh pose le problème de « nœuds cachés » bien plus que dans les réseaux à un saut (Tzu-Jane et Ju-Wei 2005) (Lu, Kaishun et Hamdi 2012) et entraine perte de paquets et famine au niveau de certains nœuds (Nandiraju, et al. 2007). Le terme « nœuds cachés » sera expliqué dans les sections suivantes. La plupart des contrôles d'admission existants sont basés sur le protocole DCF et souffrent de son inadéquation aux réseaux mesh.

3.2.1.2. Différentes approches de contrôle d'admission, différents défis

Il existe différentes approches pour réaliser un contrôle d'admission dans un réseau mesh. On classifie généralement ces approches en deux catégories, les contrôles d'admission découplés avec le routage et ceux couplés avec le routage (voir la figure 10) (Hanzo et Tafazolli 2009).

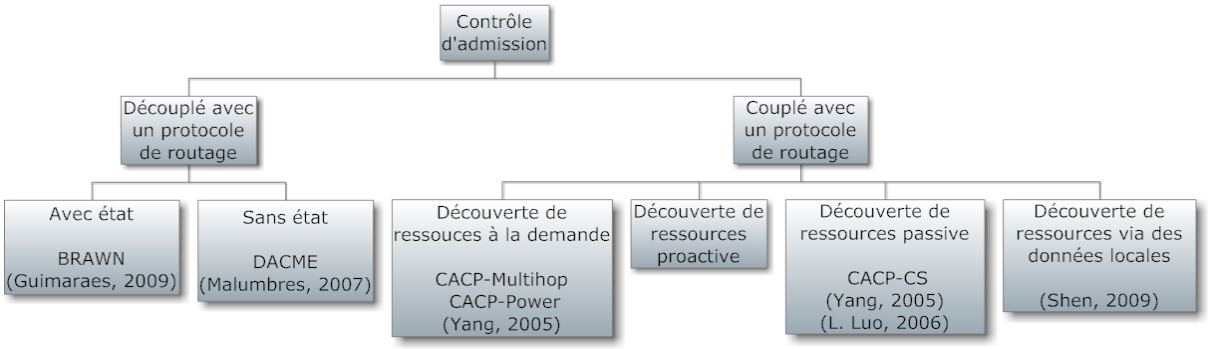


Figure 10 : Classification des protocoles de contrôle d'admission

Les solutions de contrôle d'admission (CA) découplées avec le routage ne dépendent d'aucun protocole de routage préexistant, ils supposent que la route ait été préalablement découverte (Calafate, et al. 2007) (Guimares, et al. 2009) (Hanzo et Tafazolli 2009). Le CA vérifie ensuite si cette route peut garantir les contraintes du flux. Ce découplage permet à ces CAs d'être implémentés sur n'importe quel protocole de routage et d'être ainsi modulables. Cependant, dans les CAs découplés,

le rejet d'une route entraine la perte de ressources consommées pour établir la route. Un protocole de CA découplé avec le routage peut être soit avec état ou sans état. Contrairement à un CA avec état, un CA sans état ne stocke aucune information sur un flux au niveau des nœuds intermédiaires de sa route. Un CA avec état, contrairement à un CA sans état, peut continuer à garantir les contraintes des flux, même si certains ne respectent pas leurs contraintes en termes de BP car il peut limiter le débit des flux à celui qu'ils ont demandé (Hanzo et Tafazolli 2009).

Les CAs couplés avec un protocole de routage apportent une plus grande granularité, puisque, lors de la découverte d'une route, chaque nœud vérifie s'il peut garantir ou non les contraintes du flux, et si l'un d'entre eux ne peut pas les garantir, on évite alors ce nœud dans la construction de la route. Ainsi, contrairement aux CAs découplés, si un nœud ne peut pas garantir les contraintes d'un flux, on rejette uniquement un nœud et non tout une route.

Les contrôles d'admission couplés avec un protocole de routage diffèrent principalement par leur méthode de découverte de ressources. Les méthodes de découvertes de ressources sont basées soit sur des données locales, soit sur des méthodes de découverte passive, à la demande ou proactive (Tafazolli 2009). Des exemples des différentes méthodes de découvertes de ressources seront donnés dans la section suivante.

Les méthodes de découverte de ressources peuvent engendrer de la surcharge par l'envoi de messages, une sur-estimation ou une sous-estimation des ressources ainsi qu'un important délai de mise en place des routes. La sur-estimation des ressources peut entrainer la congestion du réseau alors que la sous-estimation des ressources engendre une perte de ressources et l'admission dans le réseau de moins de flux que ce qu'il pourrait réellement supporter. Il est préférable de sous-estimer les ressources que de les surestimer car, la sous-estimation n'entraine pas de violation des contraintes des flux contrairement à la sur-estimation. Le principal objectif du CA qui est de satisfaire les contraintes des flux est alors préservé. La figure 10 présente la classification des CA dans les réseaux mesh sans fil. Dans la section suivante, nous présenterons au moins une solution de CA pour chaque type de contrôle d'admission existant. Finalement, un contrôle d'admission permet:

- d'accepter ou de rejeter un flux selon si le réseau est capable de respecter ses exigences en termes de QoS
- d'éviter les phénomènes de congestion

De plus, les CA peuvent également avoir des buts secondaires afin d'améliorer leurs performances :

- minimiser leur surcharge

- minimiser le délai d'établissement des flux
- estimer avec précision les ressources des routes afin d'éviter les phénomènes de congestion ou de perte de ressources

La principale difficulté de conception d'un CA repose dans la proposition d'une méthode d'estimation des ressources du réseau et plus particulièrement d'estimation de la bande passante car cette dernière est partagée entre l'ensemble des nœuds de sa zone d'écoute (Belbachir, et al. 2012). La connaissance des ressources du réseau permet à un CA de déterminer s'il est apte ou non à accepter un nouveau flux, il est donc primordiale à son bon fonctionnement.

3.2.2. Solutions existantes de contrôle d'admission

Cette partie présente indifféremment les contrôles d'admission dans les réseaux MANETs et les réseaux mesh, puisque les contrôles d'admission existant dans les MANETs peuvent être aisément déployés dans les WMNs.

3.2.2.1. Concepts utilisés

Afin de comprendre les solutions de CA existantes, il est indispensable de connaitre certains concepts développés ci-après.

Les voisins d'un nœud : Les voisins d'un nœud sont l'ensemble des nœuds situés dans sa zone de transmission. Un nœud j peut être le voisin d'un nœud i mais le nœud i peut ne pas être le voisin de j car chaque nœud peut posséder un rayon de zone de transmission différent.

L'interférence : Il y a interférence lors d'une transmission sur un lien lorsque le nœud récepteur ne reçoit pas le message du nœud émetteur parce qu'au moins un nœud dans les environs émet simultanément avec l'émetteur. Afin de savoir s'il y a interférence au niveau d'un nœud, les chercheurs utilisent des modèles d'interférence (Iyer, Rosenberg et Karnik 2009). Cependant, il existe de nombreux modèles d'interférence plus ou moins précise et selon le modèle, il peut y avoir ou non une interférence sur un lien (Iyer, Rosenberg et Karnik 2009).

Nœuds cachés (Nandiraju, et al. 2007) **:** Soit un émetteur qui envoie des données à un récepteur et un troisième nœud situé prés du récepteur qui n'entend pas la première transmission et émet.

L'émission du troisième nœud peut alors interférer sur la communication en cours. Ce nœud est alors le nœud caché de la communication en cours. Ce phénomène est illustré par la figure 11, A envoie des données à B, C décide d'émettre car il n'entend pas qu'il y a une communication en cours, or en émettant il interfère avec la communication entre A et B.

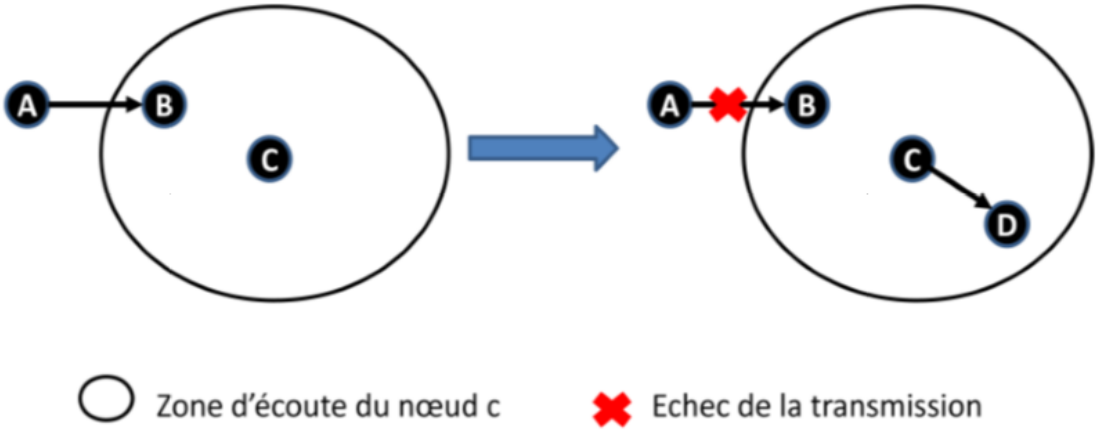


Figure 11 : C est un nœud caché de la communication entre A et B

La compétition intra-route : Lorsqu'un flux est émis sur une route, il est relayé par un ensemble de nœuds. Un nœud de la route, pour émettre le flux, doit rentrer en compétition avec, entre autre, les autres nœuds de la route qui sont situés dans sa zone d'écoute et qui émettent également le flux. Ainsi, lorsqu'un nœud accepte un flux, sa bande passante est consommée à chaque fois que le flux est émis par lui-même ou un nœud de sa zone de portée d'écoute (voir figure 12). Ce phénomène s'appelle la contention intra-route et est induit par l'accès à compétition des nœuds dans DCF (Yang et Kravets 2005).

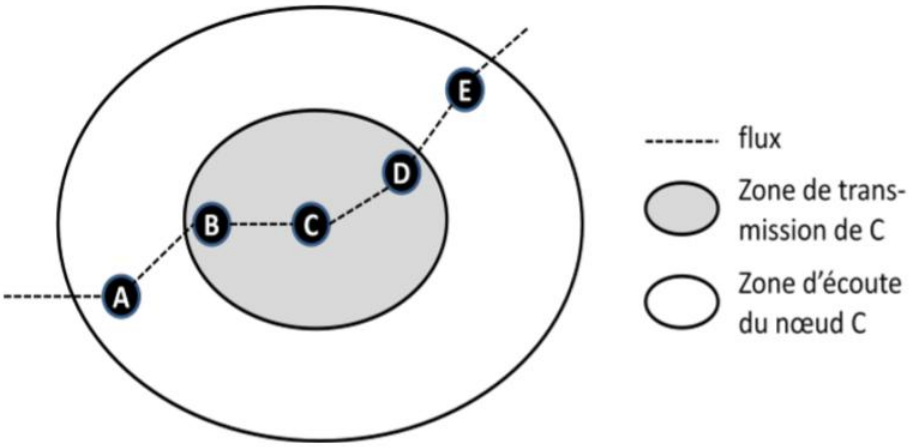


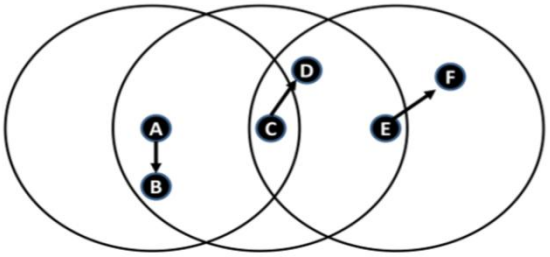
Figure 12 : Phénomène de contention intra-route; le flux est émis par cinq nœuds A, B, C, D, E appartenant à la zone d'écoute de C. L'émission du flux par les nœuds A, B, C, D, E diminuent la bande passante disponible du nœud C.

Bande passante locale d'un nœud : Un nœud ne peut émettre que lorsque tous les nœuds de sa zone d'écoute sont silencieux. La bande passante locale d'un nœud dépend des transmissions des nœuds situés dans sa zone d'écoute, elle est proportionnelle au temps de silence sur la zone d'écoute du nœud (Yang et Kravets 2005).

La parallélisation des flux : un ensemble de nœuds appartenant à la zone d'écoute d'un même nœud peuvent émettre simultanément si aucun d'entre eux ne possède un autre nœud de cet ensemble dans sa zone d'écoute. Par exemple sur la figure 13, les nœuds A et E appartiennent à la zone d'écoute de C mais A n'appartenant pas à celle de E et E à celle de A, donc, A et E, selon le protocole DCF peuvent émettre simultanément.

Blocage d'un nœud (Yang et Kravets 2005)**:** Lorsqu'un nœud E accepte un flux avec un débit supérieur à la bande passante d'un nœud C et si ce nœud C possède dans sa zone d'écoute le nœud E, alors le nœud C devient congestionné et est obligé de bloquer des flux qu'il émet ou qui le traverse, c'est le phénomène de blocage de flux. La figure 13 illustre ce phénomène. Dans cet exemple, le canal est de 2 Mbit/s. Le nœud A possède C dans sa zone d'écoute, C possède A et E dans sa zone d'écoute et E possède C dans sa zone d'écoute. Le nœud A émet en premier le flux 1 à B 0.8 Mbit/s. Le nœud C ensuite émet à son tour un flux 2 à 0.8 Mbit/s au nœud D. E ne possédant que C dans sa zone d'interférence, sa bande passante locale disponible lui permet d'envoyer un flux 3 à 0.8 Mbits/s. Or, la transmission de E entraine la congestion du nœud C qui ne peut dès lors plus satisfaire les contraintes du flux 2. Afin d'éviter le phénomène de blocage de nœuds, les nœuds doivent

considérer la bande passante locale disponible de tous les nœuds dans leur zone d'écoute pour savoir s'ils peuvent accepter ou non un nouveau flux.



Nœud	BP locale
A	2
C	2
E	2

Flux 1

Nœud	BP locale
A	1.2
C	1.2
E	2

Flux 2

Nœud	BP locale
A	0.4
C	0.4
E	1.2

Flux 3

Nœud	BP locale
A	0.4
C	congestion
E	0.4

Figure 13 : Blocage du nœud E

3.2.2.2. *Les méthodes de contrôle d'admission non couplés avec un protocole de routage*

Selon la classification des méthodes de contrôle d'admission présentée dans la section précédente et illustrée par la figure 10, il existe deux types de CA ; les CAs couplés avec un protocole de routage et ceux non couplés avec un protocole de routage. Cette partie présente quelques modèles de CAs non couplés avec un protocole de routage.

DACME (Distributed Admission for Manets Environment) est un CA sans état non couplé avec un protocole de routage qui peut être implémenté sur des réseaux ad-hoc multi-sauts dont le protocole de routage est multi-chemins (Calafate, et al. 2007). Un protocole de routage multi-chemins route les paquets d'un flux sur plusieurs routes plutôt qu'une seule. La redondance des chemins permet de mieux répartir la charge du réseau et d'éviter un arrêt de la réception du flux lorsqu'un chemin n'est plus disponible. Les agents DACME sont situés au niveau de la source et de la destination, les nœuds intermédiaires sur les routes ne font que faire suivre les paquets. La source, une fois les routes établies par le protocole de routage, envoie N messages de découverte de ressources consécutivement sur chacune d'entre elles afin d'établir la BP disponible de chacune. La destination calcule la BP disponible (notée B) de chaque route en sommant la taille des $N-1$ paquets de

découverte reçus (notée P_{size}) et en divisant ensuite par le temps écoulé entre le premier et le dernier paquet reçu (noté AIT), ainsi :

$$\mathrm{B} = \frac{P_{size}}{AIT}(\mathrm{N}-1) \qquad \text{Équation 1}$$

La destination envoie ensuite à la source la BP qu'il a calculée pour chaque route. La source décide alors si la BP de l'ensemble des routes proposées par le protocole de routage multi-chemins est suffisante ou non pour admettre le nouveau flux. De nombreuses simulations ont été menées pour valider DACME. Les résultats montrent que ce protocole engendre une surcharge et un délai raisonnable et respecte les contraintes des flux. Cependant, les simulations sont limitées à des petits réseaux faiblement congestionnés. De plus, la méthode d'estimation des ressources de DACME est approximative, les auteurs ne précisent pas combien de paquets de découverte N doivent être envoyés pour avoir une bonne évaluation de la bande passante et si ce nombre doit évoluer selon la taille et la charge du réseau. De plus, l'évaluation de la BP d'une route ne prend pas en compte le fait que lorsque le flux est admis sur une route il l'est également sur d'autre(s) route(s). L'ensemble de ces routes peuvent posséder des nœuds partageant le même canal, ainsi la BP réelle de l'ensemble des routes peut se révéler être plus petite que celle calculée dû à l'interférence inter-routes.

BRAWN (Bandwidth Reservation in Ad-hoc Wireless Networks) est un CA avec état, découplé d'un protocole de routage pour les MANETs où les nœuds possèdent des débits différents (Guimares, et al. 2009). Dans BRAWN, un flux est admis sur une route si l'ensemble des nœuds de cette route peuvent satisfaire ses contraintes en termes de débit. Tout nœud doit donc pouvoir calculer sa BP disponible. Afin d'éviter le phénomène de blocage d'un nœud, le taux de BP disponible d'un nœud i, noté AB_i, dépend de son taux de BP locale disponible MAB_i et de celui de ses voisins à un saut :

$$AB_i = \min\{MAB_j\}, j \in N_i^+ \qquad \text{Équation 2}$$

avec N_i^+ représentant l'ensemble formé du nœud i et des nœuds voisins (à un saut de i). Le taux de BP locale qu'un nœud peut utiliser est limité à une valeur maximale notée Q choisie afin que le délai des flux ait une valeur acceptable. Le taux de BP locale disponible d'un nœud i équivaut au taux de BP maximale autorisée d'un nœud (Q) moins le taux de BP utilisé par ses voisins et lui-même :

$$MAB_i = \mathrm{Q} - \sum_{\mathrm{j} \in \mathrm{N_i^+}} \mathrm{X_j}$$ Équation 3

où $\mathrm{X_j}$ est la quantité de trafic normalisée transmise par le nœud j, c'est-à-dire la quantité de trafic transmis par j sur le débit du nœud j. Lors du CA d'un flux, chaque nœud considère la compétition intra-route et suppose qu'une transmission ait réussie sur un lien si aucun nœud voisin de l'émetteur et du destinataire n'émet. Un flux de débit r est accepté au niveau d'un lien dont le nœud émetteur est i et le nœud récepteur est j si :

$$AB_i \geq \sum_{y \in ((N_i^+ \cup N_j^+ \cap chemin\ )} \frac{r}{v_y}$$ Équation 4

avec v_y le taux de transmission du nœud y. Le CA est effectué après que la source ait découvert un chemin pour le flux. Elle envoie alors un message de demande de réservation de ressources (*Reservation Request*) au destinataire le long de la route. A la réception du message, chaque nœud intermédiaire vérifie s'il peut garantir les contraintes du flux en termes de BP via l'équation 4. S'il peut, il fait suivre le message et enregistre les données du flux sinon il abandonne le message. Lorsque le destinataire reçoit le *Reservation Request*, il envoie à la source un *Reservation Reply*. A sa réception, la source commence à émettre les données. Ce protocole a été validé par simulation sur ns2. Les résultats de simulation montrent que BRAWN atteint ses objectifs en termes de respects des exigences des flux, de diminution de la perte de paquets et de délai. Cependant, pour calculer la BP disponible d'un nœud, BRAWN considère uniquement une interférence à un saut, or, il est actuellement admis qu'une transmission sur un lien ait réussi si les voisins à au moins deux sauts de l'émetteur et du récepteur n'émettent pas (Iyer, Rosenberg et Karnik 2009).

3.2.2.3. Les méthodes de contrôle d'admission couplé avec un protocole de routage

Selon la classification des méthodes de CA illustrée par la figure 10, il existe deux types de CA ; les CAs couplés avec un protocole de routage et ceux non couplés avec un protocole de routage. Cette partie présente des CAs couplés avec un protocole de routage.

CACP (Yang et Kravets 2005) (Contention-aware Admission Control Protocol) est un CA couplé avec le protocole de routage DSR dans un MANET basé sur un accès au canal à compétition. Ce protocole est une référence dans le domaine, il est cité dans la majorité des articles de CA. CACP a été le premier protocole à considérer les phénomènes de contention intra-flux et de blocage de nœuds. Lors du CA d'un flux, chaque nœud j calcule sa BP disponible notée B_j en considérant le problème de blocage et évalue la BP que consommerait ce nouveau flux notée $B_c^f(j)$. Pour calculer sa BP disponible, chaque nœud évalue dans un premier temps sa BP disponible locale selon la formule suivante :

$$B_{loc,j} = \frac{T_{idle}}{T_p} * C \qquad \text{Équation 5}$$

avec T_{idle} le temps de silence du canal sur la période d'écoute T_p, et C la bande passante du canal. Afin d'éviter le blocage d'un nœud, la BP disponible d'un nœud j équivaut à la plus petite BP disponible locale parmi l'ensemble N_j^+, c.à.d. l'ensemble formé du nœud j et des nœuds situés dans sa zone d'écoute, ainsi :

$$B_j = \min\left(B_{loc,y}\right) \forall j \in N_j^+ \qquad \text{Équation 6}$$

Afin que chaque nœud puisse découvrir la BP locale des nœuds de sa zone d'écoute, les auteurs proposent 3 méthodes de découverte de ressources différentes, déclinant ainsi CACP en trois versions ; CACP-Multihop basé sur une méthode de découverte de ressources s'effectuant à la demande, CACP-Power basé sur une méthode de découverte de ressources s'effectuant sur des données locales et CACP-CS basé sur une méthode de découverte de ressources passive. Afin d'estimer la BP nécessaire à un flux le long d'une route, CACP considère la compétition intra-flux des nœuds. Il calcule la BP consommée par un flux f de débit r au niveau d'un nœud j notée $B_c^f(j)$ selon la formule suivante :

$$B_c^f(j) = \left|\text{Route} - \text{Destination} \cap \text{N}_\text{j}^+\right| * \text{r} \qquad \text{Équation 7}$$

avec N_j^+ l'ensemble formé de j et des nœuds situés dans sa zone d'écoute. Ainsi, la BP consommée au niveau d'un nœud j par un flux f équivaut au nombre de fois qu'il est émis dans la zone d'écoute de j par son débit r. Le CA d'un flux se déroule en deux phases. La première phase s'appelle le contrôle d'admission partiel ; la source envoie un paquet RREQ pour découvrir la route, un nœud qui

reçoit le message vérifie si sa BP locale est suffisante par rapport à la BP consommée du flux, sachant que cette dernière est calculée partiellement puisque seule une partie de la route a été préalablement découverte. Si elle est suffisante, il diffuse à son tour le RREQ. La seconde phase est appelée le CA total, pendant cette phase le destinataire envoie un RREP le long de la route suivie par le RREQ. Les nœuds intermédiaires connaissent cette fois entièrement la route empruntée par le flux et peuvent donc vérifier si leur BP disponible est supérieure à la BP que consommerait le flux à son niveau. Si elle est supérieure, alors, il fait suivre le RREP, sinon il le rejette. Le flux est totalement admis lorsque la source reçoit le RREP. CACP a été validé par analyse mathématique et simulation sur NS-2. Les résultats montrent qu'il est efficace en termes de surcharge et respecte les contraintes de flux. Cependant, CACP ne considère pas la parallélisation des flux ce qui entraine une sous-estimation de la BP disponible des nœuds.

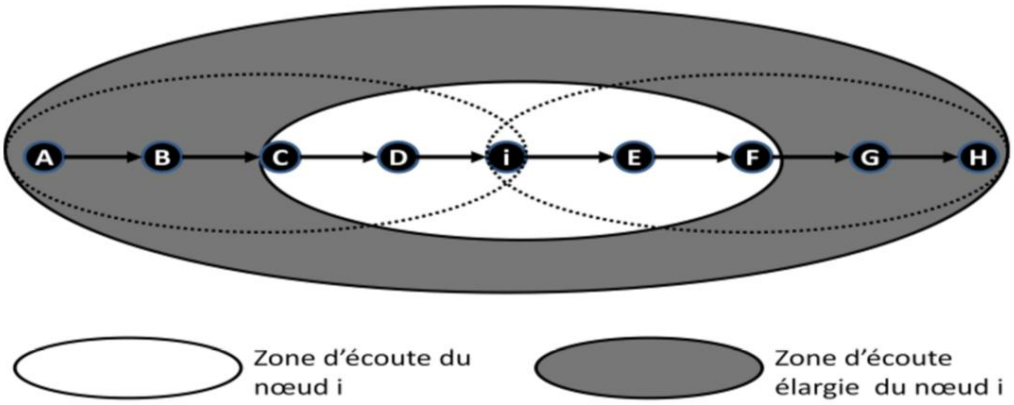


Figure 14 : Zone d'écoute et zone d'écoute élargie d'un nœud

MRCACP (Multi-rate and Contention Aware Admission Control) (Luo, et al. 2006) est un CA basé sur un protocole de routage réactif avec des entités à débits multiples. C'est un des premiers travaux à considérer la parallélisation des flux lors d'un CA. Chaque nœud i, lors de l'estimation du taux d'occupation $\rho_c^f(i)$ de son canal par un flux f, considère la contention intra-flux, c'est-à-dire le fait que le flux est émis par le nœud i et les nœuds de sa zone de portée d'écoute appartenant au chemin du flux. Cet ensemble est noté N_i^f, ainsi :

$$\rho_c^f(i) = \sum_{j=1}^{|N_i^f|} R.\left(\frac{L}{B_j} + T\right) \qquad \text{Équation 8}$$

avec L la taille d'un paquet, B_j le débit du nœud j, R le nombre de paquets émis par seconde par le flux et T le temps moyen d'attente et d'envoi de paquets de contrôle. Afin d'estimer leur taux de BP

disponible, les nœuds modifient, comme suggéré dans CACP-Power (Yang et Kravets 2005), la portée de leur zone d'écoute afin d'englober la zone d'écoute de l'ensemble des nœuds appartenant à leur zone d'écoute (voir figure 14). Cette nouvelle zone est la zone d'écoute élargie. Un nœud i obtient, en écoutant passivement sa zone d'écoute élargie sur une période de temps notée T_p, le temps d'occupation de cette zone $T_{busy}^{csn}(i)$ et en écoutant sa zone d'écoute originel sur une même période T_p, le temps d'occupation de son canal noté $T_{busy}^{local}(i)$. Il peut ainsi établir le taux d'utilisation de sa zone d'écoute élargie $\rho_{local}(i)$ et de sa zone d'écoute originel $\rho_{csn}(i)$:

$$\rho_{local}(i) = \frac{T_{busy}^{local}(\mathrm{i})}{T_p,} \text{ and } \rho_{csn}(i) = \frac{T_{busy}^{csn}(\mathrm{i})}{T_p,} \qquad \text{Équation 9}$$

La période $T_{busy}^{csn}(i)$ -$T_{busy}^{local}(i)$ représente le temps pendant lequel le nœud i peut émettre car sa zone d'écoute originelle est silencieuse. D'après la figure 14, le nœud i peut ainsi émettre simultanément avec A, B, G ou H. Ainsi, le taux de BP pendant lequel i peut émettre f en parallèle avec un nœud de sa zone d'écoute élargie est de :

$$\rho_{overlap}^{f}(i) = \frac{T_{busy}^{local} - T_{busy}^{csn}}{T_p} * R.(\frac{L}{B_i} + T) \qquad \text{Équation 10}$$

où $R.\left(\frac{L}{B_i} + T\right)$ représente le temps de transmission du flux par le nœud i. Un nœud i accepte un nouveau flux f, si son taux de BP disponible est suffisant, et s'il vérifie donc l'inéquation suivante :

$$\rho_{csn}(i) \geq \rho_c^f(i) - \rho_{overlap}^{f}(i) \qquad \text{Équation 11}$$

A la manière de CACP, le CA est partiel lors de la diffusion du RREQ par la source et est total lors de l'envoi par le destinataire du RREP. Bien que les auteurs considèrent la parallélisation de flux dans leur approche, la BP disponible des nœuds reste sous-évaluée. En effet, les auteurs ne considèrent pas le fait qu'un ensemble E de nœuds de la zone d'écoute d'un nœud i peuvent émettre simultanément un flux si, chaque nœud j de E n'appartient pas à la zone d'écoute d'aucun autre nœud de l'ensemble E, ainsi, par exemple sur la figure 14, les nœuds C et F peuvent émettre simultanément.

ACA (Admission Control Algorithm) (Shen, et al. 2009) est un CA pour les WMNs basé sur un schéma de routage réactif qui différentie les flux temps réels des flux non temps réels et où la découverte des ressources des nœuds s'effectue localement via une étude probabiliste considérant le phénomène des nœuds cachés dans le protocole DCF. Dans leur étude et contrairement aux travaux précédents (Calafate, et al. 2007) (Yang et Kravets 2005) (Guimares, et al. 2009), les auteurs considèrent qu'une transmission échoue sur un lien lorsqu'un nœud de la zone d'écoute de l'émetteur ou du récepteur émet simultanément avec le lien. Par l'observation du canal et en considérant son nombre de nœuds cachés et de nœuds dans sa zone d'écoute, chaque nœud évalue la probabilité qu'un des slots de sa zone d'écoute soit libre p_i, subisse une collision p_c ou transmette avec succès des données p_s . Un nœud peut dès lors en déduire le taux de silence de son canal R_i, le taux d'occupation de son canal R_b, et le taux d'utilisation du canal R_s après avoir estimé la durée d'une transmission réussie T_s, d'une transmission avec collision T_c et d'un slot σ selon le protocole DCF :

$$R_i = \frac{p_i \sigma}{p_i \sigma + p_s T_s + p_c T_c} \; et \; R_b = 1 - R_i, \quad et \quad R_s = \frac{p_s T_s}{p_i \sigma + p_s T_s + p_c T_c} \qquad \text{Équation 12}$$

Ainsi, un nœud peut estimer pour un taux d'occupation du canal donné R_b, le taux de données utiles envoyées sur son canal R_s. Ainsi, le taux de BP maximum notée BP_{max} utilisée pour transmettre des données avec succès sur son canal équivaut pour un nœud à la valeur de R_s lorsque son canal est totalement occupé, c.à.d. lorsque R_b=1. Ainsi, BP_{max} =R_s, lorsque R_b=1. Dans ACA, un nœud ne peut pas utiliser toute sa BP_{max} pour envoyer des données temps réel, un taux minimum de BP de BP_{max} est réservé aux paquets de contrôle, un autre taux aux flux non temps réel et un taux minimum de BP, B_{max} est dédié aux flux temps réel. Pour estimer le taux de BP consommée, noté B_c, par un flux f au niveau d'un nœud, les nœuds considèrent la contention intra-flux. Un flux temps réel sera admis le long d'une route si tous les nœuds de la route ont une bande passante maximum, disponible pour les flux temps réel, supérieure à B_c plus la BP déjà consommée par l'ensemble des flux temps réel admis au niveau de son canal :

$$R_b * c * R_{real} + B_c \leq B_{max} \qquad \text{Équation 13}$$

Avec R_{real} le taux de BP utilisé au niveau du nœud pour l'envoi de flux temps réel et C la capacité du canal. Le taux de BP envoyé pour les flux non temps réel est également ajusté selon la quantité de données temps réel.

3.2.3. Limites des solutions existantes de contrôle d'admission

Le rejet ou l'acceptation d'un flux dans tous les CAs présentés dans la partie précédente dépendent uniquement des ressources en BP des nœuds du réseau. L'admission sur la BP s'explique par la constatation suivante de Yang et *al.* (Yang et Kravets 2005) : tant que la BP demandée par un flux est respectée, son délai et sa gigue sont dès lors maitrisés. Les différentes solutions de contrôle d'admission existantes présentent chacune des caractéristiques différentes dont les principales sont résumées dans le tableau 3. D'après ce tableau, pour évaluer la BP, chaque protocole considère différents paramètres tels que la présence de nœuds cachés, la zone d'interférence, la contention intra-route, la parallélisation des flux. De plus, selon les protocoles de CA, la zone d'interférence considérée pour une transmission est différente. Or, définir correctement la zone d'interférence est très important puisque une mauvaise estimation de cette zone peut entrainer des problèmes de collisions et de perte de paquets si elle est trop petite, ou une sous-évaluation de la BP si elle est trop grande.

Tableau 3: Comparaison de protocoles de contrôle d'admission existants

	Contention intra-route	Nœuds cachés	Zone d'interférence	Blocage	PF	MD	MR
BRAWN (Guimares, et al. 2009)	Oui	Non	A 1 saut	Oui	Non	Oui	Non
DACME (Calafate, et al. 2007)	Non	Non	Non	Non	Non	Non	Oui
CACP (Yang et Kravets 2005)	Oui	Non	A deux sauts de l'émetteur	Oui	Non	Non	Oui
MRCACP (Luo, et al. 2006)	Oui	Non	A deux sauts de l'émetteur	Oui	En partie	Oui	Non
ACA (Shen, et al.	Oui	Oui	A deux sauts	Oui	Non	No	NON

2009)	de l'émetteur et du récepteur	n

Avec MD : multiples débits, MR : routage multi chemins, PF : parallélisation des flux

Tous ces protocoles utilisent un contrôle d'accès à compétition, DCF, or, ce dernier entraine une approximation du calcul de la bande passante d'un nœud puisque l'accès au réseau y est aléatoire à cause de l'algorithme de *backoff*. Les CAs permettent aux réseaux mesh de respecter les exigences des flux (si ces CAs ne sous-estiment pas la bande passante), cependant ils sont limités dans le nombre de flux pouvant être admis à cause de la faible capacité utile du réseau. Afin de pallier aux problèmes de DCF et au manque de BP des réseaux mesh, les chercheurs se sont intéressés ces dernières années aux protocoles de planification des liens. Gore et *al.* (Gore et Karandikar 2011) présentent un intéressant état de l'art sur les protocoles de planification de liens qui seront développés dans la section suivante.

3.3. La planification de liens

Les réseaux mesh souffrent de leur limitation en BP. Il existe de nombreuses méthodes afin d'améliorer leur BP comme principalement l'assignation de différents canaux aux nœuds mesh (Kapse et Shrawanakar 2011), l'utilisation d'antennes directionnelles (Zemin et Zhenglun 2011) et la planification de liens du réseau (Gore et Karandikar, Link Scheduling Algorithms for Wireless Mesh Networks 2011).

Les méthodes d'assignation de canaux (Kapse et Shrawanakar 2011), profitent du fait que chaque routeur mesh possède plusieurs antennes radio et peut donc envoyer et recevoir sur différents canaux. Ces méthodes assignent à chaque nœud, selon un algorithme pré défini, un ou plusieurs canaux afin qu'il puisse profiter de l'entière capacité de son canal et n'ait plus à le partager avec les autres nœuds de sa zone d'écoute. Cependant, les méthodes d'assignation de canaux se heurtent à la rareté des fréquences disponibles et sont donc peu déployées.

Les antennes directionnelles (Zemin et Zhenglun 2011) dans les réseaux mesh permettent d'améliorer la capacité globale du réseau ; elles vont être dirigées afin que chaque lien du réseau

puisse profiter de l'entière capacité de son canal. Cependant, le coût des antennes directionnelles est important, ce qui limite leur déploiement.

La planification des liens apparait donc comme une solution intéressante pour améliorer la BP des réseaux mesh car elle ne se heurte ni à des problèmes de coût ni à la rareté des fréquences disponibles (Pathak et Dutta 2011).

3.3.1. Présentation, objectifs et challenges de la planification de liens

La planification de liens a pour but d'assigner (en se basant sur un modèle d'interférence) à chaque nœud du réseau un ensemble de slots dans une fenêtre de temps qui se répète périodiquement, pendant ces slots le nœud peut émettre en évitant tout problème d'interférence. En évitant l'interférence, en se basant sur un accès temporel au médium et en maximisant le nombre de nœuds pouvant émettre à chaque slot (phénomène de réutilisation spatiale), la planification de liens améliore la capacité globale du réseau (Naouel Ben et Hubaux 2006).

En effet, les méthodes de planification de liens sont basées sur un accès temporel au médium. Les méthodes d'accès temporel au canal ne nécessitent pas de temps de *backoff*, diminuent l'utilisation des espaces inter-trame et évitent ainsi de longues périodes de temps de silence sur le canal. L'accès temporel au réseau permet ainsi de gagner en débit utile par rapport à un accès à compétition. La planification de liens profite également du phénomène de réutilisation spatiale ; ce phénomène permet à plusieurs nœuds d'un réseau mesh d'émettre simultanément, car l'atténuation des signaux avec la distance fait que le médium peut être réutilisé simultanément en plusieurs endroits différents sans pour autant provoquer de collisions (Gore et Karandikar, Link Scheduling Algorithms for Wireless Mesh Networks 2011). La planification de liens a trois principaux objectifs :

- l'envoi de données sans interférence,
- l'augmentation de la capacité utile du réseau,
- l'équité entre les nœuds, car chaque nœud peut envoyer la même quantité de données aux portails d'accès.

Les méthodes de planification de liens se basent sur un protocole d'interférence afin de planifier l'activation des liens en évitant les interférences. Un protocole d'interférence permet de prédire, en considérant l'ensemble des liens qui émettent simultanément, s'il va y avoir succès ou non de la

transmission sur ces liens. Il existe de nombreux protocoles d'interférence mais aucun ne permet d'éviter totalement les interférences (Iyer, Rosenberg et Karnik 2009), cependant, certains sont meilleurs que d'autres. Ainsi lors de la réalisation d'une méthode de planification de lien, la question suivante se pose : quel est le modèle d'interférence à choisir ? La planification de liens peut entrainer de nombreux calculs d'autant plus si le modèle d'interférence est complexe, il faut donc s'assurer de la réalisation en temps polynomial de la solution proposée (Sharma, Mazumdar et Shroff 2006) (Goussevskaia, Oswald et Wattenhofer, Complexity in geometric SINR 2007). Afin d'assurer l'équité entre les nœuds, c.à.d. que chaque nœud puisse envoyer la même quantité de données aux niveaux des portails (Ben Salem et Hubaux 2006), il faut considérer la géographie du réseau et savoir combien chaque routeur relaie d'informations pour d'autres routeurs. Ainsi la planification pose de nombreux défis en termes de:

- modèle d'interférence,
- complexité de calcul,
- connaissance de la topologie du réseau,
- synchronisation des nœuds.

3.3.1.1. Modèles d'interférences existants

La planification de liens repose sur un modèle d'interférence pour distribuer les slots pendant lesquels un nœud peut émettre sans risque de collision. Ainsi, il est important que le modèle d'interférence prédise au mieux l'interférence, c.à.d. s'il va y avoir interférence ou non sur un lien en considérant les activités courantes du réseau.

Un modèle d'interférence permet une conclusion binaire: le succès ou l'échec de la transmission d'un paquet sur un lien. En général, pour déduire si la transmission a réussi, le modèle d'interférence considère (Iyer, Rosenberg et Karnik 2009):

- le niveau de signal désiré par le récepteur du lien,
- le bruit thermique au niveau du récepteur du lien. Le bruit thermique est induit par l'agitation des électrons au sein d'un équipement électronique (Stallings 2007).
- la puissance des signaux envoyés par les nœuds du réseau lors de la transmission.

Il n'existe pas actuellement dans la littérature de consensus sur le modèle d'interférence optimal à utiliser (Iyer, Rosenberg et Karnik 2009). Les modèles d'interférence les plus utilisés sont le modèle d'interférence additive (Gupta et Kumar 2000), le modèle de capture à seuil (Mccanne, Floyd et Fall

s.d.) (Iyer, Rosenberg et Karnik 2009), le modèle protocolaire d'interférence (Iyer, Rosenberg et Karnik 2009) et le modèle d'interférence à K-sauts (Sharma, Mazumdar et Shroff 2006). Afin d'expliquer brièvement chacun de ces modèles, on suppose la situation suivante ; un ensemble Γ de nœuds u_i émettent simultanément avec une puissance P_{u_i}, parmi eux, le nœud u_1 souhaite envoyer des données à r_1. Le gain du signal entre un nœud $u_i \in \Gamma$ et le nœud r_1, noté G_{u_i}, représente le rapport entre la puissance à laquelle le nœud r_1 reçoit le signal de u_i et la puissance à laquelle u_i envoie le signal. Le gain G_{u_i} est inversement proportionnel à la distance séparant u_i et r_1, notée $|u_i - r_1|$. On note P_N la puissance du bruit thermique au niveau du nœud r_1. Chaque modèle a pour but de prédire si la transmission entre u_1 et r_1 va réussir ou non.

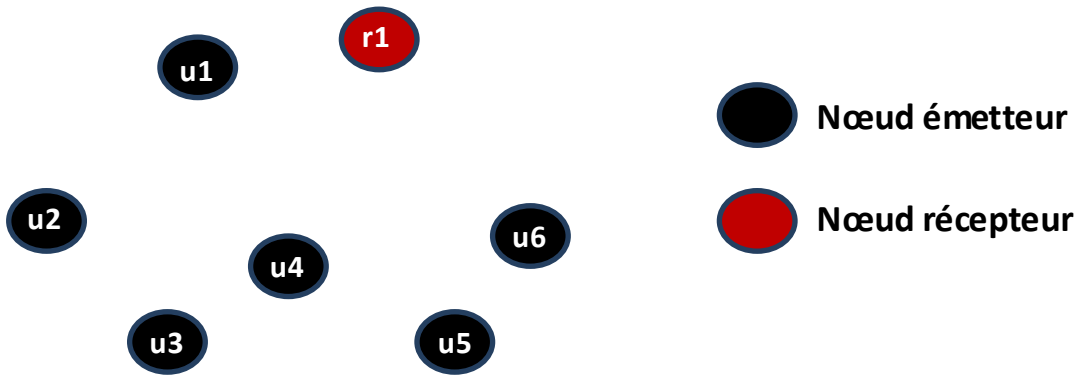


Figure 15 : les protocoles d'interférences ont pour but de prédire si le nœud r_1 reçoit avec succès le signal en provenance du nœud u_1 sachant que l'ensemble des nœuds activés simultanément est $\Gamma = \{u_1, u_2, u_3, u_4, u_5, u_6\}$

Modèle d'interférence additif : Dans ce modèle, lorsqu'une communication est en cours sur un lien, tous les nœuds qui émettent simultanément avec l'émetteur du lien vont, quelle que soit leur distance par rapport au récepteur du lien, impacter sur le résultat de la transmission (Gupta et Kumar 2000). Le modèle d'interférence additif est basé sur le signal sur interférence plus bruit ou SINR (Signal to Interference plus Noise Ratio). Chaque nœud récepteur doit avoir un signal sur interférence plus bruit supérieur à un certain seuil. Le SINR au niveau de r_1 est calculé avec la formule suivante :

$$\gamma_{r_1} = \frac{G_{u_1} P_{u_1}}{P_N + \sum_{u_i \in \Gamma - \{u_1\}} G_{u_i} P_{u_i}} \qquad \text{Équation 14}$$

Le nœud r_1 reçoit avec succès les données de e_1 si :

$$\gamma_{r_1} \geq \beta_{r_1} \qquad \text{Équation 15}$$

avec γ_{r_1} défini par l'équation 14 et β_{r_1} le seuil SINR. La valeur du seuil SINR dépend de la carte réseau du récepteur ainsi que du débit et de la modulation du paquet à décoder. Le récepteur r_1 peut décoder un paquet (codé avec un certain débit et une certaine modulation) si la valeur du SINR à laquelle il reçoit le paquet est supérieure à β_{r_1} (Reis, et al. 2006).

Modèle de capture à seuil : le modèle de capture à seuil est le modèle d'interférence utilisé par l'un des simulateurs réseau les plus populaires, ns2 (Mccanne, Floyd et Fall s.d.). ns2 considère que pour un ensemble de nœuds émettant simultanément avec un lien, chaque nœud interfère avec le lien indépendamment des autres nœuds. Ainsi, le modèle vérifie si chaque nœud appartenant à l'ensemble Γ -$\{u_1\}$ n'interfère pas avec le lien (u_1, r_1). ns2 utilise deux seuils, le seuil de capture $CpThresh$ et le seuil de réception $RThresh$. L'émission sur le lien (u_1, r_1) est considérée réussie si :

$$G_{u_1} P_{u_1} \geq \mathrm{RxThresh} \quad \text{et} \qquad \text{Équation 16}$$

$$\frac{G_{u_1} P_{u_1}}{G_{u_i} P_{u_i}} \geq \mathrm{CpThresh} \quad \forall e_i \in \Gamma \text{-}\{e_1\} \qquad \text{Équation 17}$$

Modèle protocolaire d'interférence: ce modèle, comme le modèle de capture à seuil, considère que pour un ensemble de nœuds émettant simultanément avec un lien, chaque nœud interfère avec le lien indépendamment des autres nœuds. La transmission dans ce modèle est réussie si la distance entre l'émetteur et le récepteur du lien $|u_1 - r_1|$ est inférieure à un seuil R_e (rayon de la zone d'émission) et si la distance entre chaque nœud interférant et le récepteur du lien est supérieure à un certain seuil proportionnel à la distance récepteur-émetteur $|u_1 - r_1|$:

$$|u_1 - r_1| \leq R_c \text{ et} \qquad \text{Équation 18}$$

$$|u_i - r_1| \geq (1 + \Delta)|u_1 - r_1|, \forall u_i \in \Gamma \text{-}\{u_1\} \qquad \text{Équation 19}$$

Avec Δ un paramètre positif. $(1 + \Delta)|u_1 - r_1|$ représente le rayon de la zone d'interférence du lien centré sur le récepteur r_1.

Dans (Iyer, Rosenberg et Karnik 2009), les auteurs prouvent la similarité du modèle de capture avec le modèle protocolaire d'interférence dans un environnement où la perte du signal est isotrope et la puissance de transmission, de modulation et de codage sont les mêmes pour tous les nœuds. De nombreux articles se basent sur un tel environnement pour construire leurs approches (Iyer, Rosenberg et Karnik 2009).

Modèle d'interférence à K-sauts (Sharma, Mazumdar et Shroff 2006): ce modèle considère qu'il y a interférence si un nœud situé à moins de K sauts du récepteur émet simultanément avec l'émetteur du lien. Ce modèle est finalement une simplification du modèle protocolaire d'interférence où le rayon d'interférence équivaut à K fois le rayon d'émission. Généralement K est fixé à 2.

Ces modèles d'interférence présentent des complexités différentes. Le plus complexe est le modèle d'additivité puisque, pour savoir si la transmission sur un lien est réussie ou non, il faut considérer l'ensemble des nœuds actifs simultanément, peu importe leur distance avec le récepteur du lien (Gore et Karandikar, Link Scheduling Algorithms for Wireless Mesh Networks 2011). Les autres protocoles considèrent l'interférence de chaque nœud isolément et la calcule avec plus ou moins de précision. Chacun de ces protocoles peut être considéré finalement comme une simplification du protocole de capture à seuil (Iyer, Rosenberg et Karnik 2009). Dans (Iyer, Rosenberg et Karnik 2009), les auteurs ont montré par simulation que la bande passante d'un réseau mesh basée sur le protocole DCF est bien plus importante (environ 3 fois) lorsqu'un modèle de capture à seuil est utilisé plutôt qu'un modèle d'additivité. Ainsi, l'utilisation d'un modèle d'interférence par rapport à un autre peut avoir un impact très important sur les résultats de la simulation.

Mheswari et *al.* (Maheshwari, Jain et Das 2009), ont montré expérimentalement l'additivité de l'interférence et ont également comparé expérimentalement, sur un réseau mesh de 20 nœuds basé sur l'IEEE 802.15.4, le taux d'erreur des différents modèles d'interférence pour prédire l'interférence: leurs résultats montrent que le modèle d'additivité possède le plus faible taux d'erreur. Ainsi le modèle d'additivité est le protocole qui modélise le mieux l'interférence. Il est donc important de se baser sur ce protocole plutôt que sur un autre car ces différents protocoles ont des résultats au niveau simulation très différents les uns des autres.

3.3.2. Solutions existantes de planification des liens

Les différentes solutions existantes de planifications de liens peuvent être classées selon le protocole d'interférence qu'elles utilisent. Afin de comprendre les solutions existantes de planifications de liens, il est indispensable d'introduire les concepts suivants.

3.3.2.1. *Concepts utilisés dans les solutions de planification de liens*

Fenêtre de planification : la fenêtre de planification est une période de temps découpée en N slots, qui se répète dans le temps (voir figure 16). Les algorithmes de planification choisissent, dans cette fenêtre les slots, qu'ils vont attribuer aux différents liens du réseau.

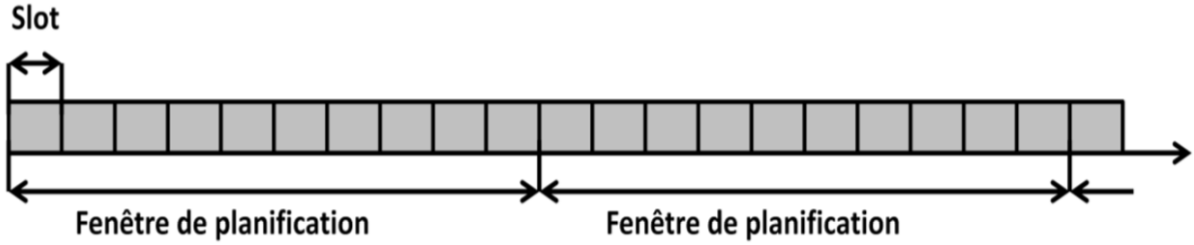


Figure 16 : Décomposition du temps en fenêtres de planification, chaque fenêtre est décomposée en N slots avec N=10

Minimisation de la fenêtre de planification : la minimisation de la fenêtre de planification consiste à associer à chaque lien du réseau un certain nombre de slots tout en minimisant la taille de la fenêtre de planification et en s'assurant qu'il n'y a pas d'interférences entre les liens. En d'autres termes, la minimisation de la fenêtre de planification permet de minimiser la période nécessaire pour que l'ensemble des liens envoie une certaine quantité de données et donc de maximiser la BP du réseau (Brar, Blough et Santi 2006) (Goussevskaia, Oswald et Wattenhofer, Complexity in geometric SINR 2007). Ainsi, la minimisation de la fenêtre de planification a pour but de maximiser l'utilisation de la BP du réseau.

Lien : si un nœud émetteur, noté u, peut envoyer avec succès des données à un nœud récepteur, noté r, lorsqu'aucun autre nœud dans le réseau n'émet déjà, alors la paire de nœuds (u, r) forme un lien.

Un graphe de communication : un graphe de communication est un graphe dirigé, noté $G(V, E)$, où V représente l'ensemble des nœuds du réseau mesh et E l'ensemble des liens du réseau.

Un graphe de conflits (Gore et Karandikar, Link Scheduling Algorithms for Wireless Mesh Networks 2011)**:** un graphe de conflits peut uniquement exister lorsque le modèle d'interférence utilisé n'est pas additif. Un graphe de conflit est noté $G_c(V_c, E_c)$. V_c représente l'ensemble des sommets du graphe. Chaque sommet $v \in V_c$ est un lien pour le graphe G(V,E), ainsi $v = (u_1, u_2) \in V_c$ si $(u_1, u_2) \in E$. E_c représente l'ensemble des liens du graphe. Un lien $(v_1, v_2) \in E_c$ si, selon le modèle d'interférence, la transmission sur v_1 ou sur v_2 échoue lorsque v_1 et v_2 sont actifs simultanément.

Problème NP-complet (Garay et David 1990): un problème pour lequel on peut vérifier en temps polynomial une solution à ce problème mais pour lequel on ne sait pas trouver de solution efficacement (en temps polynomial).

3.3.2.2. Méthodes de planification de liens basées sur l'interférence à K-sauts

Dans ce qui suit, nous présentons quelques méthodes de planification de liens basées sur le modèle d'interférence à K-sauts.

Dans (Jian, Jie et Ying-you 2010), les auteurs proposent une solution de planification de liens centralisée dans un réseau mesh basée sur un algorithme génétique multi-objectifs. Leur méthode de planification de liens a deux objectifs ; minimiser la taille de la fenêtre de planification (afin de maximiser la BP) et minimiser le délai moyen de l'ensemble des routes du réseau mesh. Pour atteindre ces deux objectifs, ils proposent deux algorithmes : OLLS (Ordered Link List Scheduling) et OLSBG (Optimal Link Scheduling Based on NSGA-II). OLLS est un algorithme qui associe, à tous les liens d'une liste ordonnée de liens, un ensemble de slots contigus en se basant sur le modèle d'interférence. Cet algorithme a la propriété d'attribuer à chaque lien des slots d'autant plus près du début de la fenêtre de planification que le lien est au début de sa liste ordonnée. OLSBG est basé sur NSGA-II (Deb, et al. 2002), l'un des algorithmes génétiques multi-objectifs les plus populaires ; il permet de trouver, suite à plusieurs itérations, la liste de liens ordonnés qui va permettre à OLLS de fournir une planification de liens qui minimise le délai moyen des flux du réseau et maximise sa BP. La solution proposée est basée sur le modèle d'interférence à K sauts avec K=2. Ainsi les auteurs estiment qu'un lien peut émettre sans conflit si aucun nœud à un saut du récepteur du lien n'émet simultanément avec l'émetteur du lien. La solution a été validée par simulation sur à la fois un réseau mesh à topologie chainée et maillée et a été comparée avec d'autres méthodes de planifications de liens. Les résultats de simulation montrent que leur solution permet de réduire la taille de la fenêtre de planification et le délai moyen des flux par rapport à d'autres solutions existantes. Cependant, leur solution souffre de nombreuses lacunes. Tout d'abord, elle repose sur un modèle d'interférence peu fiable, ainsi les résultats obtenus par le simulateur sont probablement très différents de ceux qu'on observerait dans la réalité. De plus les auteurs ne précisent pas comment leurs algorithmes pourraient être implémentés au sein d'un réseau multi sauts.

Dans (Sharma, Mazumdar et Shroff 2006), les auteurs s'intéressent au problème de maximisation du nombre de liens actifs simultanément dans un slot. Ils démontrent que ce problème généralisé à tous

les modèles d'interférence à K sauts peut être réduit à un problème de couplage K-valide maximum noté MKVMP (*Maximum K-Valid MatchingProblem*). Après avoir prouvé que ce problème est NP-complet lorsque $K \geq 2$, ils proposent plusieurs algorithmes d'approximation résolvant ce problème en temps polynomial. Les auteurs démontrent que, en utilisant le modèle d'interférence à K sauts, le problème de maximisation de la BP lors d'une planification de liens se réduit à trouver les ensembles maximum de liens qui peuvent émettre simultanément sans risque d'interférence. Les auteurs modélisent ce problème via un graphe de communication G(V,E). Le problème revient alors à déterminer les couplages maximums K-valide du graphe et ainsi à résoudre MKVMP. Un couplage K-valide, noté M, est un ensemble de liens sur le graphe qui sont tous à une distance de K sauts les uns des autres et qui peuvent donc émettre simultanément sans risque d'interférence. Ainsi deux liens $e_1 = u_1u_2$ et $e_2 = v_1v_2$ avec $e_1 \neq e_2$, appartiennent à un couplage K-valide si :

$$d(e_1, e_2) = \min_{i,j\in\{1,2\}}(d_s(u_i, v_i)) \quad \text{et} \qquad \text{Équation 20}$$

$$d(e_1, e_2) \geq K \qquad \text{Équation 21}$$

avec $d_s(x, y)$ le plus petit nombre de liens qui sépare les nœuds x et y avec $x, y \in V$. Le problème de couplage K-validemaximum revient à trouver les ensembles de couplage K-valide tels que leur cardinalité soient maximum, c'est-à-dire que l'on ne puisse plus ajouter un nouveau lien à l'un de ces ensembles sans qu'au moins deux liens de ce dernier ne respectent plus l'équation 21. Les auteurs présentent par la suite plusieurs algorithmes centralisés permettant de résoudre ce problème. Ils valident ensuite, par simulation, leurs algorithmes avec différentes valeurs de K et montrent que la valeur la plus adaptée pour K dépend de la couche physique du réseau. Cet article est un début d'ébauche d'un système de planification de liens ; cependant, les auteurs ne présentent aucune méthode permettant l'application de leurs algorithmes dans un réseau mesh.

3.3.2.3. Solutions basées sur le modèle d'interférence additif

Dans (Goussevskaia, Oswald et Wattenhofer, Complexity in geometric SINR 2007), les auteurs prouvent analytiquement que le problème de planification de l'ensemble des liens d'un réseau selon le modèle d'interférence additif avec minimisation de la taille de la fenêtre de planification est NP-complet. Les auteurs proposent un algorithme approximatif de planification de liens basé sur une

réduction du problème. Cet algorithme répartit les liens dans des ensembles. Chaque ensemble, noté R_i, cet regroupe les liens du réseau dont la longueur l est comprise entre $2^i \leq l \leq 2^{i+1}$. Ainsi, chaque lien appartient à un ensemble R_i tel que, $R_i \in R = R_0, \dots R_{l_{max}}$. Pour chaque ensemble R_i non vide, l'espace est partitionné en carré de longueur $\mu 2^i$, avec μ une valeur fixe. Chaque carré (ou cellule) est ensuite colorié de telle manière qu'aucun ne soit de la même couleur qu'un carré adjacent et que le nombre de couleurs utilisées ne soit pas supérieur à quatre. Un lien appartient à une cellule si son récepteur est dans la cellule. Puis, l'algorithme choisit aléatoirement, dans chaque cellule d'une même couleur issue de la partition d'un ensemble R_i , un lien encore non planifié. L'ensemble des liens sélectionnés sont associés à un même nouveau slot. Par exemple, d'après la figure 17, si l'algorithme choisit un lien par carré jaune, représenté par une flèche sur la figure, alors on associe tous ces liens à un même slot.

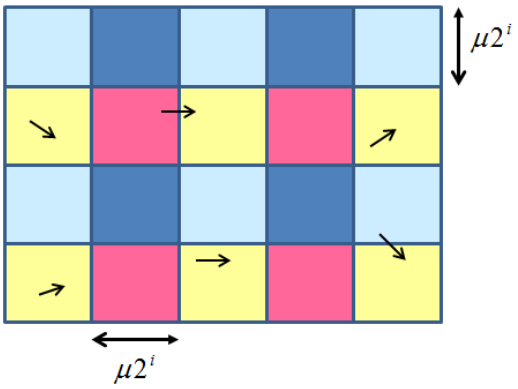


Figure 17 : Division de l'espace de l'ensemble R_i. Un lien par carré jaune est associé à un même slot.

On recommence la procédure jusqu'à ce que tous les liens de $R_i$soient planifiés puis que tous les liens de tous les ensembles R_i de R le soient également. Si on note Δ le nombre maximum de liens dans une cellule, sachant qu'il y a quatre couleurs et que le nombre d'ensembles est |R|, alors la taille de la fenêtre de planification est au maximum de $\Delta * |R| * 4$. Par la suite, les auteurs montrent que l'algorithme génère une planification sans interférence selon le modèle additif. La valeur de |R| dépend du nombre de liens et peut s'exprimer sous la forme d'une fonction g(L) avec L l'ensemble des liens du réseau. Les auteurs prouvent analytiquement que leur algorithme approxime la solution optimale du problème avec un taux $O(g(L))$. Cet article présente ainsi un algorithme permettant de résoudre approximativement le problème de planification de lien basé sur un modèle d'interférence additif. Cependant, il ne précise pas comment intégrer cet algorithme dans un réseau, de plus il considère que tous les liens supportent la même charge réseau ce qui est rarement le cas dans un réseau mesh multi-sauts, certains liens étant plus sollicités que d'autres.

Dans (Wan, Xu et Frieder 2010), les auteurs proposent deux algorithmes d'approximation pour résoudre le problème de la minimisation de la fenêtre de planification des liens (SLS pour *Shortest Link Schedule*) dans un réseau ad-hoc multi sauts. Le premier algorithme permet de résoudre le problème de la minimisation de la planification de liens dans un réseau mesh où l'ensemble des nœuds possèdent la même puissance d'émission, et le second algorithme permet de résoudre le problème SLS dans un réseau mesh où l'ensemble des nœuds sont capables de modifier leur puissance d'émission. Ils peuvent alors choisir leur puissance, cette dernière doit être comprise dans un ensemble P de valeurs possibles. Le problème de la minimisation de la fenêtre planification de liens consiste à trouver la planification de liens possédant la plus petite fenêtre de planification pour un ensemble E' de liens appartenant à l'ensemble E des liens du réseau. Les auteurs résolvent ce problème via un algorithme d'approximation glouton qui planifie les liens, slot après slot cet algorithme va faire appel à un second algorithme qui diffère selon si le réseau mesh est basé sur de nœuds avec puissance de contrôle ou non. Cet algorithme glouton associe, à chaque slot, un ensemble maximum de liens indépendants (MLSI *Maximum Independant Set Of Links*) dans E' qui n'ont encore jamais été planifiés. L'algorithme s'arrête quand tous les liens ont été planifiés. Un ensemble maximum de liens indépendant (MISL) est le plus grand ensemble I de liens indépendants, c.à.d. qui peuvent émettre simultanément sans collision selon le modèle d'interférence additif, dans un sous ensemble E'. Afin de trouver un ensemble maximal de liens indépendants dans E' qui n'ont encore jamais été planifiés, l'algorithme glouton fait soit appel à l'algorithme approximatif proposé dans (Wan, Xiaohua et Frances 2009) si le réseau est composé de nœuds ne pouvant pas contrôler leur puissance ou leur propre algorithme approximatif sinon. Ainsi, cet article présente deux algorithmes gloutons dont l'un résout le problème SLS dans un réseau mesh sans puissance de contrôle avec une approximation en $O(ln(\alpha))$ avec α le nombre d'ensembles de liens indépendants I existants dans le réseau, et un second qui résout le problème SLS dans un réseau mesh avec puissance de contrôle avec une approximation en $O(\beta ln(\alpha))$. Le paramètre β est la plus petite valeur k telle qu'il existe une partition de P en k sous-ensembles dont les éléments diffèrent au maximum d'un facteur de 2, ainsi :

$$\beta \leq 1 + log \frac{\min\limits_{p \in P} p}{\max\limits_{p \in P} p} \qquad \text{Équation 22}$$

Chaque lien, dans leurs algorithmes, est associé aux mêmes nombres de slots, or un lien dans un réseau mesh peut avoir une charge différente selon la quantité de flux qui le traverse ou celle qu'il envoie.

3.3.3. Limitations des solutions existantes de planification des liens

Les approches présentées ci-dessus, comme la majorité des approches de planification de liens existantes, ne font que proposer des algorithmes approximatifs permettant de résoudre un problème de planification de liens souvent NP-complet. Le tableau suivant résume les approches étudiées :

Tableau 4 : Tableau comparatif de systèmes de planification existants

	Problème	Modèle d'interférence	Type de problème	Approximation algorithmique
(Wan, Xu et Frieder 2010)	Trouver la planification de liens qui permet de minimiser la taille de la fenêtre de planification telle qu'un ensemble de liens L puissent émettre	Modèle d'interférence additif	NP complet	Avec puissance de contrôle : $O(\beta ln\alpha)$* Sans puissance de contrôle $O(ln\alpha)$*
(Goussevskaia, Oswald et Wattenhofer, Complexity in geometric SINR 2007)	Trouver la planification de d'un ensemble L de liens qui permet de minimiser la fenêtre de planification	Modèle d'interférence additif	NP complet	O(g(L))
(Jian, Jie et Ying-you 2010)	Trouver la planification de l'ensemble des liens d'un ensemble de chemins P tel qu'elle minimise le délai d'un flux sur chaque chemin et la taille de la fenêtre de planification	Modèle à deux sauts	NP complet	Inconnu
(Sharma, Mazumdar et Shroff 2006)	Maximiser le nombre de liens pouvant transmettre simultanément dans un	Modèle d'interférence à K-sauts	NP complet	Inconnu

slot

*variables expliquées lors de la présentation de l'article

Seul l'article (Jian, Jie et Ying-you 2010) considère les chemins des flux et les différentes demandes des liens en BP, alors que les autres travaux assignent à chaque lien une même quantité de slots. L'ensemble des articles présentés précédemment ne proposent aucune méthode pour implémenter ces algorithmes dans des réseaux mesh multi sauts, c.à.d. qu'aucun ne précise, par exemple, comment, une fois la planification effectuée, celle-ci est diffusée à travers le réseau pour que chaque lien sache à quel slot émettre ou comment, les demandes en BP, arrivent jusqu'au nœud qui planifie l'ensemble des liens. De plus, aucune de ces solutions ne modifie la planification des liens selon la charge du réseau, afin de pouvoir, par exemple, accorder plus de slots aux liens qui sont temporairement chargés et moins, à ceux qui le sont peu.

3.4. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art des contrôles d'admission et des méthodes de planification de liens existants dans les réseaux ad-hoc multi-sauts. Un contrôle d'admission permet de respecter les exigences des flux admis dans le réseau principalement en termes de bande passante et de délai. Cependant, les CAs existants souffrent d'une mauvaise approximation de la bande passante disponible induite par l'accès à compétition au réseau ; aucun nœud ne peut prédire avec précision la bande passante qu'il pourra par la suite obtenir puisqu'elle dépend de l'issue d'une compétition. De plus, les contrôles d'admission sont limités dans le nombre de flux qu'ils peuvent admettre dans le réseau car les réseaux mesh ont une faible capacité. Cette faible capacité est induite entre autre par l'accès à compétition au canal et à la perte de paquets. L'accès à compétition au canal entraine des risques de collisions, l'envoi de paquets RTS/CTS et de nombreux temps de silence qui peuvent être évités avec un accès temporel au réseau.

Les méthodes de planification de liens permettent d'augmenter la capacité utile du réseau en évitant les interférences et en se basant sur un accès temporel au médium. Cependant, les solutions existantes de planification de liens proposent généralement, uniquement des algorithmes de planification et aucun cadre général pour leur déploiement sur un réseau mesh. De plus, les méthodes de planification de liens associent à chaque lien du réseau le même nombre de slots. La planification reste fixe dans le temps et n'évolue pas selon la charge du réseau, ce qui peut entrainer

des congestions. Ces dernières pourraient être évitées si les slots étaient répartis à travers les liens selon la dynamique du réseau. Un lien fortement chargé devrait obtenir plus de slots qu'un lien faiblement chargé et le nombre de slots attribués à un lien devraient évoluer selon sa charge. Afin de pallier le manque de dynamique des méthodes de planification de liens existantes, la faible capacité utile et la mauvaise prédiction en bande passante disponible des CA existants, nous proposons dans cette thèse un contrôle d'admission avec planification de liens. L'idée est de pallier le manque de capacité utile des solutions de CA ainsi que leur difficulté à prédire leur BP disponible en y intégrant une solution de planification de liens, et de pallier le manque de dynamique des solutions de planification des liens en l'associant à un CA. Le but de cette solution est de respecter les exigences d'un grand nombre de flux à fortes contraintes dans un réseau mesh tout en améliorant la capacité utile du réseau et en diminuant la perte de paquets.

Chapitre 4. Contrôle d'admission avec planification des liens

4.1. Introduction

Le chapitre précédent présente comment les solutions de contrôle d'admission et de planification de liens permettent d'améliorer la qualité de service d'un réseau mesh. Cependant, il souligne également leurs limites en mettant en évidence le nombre restreint de flux que les contrôles d'admission peuvent accepter à cause de la capacité limitée d'un réseau mesh. Il soulève également le problème des contrôles d'admission existants en termes de sous-estimation ou surestimation des ressources du réseau pouvant entrainer respectivement soit une perte des ressources soit une congestion du réseau. Il montre que les solutions de planifications de liens existantes ne s'adaptent généralement pas à la charge du réseau et limitent sa dynamique. Ce chapitre introduit notre nouveau modèle de contrôle d'admission avec planification des liens (Dromard, Khoukhi et Khatoun, An Admission Control Scheme Based on Transmission Scheduling for Wireless Mesh networks 2012) (Dromard, Khoukhi et Khatoun, An Admission Control Scheme Based on Links' Activity Scheduling for Wireless Mesh Networks 2012) qui a pour objectifs principaux de respecter les contraintes des flux, d'augmenter la capacité du réseau et ainsi le nombre de flux admis. Notre solution propose de recalculer la planification des liens à chaque admission d'un nouveau flux. Ainsi, le système de planification évolue avec la charge du réseau, et le contrôle d'admission profite de l'augmentation de la bande passante utile et de l'estimation rigoureuse des ressources du réseau induite par la planification. Ce chapitre se termine sur l'évaluation de notre solution.

4.2. Objectifs de notre contrôle d'admission avec planification des liens dans un réseau mesh

Un contrôle d'admission a pour but d'accepter ou de rejeter un nouveau flux selon si le réseau est capable de supporter ses contraintes ainsi que les contraintes des flux préalablement admis. Le chapitre précédent présente le fonctionnement et les objectifs d'un CA dans un réseau mesh, ainsi que nombreux protocoles de CA existants (Calafate, et al. 2007) (Guimares, et al. 2009) (Yang et

Kravets 2005) (Luo, et al. 2006) (Shen, et al. 2009). Il met également en évidence les problèmes des Cas existants tel que :

- le problème de l'estimation des ressources et tout particulièrement de la bande passante pouvant entrainer une sous-utilisation des ressources du réseau ou des problèmes de congestion. La plupart des CAs actuels dans les réseaux mesh sont basés sur un accès à compétition au réseau, il est donc impossible de pouvoir estimer avec précision le débit que possédera un nœud, car la valeur du débit dépend de l'issue de la compétition pour l'accès au réseau, issue qui est aléatoire. Or, une mauvaise estimation de la bande passante peut avoir de graves conséquences. Si la bande passante est sous-estimée, le CA peut refuser d'admettre des flux qui auraient pu être acceptés par le réseau. En revanche, si la BP est surestimé, alors le CA peut admettre plus de flux que le réseau peut supporter entrainant une congestion du réseau.
- le problème du faible nombre de flux admis sur le réseau. Les CAs sont limités dans le nombre de flux qu'ils peuvent admettre par la faible capacité des réseaux mesh. La bande passante utile est d'autant plus limitée que la majorité des CAs existants utilisent un contrôle d'accès à compétition au canal. Or, les contrôles d'accès à compétition induisent de nombreux temps d'espace inter-trame, de *back-off* et de reprise suite à des collisions qui diminuent d'autant le débit utile des nœuds.

Afin de pallier aux problèmes rencontrés dans la majorité des CAs existants (problème du faible nombre de flux admis et problème d'estimation des ressources), nous proposons une solution de contrôle d'admission intégrant un système de planification de liens. La planification de liens permet de résoudre à la fois le problème du manque de précision de l'estimation de la BP disponible des nœuds et du faible nombre de flux admis dans le réseau induit par l'utilisation d'un accès à compétition au canal offrant un faible débit utile au nœud. Un système de planification de liens utilise un accès temporel au réseau, chaque nœud du réseau est associé à un certain nombre de slots pendant lesquels il peut émettre sans risque d'interférence. Ainsi, la BP utilisée par un nœud est maitrisée puisqu'elle correspond à la capacité du canal par la durée pendant laquelle le nœud peut émettre. De plus, la planification de liens augmente le débit utile des nœuds car elle évite, entre autre, la collision entre les nœuds, le temps perdu induit par l'accès à compétition au canal, l'algorithme de *backoff* et l'envoi de paquets RTS-CTS. La planification des liens permet également de calculer le délai des flux car on connait dès lors quand un nœud de la route émet un paquet pour le flux. Ainsi, en connaissant les slots pendant lesquels un flux est émis sur chaque nœud de sa route, le délai du flux peut être calculé. Notre solution de CA avec planification dynamique des liens a pour objectifs :

- de garantir la qualité de service des flux admis en termes de bande passante et de délai.
- d'augmenter la capacité globale du réseau et ainsi le nombre de flux admis sur le réseau.

Actuellement, la majorité des travaux sur la planification de liens propose uniquement des algorithmes de planification de liens assignant à chaque nœud du réseau une même quantité de slots ; ils ne considèrent donc pas la demande des nœuds en termes de bande passante. L'originalité de notre solution réside dans :

- la formulation du problème d'admission d'un flux dans un réseau mesh avec planification des liens sous forme de problème de programmation linéaire en variables binaires,
- une preuve de la NP-complétude du problème d'admission d'un flux dans un réseau mesh avec planification des liens,
- un algorithme de calcul du délai des flux dans un réseau mesh avec planification de liens,
- un algorithme permettant de résoudre le problème d'admission en termes de délai et de BP d'un nouveau flux et de sa planification,
- l'intégration de la cette algorithme dans un contrôle d'admission basé sur la bande passante et le délai.

4.3. Modélisation du réseau et problématique

Notre réseau mesh est modélisé dans un plan par un graphe étiqueté et orienté $G(V, E, f)$ où V représente l'ensemble des nœuds du réseau, E l'ensemble des liens du réseau et f une fonction telle que $f: E \rightarrow \mathbb{R}$. Tous les liens sont orientés, c.à.d. si (u, v) est un lien de E et si le lien (v, u) existe, $(u, v) \neq (v, u)$. Le nœud u d'un lien $(u, v) \in E$ est l'émetteur du lien et le nœud v est le récepteur du lien. Dans notre graphe, il existe un lien entre tous les nœuds du réseau, ainsi qu'entre un nœud et lui-même (voir figure 18) :

$$\forall u, v \in V, (u, v) \in E \qquad \text{Équation 23}$$

La fonction f associe à chaque lien $(u, v) \in E$ lorsque $u \neq v$, la puissance notée P_{uv} à laquelle le récepteur du lien, le nœud v, perçoit le signal émis par u. La fonction f associe à chaque lien $(u, u) \in E$ entre un nœud u et lui même, le bruit thermique au niveau du nœud u, noté P_{uu} , ainsi $f : E \rightarrow \mathbb{R}$ est définie telle que :

$$f(u,v) = P_{uv}$$ Équation 24

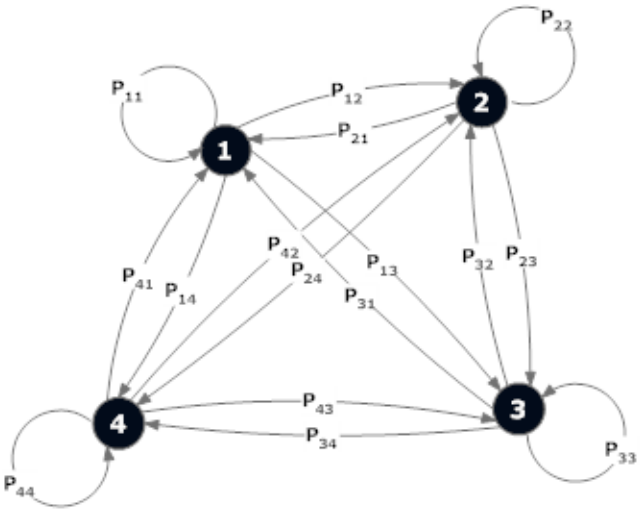


Figure 18 : Graphe complet étiqueté et orienté $G(V,E,f)$

Les valeurs des puissances peuvent être obtenues expérimentalement, préalablement au déploiement du réseau ou via un modèle d'affaiblissement de propagation comme le modèle d'affaiblissement logarithmique où la puissance du signal reçue diminue proportionnellement à la distance entre le récepteur et l'émetteur (Rappaport 2001).

Certains nœuds dans le réseau jouent le rôle de portail d'accès à Internet et forment un sous ensemble V^* de V. L'ensemble des flux admis sur le réseau sont en destination d'Internet, ainsi tous les chemins des flux sur le réseau mesh terminent par un nœud appartenant à V^*. La matrice de puissance P de taille $|V| * |V|$ peut être extraite du graphe $G(V,E,f)$. Chaque élément p_{ij} de la matrice P équivaut, si $i \neq j$, à la puissance à laquelle le nœud récepteur j reçoit le signal du nœud émetteur i, et si $i = j$, p_{ij} équivaut au bruit thermique au niveau du nœud i. La matrice de puissance P du graphe de la figure 18 est :

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{34} & P_{44} \end{pmatrix}$$ Équation 25

La matrice de taille |E|*|V|, notée L^e est la matrice des nœuds émetteurs, elle permet de retrouver le nœud émetteur de chaque lien de l'ensemble E. Chaque élément l^e_{ij} de la matrice L^e équivaut à 1 si le nœud j est le nœud émetteur du lien $e_i \in E$ et 0 sinon. La $i^{ème}$ ligne de la matrice L^e est notée $l^e_{i,}$. La matrice de taille |E|*|V|, notée L^r est la matrice des nœuds récepteurs, elle permet de

retrouver le nœud émetteur de chaque lien de l'ensemble E. Chaque élément l_{ij}^{r} de la matrice L^{r} équivaut à 1 si le nœud j est le nœud récepteur du lien $e_i \in E$ et 0 sinon. La $i^{ème}$ ligne de la matrice L^{e} est notée $l_{i,}^{r}$. La puissance de réception $P_{u,z}$ au niveau du nœud récepteur z d'un lien $e_j = (w, z)$ d'un signal émis par le nœud émetteur u d'un lien $e_i = (u, v)$ peut être obtenue via le matrices P, L^{r} et L^{e} :

$$P_{u,z} = l_{i,}^{e} * P * t(l_{j,}^{r}) \qquad \text{Équation 26}$$

avec $t(l_{j,}^{r})$ la transposée de la matrice linéaire $l_{i,}^{r}$. Soit un lien $e_i = (u, v)$ **,** les matrices P, L^{r} et L^{e} permettent également d'obtenir le bruit thermique $P_{u,u}$ au niveau d'un nœud u **:**

$$P_{u,u} = l_{i,}^{e} * P * t(l_{i,}^{e}) \qquad \text{Équation 27}$$

Les équations 26 et 27 permettent respectivement d'obtenir la puissance à laquelle un nœud reçoit le signal d'un autre nœud et le bruit thermique au niveau d'un nœud. Par la suite, ces formules seront utilisées lors du calcul du signal sur interférence plus bruit d'un nœud, calcul nécessaire pour déterminer si un nœud reçoit avec succès un paquet ou non selon le modèle d'interférence additif.

4.3.1. Découpage du temps et modélisation de l'interférence

Dans notre modèle, tous les nœuds du réseau sont synchronisés. Cette synchronisation peut être, par exemple, réalisée par la fonction de synchronisation dans le temps TSF (Timing Synchronization Function) proposée par la norme IEEE 802.11 (IEEE 1997). Le temps est divisé en fenêtres, appelées fenêtres de planification dont la durée est notée T_f. Chaque fenêtre est divisée en N slots de durée équivalente dont les N_c premierssont réservés pour l'envoi de paquets de contrôle, tels que des paquets dédiés au routage ou au CA. L'accès au canal durant ces N_c premiers slots est à compétition. Les N_p slots suivants de la fenêtre de planification sont dédiés à la planification des liens, c.à.d. que l'un de ces slots pourra être utilisé par un nœud pour émettre un flux uniquement si le slot a été préalablement réservé pour l'émission, par ce nœud, de ce flux. L'accès au canal durant les N_p derniers slots de la fenêtre de planification est donc temporel. La figure 19 présente un exemple de division du temps.

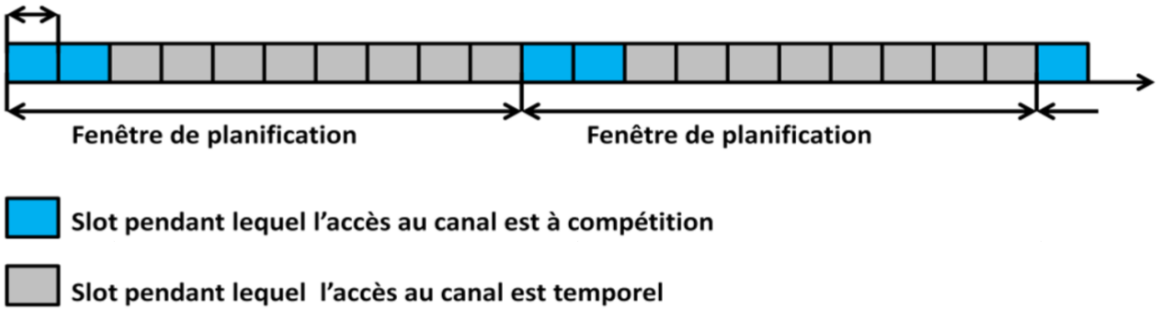


Figure 19 : Division du temps en fenêtres de planification composées de 10 slots $(N = 10)$ dont 2 sont réservés à l'accès à compétition au canal $(N_c = 2)$ et 8 à l'accès temporel au canal $(N_p = 8)$

Le premier slot d'une fenêtre de planification porte le numéro 0, le second slot le numéro 1, le troisième slot le numéro 2, etc. On suppose que tous les nœuds du réseau ont le même débit et que tous les paquets de données ont la même taille. Ainsi, la durée d'émission d'un paquet de données est la même pour tous les nœuds et est notée T_{paq}. Afin de garantir la meilleure granularité possible de la planification d'un lien, la durée d'un slot, notée T_{slot}, correspond au temps d'émission d'un paquet de donnée, ainsi $T_{paq} = T_{slot}$. Lors de l'émission d'un paquet de données, l'accès au canal est temporel et non à compétition, un nœud n'utilise donc ni paquets RTS/CTS ni algorithme de *backoff*. Le temps d'émission d'un paquet par un nœud équivaut alors à :

$$T_{paq} = T_{slot} = T_{DIFS} + T_{plcp} + \frac{L}{C} + T_{SIFS} + T_{plcp} + T_{ack} \qquad \text{Équation 28}$$

Chaque paramètre de l'équation 28 est expliqué dans le tableau suivant.

Tableau 5 : Description des paramètres nécessaires au calcul du temps d'émission d'un paquet de données dans un réseau mesh à accès temporel

Paramètre	Description
T_{DIFS}	Durée de l'espace inter-trame DIFS (définie dans le standard IEEE 802.11 (IEEE 1997))
T_{SIFS}	Durée de l'espace inter-trame SIFS (définie dans le standard IEEE 802.11 (IEEE 1997))
T_{plcp}	Durée de transmission de l'entête PLCP d'un paquet
T_{ack}	Durée de transmission de l'acquittement (ack) du paquet de données
L	Taille d'un paquet de données entête comprise
C	Capacité du canal

L'entête PLCP est souvent oubliée dans les articles lors du calcul de la durée de transmission d'un paquet, comme dans (Yang et Kravets 2005). Cet entête est généralement envoyé à 1Mbit/s (Atelin 2008). Elle est ajoutée au niveau physique de la pile TCP/IP et permet de synchroniser le récepteur et l'émetteur et de préciser le débit de la trame MAC.

Dans notre modèle, on considère que l'envoi de données sur un lien est réussi si le lien ne subit pas de conflits primaires et si le modèle d'interférence additif est respecté. Il y a conflit primaire sur un lien si l'un des nœuds du lien (Djukic et Valaee 2009) :

- reçoit des données en même temps de plusieurs nœuds,
- reçoit des données et transmet simultanément,
- émet des données différentes sur plusieurs récepteurs en même temps.

La figure suivante modélise les trois situations précédentes. Pour chacune de ces trois situations le lien *(A,B)* est en conflit primaire.

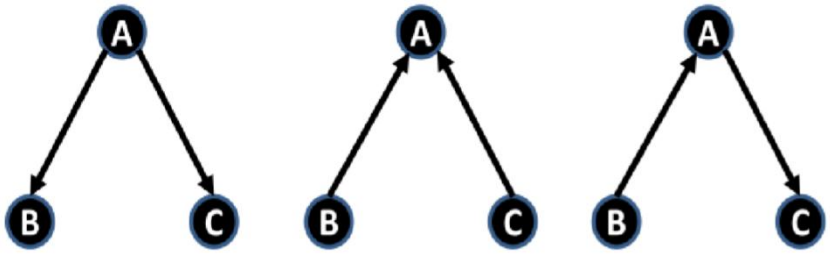


Figure 20: Trois situations où le lien (A,B) est en conflit primaire

Dans notre réseau, si un ensemble de nœuds Γ envoie simultanément des données, j reçoit avec succès le paquet que lui envoie i ($i \in \Gamma$) s'il n'y a pas de conflit primaire au niveau du lien et si le modèle d'interférence additif est respecté, c.à.d. si :

$$\frac{P_{ij}}{P_{jj} + \sum_{u \in \Gamma - \{i\}} P_{uj}} \geq \beta \qquad \text{Équation 29}$$

avec β le seuil SINR à partir duquel la transmission échoue et P_{jj}, pour rappel, le bruit thermique au niveau du nœud j.

Dans notre modèle, chaque nœud recevant avec succès un paquet de données, renvoie un acquittement. Ainsi, on estime qu'une transmission sur un lien (i,j) est réussie si, j reçoit avec succès le paquet de données envoyé par i et i l'acquittement envoyé par j. Tous les nœuds débutent la transmission d'un paquet de données au début d'un slot. Comme la durée de transmission d'un paquet de donnée est équivalente pour tous les nœuds, l'ensemble des nœuds émettant sur un même slot, commencent et finissent de transmettre un paquet de données simultanément. Ainsi, un paquet de données ne peut pas interférer avec un acquittement.

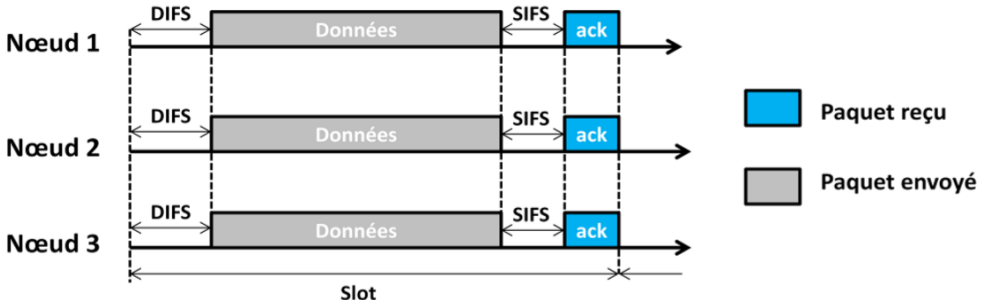


Figure 21 : Trois nœuds envoient un paquet de données sur un même slot

La figure 21 présente trois nœuds qui émettent un paquet de données sur un même slot. Ces trois nœuds commencent et finissent la transmission du paquet de données simultanément, il en est de même pour la transmission de l'acquittement. Ainsi, aucun paquet de données ne peut interférer avec un acquittement, par contre, les paquets de données peuvent interférer entre eux et les acquittements aussi. Soit Υ, un ensemble de liens (u,v) émettant simultanément, il n'existe aucune interférence sur l'ensemble Υ des liens (u,v) qui émettent un paquet de données sur un même slot, si aucun paquet de données n'interfère avec un autre paquet de données (voir équation 30),si aucun ack n'interfère avec un autre ack (voir équation 31) et s'il n'y a aucun conflit primaire (voir équation 32) .

$$\forall(u,v) \in \Upsilon, \frac{P_{uv}}{P_{vv} + \sum_{(i,j)\in\Upsilon-\{(u,v)\}} P_{iv}} \geq \beta$$ Équation 30

$$\forall(u,v) \in \Upsilon, \frac{P_{vu}}{P_{uu} + \sum_{(i,j)\in\Upsilon-\{(u,v)\}} P_{ju}} \geq \beta$$ Équation 31

$$\forall(u,v),(w,z) \in \Upsilon \text{ tel que } (w,z) \neq (u,v), u \neq z, v \neq z, v \neq w, u \neq w$$ Équation 32

Si les trois formules ci-dessus sont vérifiées alors la transmission sur chaque lien de l'ensemble Υ est un succès. A notre connaissance, notre modèle de planification de liens est le premier à considérer que les paquets de données peuvent uniquement interférer avec un ou des paquets de données et qu'un paquet d'acquittement peut uniquement interférer avec un ou des acquittements. Les modèles existants, généralement, ne considèrent pas l'acquittement (Gore, Karandikar et Jagabathula 2007) (Wan, Xu et Frieder 2010) ou, s'ils le font (Brar, Blough et Santi 2006), ils considèrent alors que les paquets d'acquittement peuvent également interférer avec ceux de données, diminuant alors les possibilités de réutilisation spatiale et sous-estimant la bande passante des nœuds. En utilisant l'équation 26 qui permet de calculer la puissance à laquelle un nœud reçoit un signal et l'équation 27 qui permet de calculer le bruit thermique au niveau d'un nœud, les équations 30, 31 et 32 peuvent se réécrire ainsi :

$$\forall e_i \in \Upsilon, \frac{l^e_{i,} * P * t(l^r_{i,})}{l^r_{i,} * P * t(l^r_{i,}) + \sum_{e_j\in\Upsilon-\{e_i\}} l^e_{j,} * P * t(l^r_{i,})} \geq \beta$$ Équation 33

$$\forall e_i \in \Upsilon, \frac{l^r_{i,} * P * t(l^e_{i,})}{l^e_{i,} * P * t(l^e_{i,}) + \sum_{e_j\in\Upsilon-\{e_i\}} l^r_{j,} * P * t(l^e_{i,})} \geq \beta$$ Équation 34

$$\forall e_i, e_j \in \Upsilon \; tel\; que\; e_i \neq e_j,\; l^e_{i,} \neq l^e_{j,}\; l^r_{i,} \neq\; l^r_{j,}\; l^r_{i,} \neq\; l^e_{j,}\; l^e_{i,} \neq l^r_{j,}$$ Équation 35

En utilisant le modèle d'interférence additif, le réseau peut être modélisé par un graphe de communication directionnel $G'(V,E')$ où V représente l'ensemble des nœuds du réseau et E' l'ensemble des liens directionnels. Un lien $(u,v) \in E'$ si le nœud $u \in V$ peut recevoir avec succès les paquets envoyés par v et si le noeud $v \in V$ peut recevoir les paquets envoyés par u, c.à.d. si les deux inéquations suivantes sont vérifiées:

$$\forall (u,v) \in E' \; et \; u \neq v \; , \frac{P_{uv}}{P_{vv}} \geq \beta \; et \; \frac{P_{vu}}{P_{uu}} \geq \beta \qquad \text{Équation 36}$$

D'après l'équation 36, tous les liens de l'ensemble E'sont bidirectionnels.

4.3.2. Les flux admis dans le réseau

L'ensemble des flux admis sur le réseau est représenté par l'ensemble F. Chaque flux $f \in F$ est associé à un quadruplet (p_f, d_f, N_f, s_f) où p_f représente le chemindu flux, d_f le délai du flux, N_f le nombre minimum de slots que doit réserver chaque lien du chemin pour garantir la BP minimum requise par le flux. Le chemin du fluxest composé d'un n-uplets de liens, un n-uplet est un ensemble de n éléments ordonnés, ainsi $p_f = (e_0, e_1 \ldots, e_n)$.

Définition 1: Une route $p_f = (e_0, e_1 \ldots, e_n)$ est valide si les données du nœud émetteur du lien$e_0$peuvent traverser avec succès chaque lien de la route dans l'ordre du n-uplet jusqu'à un portail d'accès qui les achemine vers un réseau plus large.

Ainsi, une route p_f est valide si les trois contraintes suivantes sont respectées :

$$\forall e_i \in p_f - \{e_n\} \; et \; e_i = (u_i, v_i), v_i = u_{i+1} \qquad \text{Équation 37}$$

$$\forall e_i \in p_f, e_i \in E' \qquad \text{Équation 38}$$

$$e_n = (u_n, v_n), v_n \in V^* \qquad \text{Équation 39}$$

L'équation 37 indique que le nœud récepteur d'un lien est le nœud émetteur du lien suivant. L'équation 38 précise que chaque lien de la route appartient à l'ensemble E', c.à.d. que les données peuvent être envoyées avec succès sur chaque lien de la route. L'Équation 39 oblige le nœud destination de la route à être un nœud portail d'accès qui envoie les données sur Internet.

4.3.3. La bande passante et la planification d'un flux

Pour chaque flux admis dans le réseau, l'ensemble des liens de sa route réservent un même nombre de slots afin de garantir sa BP minimum. Si un flux f requiert une bande passante minimum, notée B_f^{min}, alors le nombre minimum de slots, noté N_f, que chaque lien sur la route du flux doit réserver par fenêtre de planification pour garantir cette BP est de :

$$N_f = \left\lceil \frac{B_f^{min} * T_f}{C * T_{slot}} \right\rceil \qquad \text{Équation40}$$

avec C la capacité du canal, $\lceil x \rceil$ la fonction plafond qui associe à une valeur x le plus petit entier supérieur ou égale à x et T_f pour rappel, la durée d'une fenêtre de planification.

Un lien e_i possède, pour chaque flux f qui le traverse, une planification notée ψ^{f,e_i}. Cette planification est une liste ordonnée croissante de N_f éléments, ainsi $\psi^{f,e_i}=(\psi_1^{f,e_i}, \psi_2^{f,e_i} \dots \psi_{N_f}^{f,e_i})$. Chaque élément de la liste ψ_1^{f,e_i} représente un numéro de slot où le lien e_i peut émettre un paquet du flux f si le nœud émetteur du lien e_i en possède au moins un en attente. La planification d'un lien pour un flux f est composée de N_f éléments afin de garantir la bande passante du flux tout en minimisant le nombre de slots réservés par lien pour un flux. Un lien peut émettre un paquet du flux f sur le numéro de slot j d'une fenêtre de planification si $j \in \psi^{f,e_i}$. La figure 22 présente la fenêtre de planification d'un lien e_i, la planification de e_i pour le flux f est $\psi^{f,e_i}=(3, 5, 9)$. Pour rappel, le premier slot d'une fenêtre de planification porte le numéro 0.

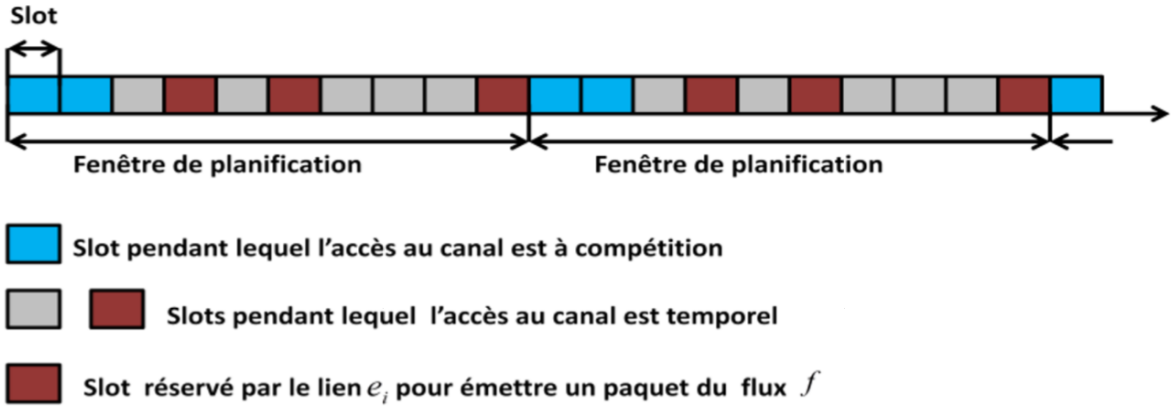


Figure 22 : Le nœud e_i a réservé trois slots pour le flux f par fenêtre de planification

La planification d'un flux peut également être représentée par une matrice S^f de taille |E|*N_p avec N_p le nombre de slots réservables par fenêtre de planification (en gris et rouge sur la figure 22), et E l'ensemble des liens du réseau. Chaque élément s_{ij}^f de la matrice S^f équivaut à 1 si le slot numéro $j-1+N_c$ est réservé par le lien $e_i \in E$ pour émettre le flux f ou à 0 dans le cas contraire. Par exemple, sur la figure 22, les slots numéro 3, 5 et 9 sont réservés pour l'émission du flux f par le lien e_i, la taille de la fenêtre de planification équivaut à *N=10,* le nombre de slots réservables planifiables est de 8 ($N_p = 8$) et le nombre de slots de non réservables et de contrôle est de 2 ($N_c = 2$). La $i^{ème}$ ligne de la matrice S^f, notée $s_{i,}^f$, représente la planification du lien e_i pour le flux f et équivaut à $s_{i,}^f = 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1$. Lorsqu'un flux f est admis sur le réseau, la matrice planification du flux S^f doit être valide. Une planification de flux valide est définie comme suit.

Définition 2 *: La planification S^f* d'un flux *est valide si, uniquement les liens appartenant au chemin du flux f ont des slots réservés pour émettre f et que chacun de ces liens possèdent la quantité minimum de slots par fenêtre de planification pour satisfaire la bande passante requise par le flux.*

D'après la définition 2, une matrice de planification de flux S^f est valide si les deux contraintes suivantes sont vérifiées :

$$\forall i\ tel\ que\ e_i \in p_f, \sum_{j=1}^{j=N_p} s_{ij}^f = N_f \qquad \text{Équation 41}$$

$$\forall i\ tel\ que\ e_i \notin p_f, \sum_{j=1}^{j=N_p} s_{ij}^f = 0 \qquad \text{Équation 42}$$

L'équation 41 vérifie que l'ensemble des liens du chemin du flux possèdent le nombre minimum de slots par fenêtre de planification pour satisfaire la BP requise par le flux. L'équation 42 vérifie que les liens n'appartenant pas au chemin du flux n'ont aucun slot réservé pour émettre le flux. La matrice de planification du réseau est notée S, elle est de taille $|E| * N_p$ et est la somme de l'ensemble des matrices de planification de tous les flux admis du réseau, ainsi :

$$S = \sum_{\forall f \in F} S^f \qquad \text{Équation 43}$$

Chaque élément s_{ij} de la matrice S équivaut à 1 si le slot numéro $j-1+N_c$ est réservé pour le lien $e_i \in E$ et 0 sinon. La matrice S de planification du réseau doit être à tout instant valide. Une planification du réseau valide est définie comme suit :

Définition 3 *: Une matrice S est valide si elle est la somme de planification de flux valides, s'il n'existe aucun conflit primaire et si elle respecte le modèle d'interférence additif.*

Chaque élément s_{ij} de la matrice S représente la planification du lien e_i au slot $j+N_c-1$. Ainsi la matrice de planification S vérifie le modèle d'interférence additif et ne possède aucun conflit primaire si les trois inéquations suivantes sont respectées :

$$\forall j \in [1, N_p] \; et \; \forall i, e_i \in |E|,$$

$$\frac{s_{ij} * l^e_{i,} * P * t(l^r_{i,}) - \Lambda(1 - s_{ij})}{s_{ij} * l^r_{i,} * P * t(l^r_{i,}) + \sum_{\forall y, e_y \in E - \{e_i\}} s_{yj} * l^e_{j,} * P * t(l^r_{i,})} \geq \beta$$

Équation 44

$$\forall j \in [1, N_p] \text{ et } \forall i, e_i \in |E|,$$

$$\frac{s_{ij} * l^r_{i,} * \mathrm{P} * t(l^e_{i,}) - \Lambda(1 - s_{ij})}{s_{ij} * l^e_{j,} * \mathrm{P} * t(l^e_{i,}) + \sum_{\forall y, e_y \in E - \{e_i\}} s_{yj} * l^r_{j,} * \mathrm{P} * t(l^e_{i,})} \geq \beta$$

Équation 45

$$\forall j \in [1, N_p] \; et \; \forall v \in [1, |V|], \quad \sum_{i=1}^{i=|E|} s_{ij} * l^e_{iv} + s_{ij} * l^r_{iv} \leq 1$$

Équation 46

avec Λ un grand entier positif. Cet entier est introduit afin que, si un lien e_i n'a pas réservé le slot numéro $j-1+N_c$ et que donc $s_{ij} = 0$, les deux premières contraintes soient toujours vérifiées. L'équation 44vérifie que le modèle d'interférence additif lors de l'envoi d'un paquet de données est respecté pour chaque lien du réseau et chaque slot planifiable de la fenêtre de planification. Lorsqu'un lien e_i n'a pas réservé le slot numéro $j-1+N_c$, $s_{ij} = 0$, il n'y donc alors aucun risque d'interférence additif au niveau du lien e_i, la présence du grand entier positif Λ permet dans ce cas de vérifier l'équation et ainsi le fait qu'il n'y ait aucun risque d'interférence au niveau du lien e_i. L'équation 45 vérifie si le modèle d'interférence additif, lors de l'envoi d'un acquittement, est

respecté pour chaque lien du réseau et chaque slot planifiable de la fenêtre de planification. L'entier positif Λ joue dans cette équation le même rôle que dans l'équation précédente. L'équation 46 vérifie qu'il n'y ait aucun conflit primaire, c.à.d. que tout nœud v soit planifié comme récepteur ou émetteur au maximum une seule et unique fois par slot j planifiable.

Le réseau doit toujours être valide, la validité d'un réseau mesh est définie comme suit :

Définition 4 : *Un réseau est valide si l'ensemble des flux admis dans le réseau ont un chemin valide (voir définition 3), si l'ensemble des planifications de flux S^f (voir définition2) et la planification S du réseau sont valides (voir définition 4).*

4.3.4. Le délai d'un flux

Un flux f suit un chemin $p_f = (e_0, e_1 \ldots, e_n)$ où le nœud émetteur de chaque lien e_i est noté u_i. Chaque nœud possède une file d'attente premier arrivé premier servi (First In First Out-FIFO) et N_f slots réservés pour chaque flux f qu'il doit transmettre. Un nœud émet un paquet à un slot réservé si, au début de ce dernier, il en a au moins un en file d'attente. Pour rappel, chaque nœud u_i d'un lien e_i, sauf la destination, sur la route d'un flux f, possède un vecteur de planification pour ce flux $\psi^{f,e_i} = (\psi_1^{f,e_i}, \psi_2^{f,e_i} \ldots \psi_{N_f}^{f,e_i})$. Chaque élément représente un numéro de slot pendant lequel le nœud u_i peut émettre un paquet du flux f (s'il possède alors, au moins un paquet en attente). Les éléments d'un vecteur ψ^{f,e_i} sont rangés par ordre croissant, ainsi $\psi_1^{f,e_i} < \psi_2^{f,e_i} \ldots < \psi_{N_f}^{f,e_i}$ avec 1, 2,… N_f l'indice d'un élément.

Le temps écoulé entre le moment où un nœud reçoit intégralement un paquet et le moment où il l'a entièrement retransmis correspond au délai du paquet au niveau du nœud. Un nœud transmet un paquet soit pendant la fenêtre de planification où il reçoit le paquet, soit au cours de la fenêtre suivante. Les figures 23 et 24 illustrent ce phénomène. Par exemple, si un flux passant par les nœuds A, B et C, est planifié sur le slot numéro 3 par A et sur le slot numéro 7 par B, alors le délai d'un paquet du flux au niveau du nœud B est de 5 slots (7-3=4) (voir figure 23). Si, par contre, le nœud A réserve le slot numéro 7 pour émettre un paquet du flux et B le slot numéro 3, sachant que le nombre de slots par fenêtre de planification est de 12, alors le délai d'un paquet du flux au niveau du nœud B est de 7 slots (12-7+3=8) (voir figure 24).

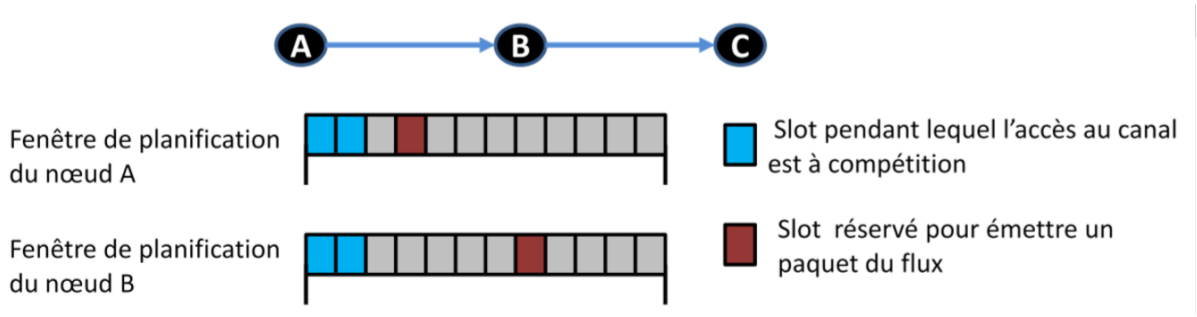


Figure 23 : Le nœud B émet le paquet en provenance du nœud A au cours de la fenêtre de planification où il a reçu le paquet

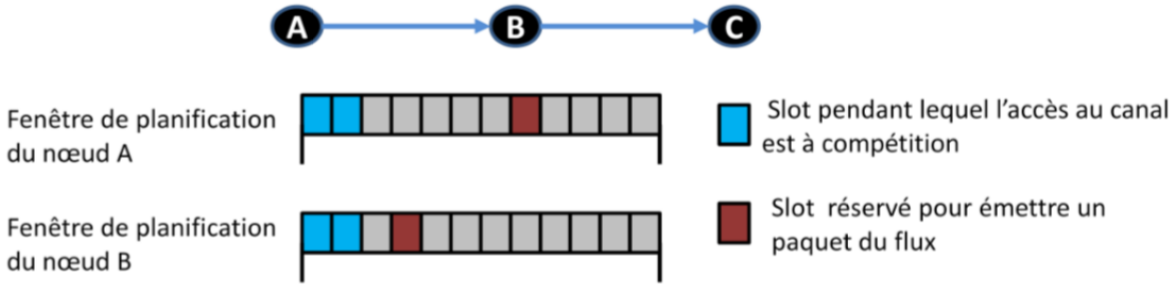


Figure 24 : Le nœud B émet le paquet en provenance de A au cours de la fenêtre de planification suivant celle où il a reçu le paquet.

Ainsi, si un nœud u_{i-1} émet un paquet au slot numéro $\psi_j^{f,e_{i-1}}$ et le nœud suivant u_i le retransmet au nœud ψ_z^{f,e_i} alors le délai du paquet au niveau du nœud u_i équivaut à :

$$d_i(j) = \begin{cases} \psi_z^{f,e_i} - \psi_j^{f,e_{i-1}} + 1 \; si \; \psi_z^{f,e_i} > \psi_j^{f,e_{i-1}} \\ N - \psi_z^{f,e_i} + \psi_j^{f,e_i-1} + 1 \; si \; \psi_z^{f,e_i} < \psi_j^{f,e_{i-1}} \end{cases}$$ Équation 47

Le délai d'un paquet équivaut à la somme des délais du paquet à chaque nœud intermédiaire de la route du paquet. Le délai d'un flux est le délai maximum que l'un des paquets du flux peut obtenir.

Le délai d'un paquet au niveau d'un nœud u_i, transmis à ce dernier au slot numéro $\psi_j^{f,e_{i-1}}$ est d'autant plus important qu'il a, lorsqu'il reçoit le paquet, de nombreux paquets en attente dans sa file d'attente FIFO. En effet, il devra alors attendre que tous les paquets de sa file d'attente FIFO soient envoyés pour émettre le paquet reçu au slot numéro $\psi_j^{f,e_{i-1}}$. Or, la file d'attente FIFO d'un nœud est d'autant plus longue qu'il reçoit un paquet à chaque slot réservé de son nœud précédent.

Ainsi, le délai d'un paquet, reçu par le nœud u_i au slot numéro $\psi_j^{f,e_{i-1}}$ ne peut pas être supérieur à celui qu'il aurait obtenu si son prédécesseur envoyait un paquet à chaque slot réservé. Ainsi, pour estimer le délai maximum d'un paquet au niveau d'un nœud intermédiaire de la route, on suppose par la suite, que son prédécesseur lui envoie un paquet à chaque slot réservé.

Au cours de la première fenêtre de planification, un nœud u_{i-1} envoie donc N_f paquets au nœud suivant sur la route du flux, u_i. Ce dernier fait suivre uniquement une partie de ces N_f paquets car, à certains de ses N_f slots réservés, il ne possède aucun paquet en attente, ces slots sont appelés des *slots perdus* (voir figure 25). A la fin de la première fenêtre de planification, un nœud u_i, intermédiaire sur la route d'un flux f, a donc autant de paquets dans sa file d'attente FIFO que de *slots perdus*. Sur la figure 25, le nœud B ne possède au début de la première fenêtre de planification aucun paquet en attente. Au cours de la première fenêtre de planification, le nœud B, à son slot réservé numéro 3, ne possède aucun paquet en attente à émettre, ce dernier est un *slot perdu* de B.

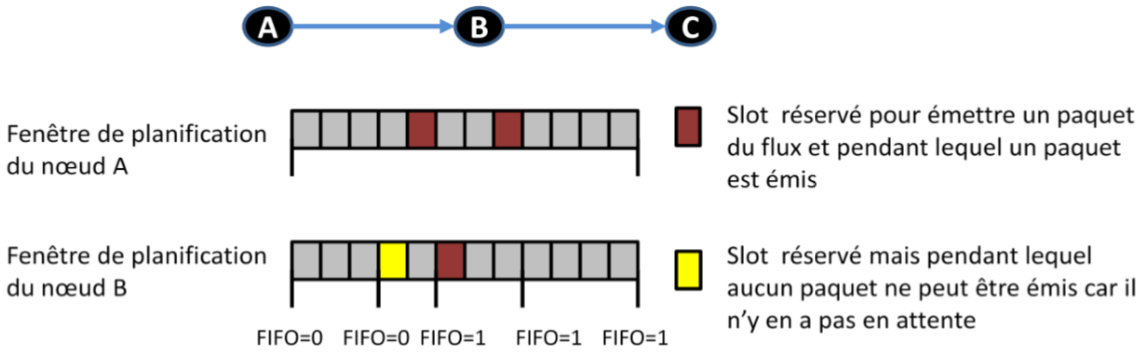


Figure 25 : Illustration du phénomène de *slots perdus*, lorsqu'un nœud ne peut émettre de paquet lors d'un slot réservé car il n'en possède aucun en attente

Lors de la seconde fenêtre de planification, u_i reçoit de nouveau N_f paquets de son prédécesseur sur la route du flux. Au cours de cette dernière, u_i peut émettre à chaque slot réservé, car il possède au début de la fenêtre autant de paquets dans sa file d'attente qu'il a de *slots perdus*. Il stocke également, à la fin de la seconde fenêtre de planification autant de paquets en attente qu'il a de *slots perdus* puisqu'il aura pu réémettre sur les N_f paquets reçus de son prédécesseur, N_f paquets moins son nombre de slots perdus. Ainsi, le nœud u_i est, au début de la troisième fenêtre, dans le même état qu'au début de la deuxième fenêtre, il se retrouvera donc également dans la même situation au début de la quatrième, la cinquième, etc. On peut en conclure que, à partir de la fin de la première fenêtre de planification, le nœud u_i envoie un paquet à chaque slot réservé.

En d'autres termes, une fois qu'un nœud u_i transmet le $N_f^{ème}$ paquet que son prédécesseur lui a envoyé, il fait suivre un paquet à chacun de ses slots planifiés. Ainsi, si le nœud u_i envoie le $N_f^{ème}$

paquet au slot ψ_z^{f,e_i} alors il transmettra, par la suite, chaque paquet reçu au slot numéro $\psi_j^{f,e_{i-1}}$ d'une fenêtre de planification au slot numéro ψ_x^{f,e_i} tel que l'indice x du slot peut être obtenu via la formule suivante :

$$x = \begin{cases} (z+j)\%N_f \ si \ j+z > N_f \\ z+j \ si \ j+z \leq N_f \end{cases} \qquad \text{Équation 48}$$

L'algorithme 1 suivant permet d'établir le numéro d'indice z du numéro de slot réservé de u_i pendant lequel ce dernier envoie le $N_f{}^{ème}$ paquet reçu de son prédécesseur sur la route du flux u_{i-1}. La connaissance de cet indice z permet de calculer, via l'équation 48, l'indice x du slot réservé pendant lequel le nœud u_i émet un paquet qu'il reçoit au slot numéro $\psi_j^{f,e_{i-1}}$.

Entrée: les vecteurs de planification du flux de u_i, ψ^{f,e_i} et de u_{i-1}, $\psi^{f,e_{i-1}}$
Sortie: l'indice z

1: initialiser l' indice $y = 1$ et $z = 1$
2: **Tant que** $z \leq N_f$ **Faire**
3: **Tant que** $\psi_z^{f,e_i} > \psi_y^{f,e_{i-1}}$ *and* $z \leq N_f$ **Faire**
4: $y++$ et $z++$
5: **Fin Tant que**
6: **Si** $z \leq N_f$ **Alors**
7: $z++$
8: **Fin Si**
9: **Si** $z > N_f$ **Alors**
10: $z = N_f + y - 1$
11: **Retourne** z
12: **Fin Si**
13: **Fin Tant que**

Algorithme 1 : Algorithme qui retourne l'indice z du numéro de slot ψ_z^{f,e_i} pendant lequel le nœud u_i fait suivre le $N_f{}^{ème}$ paquet reçu de u_{i-1}

Cet algorithme prend en entrée les vecteurs de planification du flux du lien e_i et e_{i-1} notés respectivement ψ^{f,e_i} et $\psi^{f,e_{i-1}}$. L'indice z est initialisé à 1 et l'indice y du numéro de slot réservé par le nœud u_{i-1} à 1 également (ligne 1). L'algorithme effectue ensuite une boucle ; tant que l'indice z n'est pas supérieur à N_f, on vérifie si ψ_z^{f,e_i} est située après $\psi_y^{f,e_{i-1}}$ (ligne 3) et donc si le nœud u_i pourrait envoyer le paquet, reçu au slot numéro $\psi_y^{f,e_{i-1}}$, au slot numéro ψ_z^{f,e_i} de la même fenêtre de planification. Si c'est le cas, on augmente les indices z et y de 1 (ligne 4). Si la vérification échoue, c.à.d si $\psi_z^{f,e_i} < \psi_y^{f,e_{i-1}}$, alors z est incrémenté de 1. La boucle continue tant que $z \leq N_f$, puisque l'indice maximum d'un slot réservé est N_f. L'algorithme s'arrête lorsque z équivaut à $N_f + 1$. En effet à ce stade, on sait que y-1 éléments du vecteur ψ^{f,e_i} sont situés après un élément à chaque

fois différent du vecteur ψ^{f,e_i-1}, on en déduit que $y-1$ paquets reçus par u_i au cours d'une fenêtre de planification peuvent être réémis au cours de cette même fenêtre. Ainsi, le $N_f{}^{ème}$ paquet que u_i reçoit en provenance de u_{i-1} peut être réémis au numéro de slot réservé dont l'indice z équivaut à N_f- $y-1$. La valeur de l'indice$z$ est finalement retournée par l'algorithme.

A partir de la seconde fenêtre de planification, le nœud u_i retransmettra toujours un paquet reçu de son prédécesseur u_{i-1} au $j^{ème}$ slot réservé de ce dernier (c.à.d. à $\psi_j^{f,e_{i-1}}$) à son slot réservé d'indice x. Nous introduisons pour chaque nœud u_i, la fonction $\rho^i: N \to N$ qui renvoie, pour l'indice$j$ du numéro de slot $\psi_j^{f,e_{i-1}}$ planifié par u_{i-1},l'indice x du numéro de slot ψ_x^{f,e_i} pendant lequel u_i retransmet le paquet qu'il a reçu au cours du numéro de slot $\psi_j^{f,e_{i-1}}$ d'une fenêtre de planification.

Un nœud u_i envoie donc au slot numéro $\psi_{\rho^i(j)}^{f,e_i}$ le paquet qu'il a reçu du nœud u_{i-1} au slot numéro $\psi_j^{f,e_{i-1}}$. Le calcul du délai d'un paquet présenté dans l'équation 47 peut ainsi se réécrire comme suit :

$$d_i(j) = \begin{cases} (\psi_{\rho^i(j)}^{f,e_i} - \psi_j^{f,e_i-1}) * \mathrm{T_{slot}} & \text{si } \psi_{\rho^i(j)}^{f,e_i-1} > \psi_j^{f,e_i-1} \\ N + \psi_{\rho^i(j)}^{f,e_i} - \psi_j^{f,e_i-1}) * \mathrm{T_{slot}} & \text{si } \psi_{\rho^i(j)}^{f,e_i} \leq \psi_j^{f,e_i-1} \end{cases} \qquad \text{Équation 49}$$

Lorsqu'un nœud ne peut pas retransmettre un paquet au cours de la fenêtre de planification où il l'a reçu, le délai du paquet au niveau du nœud est calculé selon la seconde ligne de l'équation 49 sinon il est calculé selon la première ligne. Pour calculer le délai d'un paquet, on considère le pire cas, celui où tous les nœuds du chemin émettent un paquet à tous les slots qu'ils ont de planifiables. On note $d: N \to N$, la fonction qui associe au numéro de slot sur lequel le paquet est envoyé par le nœud u_0 le délai du paquet. Ainsi, le délai d'un paquet émis par le nœud u_0 à ψ_j^{f,e_0} équivaut à :

$$d(j) = d_1(j) + d_2(\rho^1(j)) + d_3(\rho^2(\rho^1(j))) \dots d_n\left(\rho^{n-1}\left(\rho^{n-2}(\dots\rho^1(j)\dots)\right)\right) \qquad \text{Équation 50}$$

avec $d_1(j)$ le délai du paquet au niveau du nœud u_1, $d_2(\rho^1(j))$ le délai du paquet au niveau du nœud 2, $d_n\left(\rho^{n-1}\left(\rho^{n-2}(\dots\rho^1(j)\dots)\right)\right)$ le délai du paquet au niveau du nœud précédent le nœud destination du flux, etc.

Comme le nœud source u_0 d'un flux peut émettre un paquet sur N_f numéros de slot différents, le délai maximum d'un paquet peut uniquement prendre N_f valeurs possible. Ainsi, le délai d_f d'un flux

f est le plus grand des délais parmi les N_f délais maximums que peut obtenir un paquet du flux, ainsi :

$$d_f = \max_{j \in [1, N_f]} d(j) \qquad \text{Équation 51}$$

Nous avons, dans cette section, présenté une méthode qui permet, en connaissant la planification du flux au niveau de chaque lien de sa route, calculer son délai. Afin d'établir le délai maximum du flux, on suppose que chaque nœud envoie au nœud suivant sur la route de flux un paquet à chaque slot qu'il a réservé. Le calcul du délai s'effectue en plusieurs étapes :

1. L'algorithme 1 est lancé pour chaque nœud intermédiaire sur la route du flux. Ce dernier établit pour chaque nœud l'indice de son slot réservé sur lequel il envoie le $N_f^{ème}$ paquet qu'il reçoit du nœud précédent sur le chemin du flux.
2. Pour chaque nœud u_{i-1} sur la route du flux, sauf la destination et chacun de ses N_f slots réservés, on effectue l'équation 48. Cette formule permet, pour un indice j de numéro de slot planifié d'un nœud u_{i-1}, d'obtenir l'indice x du numéro de slot planifié du nœud suivant u_i pendant lequel il fera suivre le paquet que le nœud précédent u_{i-1} lui envoie à son numéro de slot planifié dont l'indice est j. Ainsi, cette deuxième étape permet d'établir pour chaque nœud u_i, le résultat de la fonction $\rho^i : N \to N$.
3. Pour chaque nœud u_{i-1} et chacun de ses N_f slots réservés, on effectue l'équation 49. Cette formule permet, d'obtenir le délai maximum $d_i(j)$ d'un paquet au niveau d'un nœud u_i lorsque ce dernier le reçoit du nœud u_{i-1} au numéro de slot ψ_i^{f,e_0} .
4. L'équation 50 est ensuite réalisée pour chaque N_f slots réservés du nœud u_0. Cette formule permet d'établir le délai maximum d'un paquet lorsqu'il est envoyé par le nœud u_0 à son slot planifiée dont l'indice est j, le délai d'un paquet étant la somme des délais à chaque nœud intermédiaire de sa route.
5. L'équation 51 permet finalement d'obtenir le délai du flux. Cette dernière calcule le délai maximum d'un flux comme étant l'un des plus importants délais parmi les N_f délais possibles maximums d'un paquet du flux.

4.3.5. Formulation du problème d'admission d'un flux

Lorsqu'un nœud souhaite envoyer un nouveau flux f sur le réseau, il effectue une demande de d'admission pour ce flux. Dans cette demande, le nœud source précise le délai maximum requis pour le flux, noté d_f^{max} et le nombre minimum de slots nécessaires sur chaque nœud pour satisfaire sa bande passante noté N_f . Le CA est effectué par le nœud destination du flux ; ce nœud est un portail d'accès qui envoie les données sur un réseau plus large et appartient à l'ensemble V^*. Le destinataire doit décider s'il peut admettre ou non le flux le long de la route $p_f = (e_0, e_1, \dots e_n)$ empruntée par la demande de réservation. La route p_f découverte par la demande de réservation est valide dans le sens de la définition 1 car :

- chaque lien de la route p_f a pour nœud récepteur le nœud émetteur du prochain lien (si ce dernier n'est pas le nœud destination) de la route p_f et vérifie donc l'équation 37.
- le dernier nœud de la route est un nœud portail d'accès Internet, ainsi, la route p_f vérifie l'équation 39.
- chaque nœud recevant une demande de réservation vérifie si le lien (i, j) que vient d'emprunter le message appartient à E', et donc si des paquets de données peuvent être envoyés avec succès du nœud i à j et du nœud j à i. Si $(i, j) \in E'$ alors le nœud accepte le paquet de demande d'admission, et s'il est un nœud intermédiaire sur la route du flux le fait suivre. Pour rappel, un lien appartenant à E' est un lien de l'ensemble E qui vérifie l'équation 36. Si $(i, j) \notin E'$ alors le nœud détruit le paquet. Ainsi toute route p_f vérifie l'équation 38.

D'après la définition 1, la route p_f est valide car elle vérifie les équations 37, 38 et 39. Si la demande de délai du flux est négligée ou si le flux ne nécessite pas de délai maximum (le délai maximum du flux étant alors fixé à $+\infty$), alors le problème d'admission d'un flux le long d'un chemin p_f dans un réseau valide (voir définition 4) peut s'écrire sous forme d'un programme linéaire en variables binaires (PLVB). Ce programme a pour but de trouver la nouvelle planification S^f du flux tel que le nombre de slots planifiés pour ce flux soient minimums, que la contrainte en termes de bande passante du flux soit respectée et que de cette nouvelle planification n'engendre aucun conflit primaire et aucune interférence sur le réseau selon le modèle d'interférence additif. Ce programme linéaire à variables binaires se présente comme suit :

Minimiser

$$\sum_{i=1}^{i=N_p} \sum_{j=1}^{j=|E|} s_{ij}^f, s_{ij}^f \in \{0,1\}$$ Équation 52

sous les contraintes :

1. $\forall i\ , e_i \in p_f, \sum_{j=1}^{j=N_p} s_{ij}^f = N_f$ Équation 53

2. $\forall i\ tel\ que\ e_i \notin p_f, \sum_{j=1}^{j=N_p} s_{ij}^f = 0$ Équation 54

3. $\forall j \in [1, N_p]\ et\ \forall i, e_i \in E, s_{ij}^f + s_{ij} \leq 1$ Équation 55

4. $\forall j \in [1, N_p]\ et\ \forall i, e_i \in E,$

$$\frac{(s_{ij} + s_{ij}^f) * l_{i,}^e * P * t(l_{i,}^r) - \Lambda\left(1 - (s_{ij} + s_{ij}^f)\right)}{(s_{ij} + s_{ij}^f) * l_{i,}^r * P * t(l_{i,}^r) + \sum_{\forall y, e_y \in E - \{e_i\}} (s_{yj} + s_{yj}^f) * l_{y,}^e * P * t(l_{i,}^r)} \geq \beta$$ Équation 56

5. $\forall j \in [1, N_p]\ et\ \forall i, e_i \in E,$

$$\frac{(s_{ij} + s_{ij}^f) l_{i,}^r * P * t(l_{i,}^e) - \Lambda\left(1 - (s_{ij} + s_{ij}^f)\right)}{(s_{ij} + s_{ij}^f) * l_{i,}^e * P * t(l_{i,}^e) + \sum_{\forall y, e_y \in E\ - \{e_i\}} (s_{yj} + s_{yj}^f) * l_{y,}^r * P * t(l_{i,}^e)} \geq \beta$$ Équation 57

6. $\forall j \in [1, N_p]\ et\ \forall v \in [1, |V|], \sum_{i=1}^{i=|E|} (s_{ij} + s_{ij}^f) * l_{iv}^{\ e} + (s_{ij} + s_{ij}^f) * l_{iv}^{\ r} \leq 1$ Équation 58

Le tableau suivant récapitule l'ensemble des variables utilisées dans le programme linéaire à variables binaires d'admission d'un flux dans un réseau mesh.

La résolution de ce programme linéaire à variables binaires retourne (si une solution existe) l'ensemble des valeurs des $N_p * |E|$ éléments s_{ij}^f de la matrice de planification S^f du flux f tel que la somme des éléments de cette matrice soit minimale (voir équation 52), sachant que, chaque élément peut prendre uniquement une valeur binaire.

Tableau 6 : Tableau de description des paramètres

Paramètre	Description
s_{ij}^f	Elément de la matrice S^f de planification du flux f. Sa valeur détermine si le lien e_i est planifié sur le $j^{ème}$ slot planifiable de la fenêtre de planification pour émettre le flux ($s_{ij}^f = 1$) ou non ($s_{ij}^f = 0$).
N_f	Nombre de slots à planifier par fenêtre de planification pour le flux f afin de respecter sa bande passante.
s_{ij}	Elément de la matrice S de planification du réseau. Sa valeur détermine si le lien e_i est planifié sur le $j^{ème}$ slot planifiable de la fenêtre de planification pour émettre le flux f ($s_{ij} = 1$) ou non ($s_{ij} = 0$).
P	Matrice de puissance dont chaque élément p_{ij} équivaut si $i \neq j$ la puissance à laquelle le nœud j reçoit un signal envoyé par i et si $i = j$, p_{ij} le bruit thermique au niveau du nœud j.
E	Ensemble des liens du réseau, toute paire de nœuds $u, v \in V$ du réseau forme un lien $(u, v) \in E$.
l_{ij}^e	Elément de la matrice L^e des nœuds émetteurs. l_{ij}^e équivaut à 1 si le nœud j est le nœud émetteur du lien $e_i \in E$ et 0 sinon.
l_{ij}^r	Elément de la matrice L^r des nœuds récepteurs. l_{ij}^r équivaut à 1 si le nœud j est le nœud récepteur du lien $e_i \in E$ et 0 sinon.
$l_{i,}^r$	Représente la $i^{ème}$ ligne de la matrice L^r des nœuds récepteurs. Le $j^{ème}$ élément de cette matrice linéaire équivaut à 1 si le lien e_i a comme nœud récepteur le $j^{ème}$ nœud du réseau sinon il équivaut à 0.
$l_{i,}^e$	Représente la $i^{ème}$ ligne de la matrice L^e des nœuds émetteurs. Le $j^{ème}$ élément de cette matrice linéaire équivaut à 1 si le lien e_i a comme nœud émetteur le $j^{ème}$ nœud du réseau sinon il équivaut à 0.
β	Seuil SINR en dessous duquel le nœud ne reçoit pas avec succès les données
Λ	Grand nombre entier positif
$t()$	Cette fonction renvoie la transposée d'une matrice

Une matrice planification S^f du flux f solution du PLVB vérifie un ensemble de six contraintes :

1. la première contrainte (voir équation 53) assure que chaque lien de la route réserve, pour le flux, uniquement le nombre minimum N_f de slots nécessaire au respect de la contrainte en termes de bande passante du flux.
2. la seconde contrainte (voir équation 54) vérifie que chaque lien n'appartenant pas à la route du flux ne planifie aucun slot pour émettre le flux f. D'après la définition 2, une matrice de planification de flux S^f vérifiant les deux premières contraintes est une matrice de planification de flux valide.
3. la troisième contrainte (équation 55) assure que chaque lien e_i du réseau ne peut émettre qu'une seule et unique fois sur un slot planifiable. Ainsi, sur le $j^{ème}$ slot planifiable de la fenêtre de planification un lien peut soit être planifié pour émettre un flux ($s_{ij} + s_{ij}^f$=1) soit aucun flux ($s_{ij} + s_{ij}^f$=0).
4. la quatrième contrainte (équation 56) vérifie que le modèle d'interférence additif est respecté pour l'ensemble des liens sur chacun des slots planifiables du réseau lors de l'envoi des paquets de donnée. Ainsi, pour tout lien $e_i = (u, v)$ qui émet sur un slot j planifiable, ($s_{ij} + s_{ij}^f$ =1), la quatrième contrainte garantit que la puissance à laquelle le nœud récepteur$v$ du lien e_i reçoit le signal du paquet de données envoyé par le nœud émetteur u $((s_{ij} + s_{ij}^f) * l_{i,}^e * \mathrm{P} * t(l_{i,}^r))$ sur la somme de la puissance thermique au niveau du nœud récepteur du lien e_i $((s_{ij} + s_{ij}^f) * l_{i,}^r * \mathrm{P} * t(l_{i,}^r))$ et la somme des interférences que le nœud récepteur v du lien e_i reçoit en provenance des autres nœuds émetteurs des liens qui sont planifiés également sur le $j^{ème}$ slot planifiable ($\sum_{\forall y, e_y \in E - \{e_i\}} (s_{yj} + s_{yj}^f) * l_{y,}^e * \mathrm{P} * t(l_{i,}^r)$), est supérieure au seuil SINR β. Si le $j^{ème}$ slot planifiable n'est pas réservé pour le lien e_i, ($s_{ij} + s_{ij}^f$=0), alors il n'y a aucun risque que le nœud récepteur du lien e_i subisse une interférence, la contrainte doit donc toujours être vérifiée. Pour cela, un grand nombre entier positif Λ a été introduit.
5. La cinquième contrainte (équation 57) vérifie que le modèle d'interférence additif est respecté pour l'ensemble des liens sur chacun des slots planifiables du réseau lors de l'envoi de l'acquittement d'un paquet de données. Ainsi, pour tout lien e_i tel que $e_i = (u, v)$ ayant réservé le $j^{ème}$ slot planifiable ($c.a.d.$ ($s_{ij} + s_{ij}^f$=1)), la quatrième contrainte garantit que la puissance à laquelle le nœud émetteur u du paquet de données du lien e_i reçoit le signal de l'acquittement de son paquet envoyé par v ($c.a.d.$ $(s_{ij} + s_{ij}^f) * l_{i,}^r * \mathrm{P} * t(l_{i,}^e)$) sur la somme de la puissance thermique au niveau du nœud u du lien e_i ($c.a.d.$ $(s_{ij} + s_{ij}^f) * l_{i,}^e * \mathrm{P} * t(l_{i,}^e)$)

et la somme des interférences que le nœud $u$du lien e_i reçoit en provenance des nœuds récepteurs des autres liens qui sont planifiés également sur le $j^{ème}$ slot planifiable (c.à.d.$\sum_{\forall y, e_y \in E-\{e_i\}}(s_{yj} + s_{yj}^f) * l_{y,}^r * \mathrm{P} * t(l_{i,}^e)$) est supérieure au seuil SINR β. Si le $j^{ème}$ slot planifiable n'est pas réservé pour le lien e_i, c'est-à-dire si ($s_{ij} + s_{ij}^f$=0), alors il n'y a aucun risque d'interférence au niveau du lien, la contrainte doit donc toujours être vérifiée. Afin qu'elle soit toujours vérifiée, un grand nombre entier positif Λ est introduit dans l'équation.

6. La sixième contrainte (équation 58) vérifie qu'il n'y ait aucun conflit primaire pour l'ensemble des liens sur l'ensemble des slots planifiables. Ainsi, un nœud ne peut pas émettre ni recevoir sur plusieurs liens à la fois sur un même slot.

Lorsqu'un nouveau flux est admis sur le réseau, une nouvelle matrice S_{new} de planification du réseau est calculée. Cette matrice est la somme de l'ancienne matrice de planification du réseau S_{old} et de la matrice S^f de planification du nouveau flux. S_{new} est la somme de planifications de flux valides puisque la planification S^f du nouveau flux est valide et de l'ancienne matrice S_{old} de planification du réseau valide qui est donc, d'après la définition 3 la somme de planifications de flux valides. D'après les contraintes 4, 5 et 6 du PLVB, la nouvelle matrice S_{new} respecte le modèle d'interférence additif et évite tout conflit primaire. La nouvelle matrice S_{new} est donc, d'après la définition 3, est valide.

Comme la route d'un flux, la matrice de planification d'un flux et la matrice de planification du réseau sont toujours valides, d'après la définition 4, le réseau mesh est toujours valide. S'il n'existe pas de planification du flux S^f réalisable pour le programme linéaire en variables binaires ci-dessus alors le flux est rejeté. Le problème d'admission d'un flux présenté ci-dessus est très difficile et est, plus précisément NP-complet. Un problème est NP-complet ou NP-difficile s'il vérifie les deux propriétés suivantes (Garay et David 1990):

- toute solution à ce problème peut être vérifié en temps polynomiale, il appartient donc à la classe des problèmes NP.
- il n'existe aucune algorithme en temps polynomial qui permette pour toutes les instances du problème de déterminer si un nouveau flux peut être admis ou non, à moins que P=NP. P représente la classe de problèmes qui peuvent se résoudre en temps polynomial. Il est communément admis qu'un problème de classe NP-complet (qui appartient donc également à la classe NP) ne peut pas se résoudre efficacement (en temps polynomiale) et donc appartenir à la classe de complexité P. Ainsi, un problème NP-complet est toujours de classe NP mais jamais de classe P à moins que P=NP.

Par la suite, dans cette section, nous apporterons une preuve de la NP-complétude de notre problème. Prouver la NP-complétude de notre problème, nous a permis d'éviter l'écueil de chercher un algorithme qui solutionne notre problème en un temps raisonnable (c.à.d. polynomiale). De plus cette preuve, pourra, par la suite, être réutilisée par toute personne souhaitant également solutionner un problème de contrôle d'admission d'un flux dans un réseau mesh planifié.

Garay *et al.*, dans l'un des livres les plus célèbres sur la théorie de la NP-complétude (Garay et David 1990), proposent une méthode en quatre étapes pour prouver qu'un problème B est NP-complet :

- Prouver que le problème appartient à la classe de complexité NP. Un problème appartient à la classe de complexité NP si, pour toute solution du problème, on peut vérifier en temps polynomial si elle est valide ou non.
- Sélectionner un problème A déjà connu comme NP-complet
- Construire une fonction f qui transforme l'ensemble des instances du problème A en une instance du problème B (voir figure 26).
- Prouver que f est polynomiale.

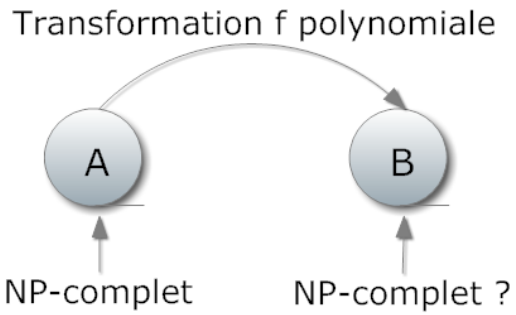


Figure 26 : Le problème A qui est NP-complet est transformable en un problème B via une fonction polynomiale f. Si B est également NP alors le problème B est NP-complet.

Afin de prouver que le problème d'admission d'un flux en négligeant le délai ou en supposant que le délai maximum toléré par le flux est infini est NP-complet, nous introduisons deux lemmes que nous prouvons par la suite.

Lemme 1. Toute solution du problème de contrôle d'admission est vérifiable en un temps polynomial et est donc NP.

Preuve du lemme 1 : Pour prouver qu'une solution de notre problème, c.à.d. qu'une matrice de planification de flux S^f, vérifie le problème d'admission de flux, il faut contrôler que cette matrice respecte les six contraintes du programme linéaire à variables binaires présenté plus tôt précédemment :

1. La vérification du respect de la première contrainte nécessite de contrôler si l'ensemble des liens de la route p_f ont N_f slots planifiés pour le flux sur leur N_p slots planifiables. La vérification de cette contrainte s'effectue en O($|p_f| * N_p$).
2. Pour vérifier que la matrice respecte la seconde contrainte, il faut contrôler que l'ensemble des liens n'appartenant pas au chemin du flux ne possède aucun slot pour émettre le flux. La vérification de cette contrainte s'effectue en O((|E|-$|p_f|$) $* N_p$) car $|E| - |p_f|$ liens n'appartiennent pas au chemin du flux et que chaque lien a N_p slots planifiables.
3. La vérification du respect de la troisième contrainte nécessite de contrôler si chaque lien pour chaque slot planifiable n'est planifié qu'une seule et unique fois. Cette vérification s'effectue en O(N_p*|E|) comme il y a N_p slots planifiables et $|E|$ liens.
4. Pour vérifier que la solution respecte la quatrième contrainte, il faut contrôler si chaque lien pour chaque slot considère le modèle d'interférence additif lorsqu'il envoie un paquet de données. Pour estimer si un lien peut émettre selon le modèle d'interférence additif sur un lien, la puissance à laquelle le nœud du lien reçoit le signal sur la somme doit être divisé par la somme de la puissance thermique du nœud et la somme de l'interférence causé par l'ensemble des autres liens du réseau. Le calcule de la somme de l'interférence causé par l'ensemble des autres liens du réseau s'effectue en O(|E|). Ainsi, La vérification de cette contrainte s'accomplit donc en O(|E|*$|N_p|$*|E|) .
5. Pour vérifier que la solution respecte la quatrième contrainte, il faut contrôler si chaque lien pour chaque slot considère le modèle d'interférence additif lorsqu'il envoie un acquittement. La vérification de cette contrainte, comme la précédente s'accomplit donc en O(|E|*$|N_p|$*|E|).
6. La vérification du respect de la dernière contrainte nécessite de contrôler si parmi l'ensemble des liens émettant sur le même slot, aucun ne possède en commun, avec un autre lien, un nœud. La vérification de cette contrainte s'effectue en $O(|E| * |V| * N_p * 2)$.

La vérification d'une solution pour le problème d'admission d'un flux lorsque le délai maximum du flux est fixé à $+\infty$ s'effectue donc en $O(|p_f| * N_p + (|E| - |p_f|) * N_p + N_p * |E| + N_p * |E|^2 * 2 + |E| * |V| * N_p * 2)$ et donc en temps polynomial. Le problème d'admission d'un flux lorsque le délai maximum du flux est fixé à $+\infty$ appartient donc à la classe de problème NP ; le lemme 1 est donc vérifié.

Lemme 2. L'ensemble des instances d'un problème de k-coloration d'un graphe pré-colorié planaire extérieur biparti, connu pour être NP-complet, peuvent être transformées via une fonction polynomiale en une instance d'un problème d'admission d'un flux lorsque le délai est fixé à $+\infty$.

Preuve du lemme 2 : Dans (Marx 2005) (Jiri 2003), les auteurs prouvent que le problème d'extension de la k-coloration des liens d'un graphe pré-colorié planaire extérieure biparti est un problème NP-complet. Un graphe est biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles U et V telle que chaque arête ait une extrémité dans U et l'autre dans V (voir figure 27). Un graphe est planaire extérieur s'il est non dirigé et s'il peut être dessiné dans le plan sans et de telle façon qu'aucun sommet ne soit entouré par des arêtes (voir figure 27). La k-coloration des liens d'un graphe consiste à attribuer à chaque lien une couleur parmi un ensemble de k couleurs, telle que deux sommets adjacents ne possèdent pas la même couleur. Les auteurs de (Marx 2005) (Jiri 2003), montrent que pour un graphe planaire extérieure G(V^{col},E^{col}) bipartie, si un sous-ensemble d'arêtes du graphe on déjà été k-coloriés, alors k-colorier l'ensemble du graphe est un problème NP-complet.

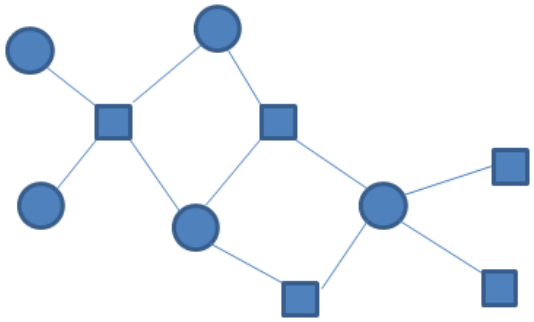


Figure 27 : Graphe biparti planaire extérieur. Chaque arête a un sommet carré et un sommet rond ; le graphe est donc bien biparti.

Afin de prouver que l'ensemble des instances du problème de la k-coloration des liens d'un graphe pré-colorié planaire extérieure biparti peuvent être transformées en une instance de notre problème, nous utilisons la preuve par restriction (Garay et David 1990). La preuve par restriction consiste à montrer que le problème, dont on souhaite démontrer la NP-complétude, dans certains cas, contient le problème qui est reconnu comme NP-complet. La difficulté de cette preuve réside dans la restriction à effectuer sur le problème dont on souhaite démontrer la NP-complétude pour montrer qu'il est identique au problème NP-complet connu. Il est communément admis (Garay et David 1990) que cela ne requiert pas que le problème à restreindre et le problème NP-complet soit de parfaits dupliquas, mais qu'il y est une relation « évidente » de correspondance entre les deux.

Nous supposons la restriction suivante sur notre problème d'admission d'un flux dont le délai maximum requis est fixé à l'infini. Soit un graphe $G(V'', E'')$ biparti extérieur qui représente notre réseau mesh. Supposons qu'un ensemble de flux ont déjà été admis sur le réseau, chacun requiert un slot par fenêtre de planification par lien de sa route pour satisfaire sa bande passante. De plus, tous les chemins de l'ensemble des flux sont totalement disjoints, ils ne possèdent donc aucun lien de

l'ensemble E'' en commun, ainsi chaque lien de E'' est associé à un seul et unique slot. On souhaite admettre un nouveau flux f' qui requiert un slot par fenêtre de planification et un délai maximum fixé à l'infini. Ce nouveau flux f' passe par une route valide qui comprend l'ensemble des liens encore non planifiés du réseau. La puissance de réception, au niveau d'un nœud u, d'un signal envoyé par un nœud v est de 0 si $(u,v) \notin E''$ et 2 si $(u,v) \in E''$. On fixe β à 1 et la puissance thermique de l'ensemble des nœuds à 0.5. Ainsi, les nœuds d'un lien (u,v) peuvent recevoir un signal avec une puissance supérieure à 0, seulement si le nœud émetteur de ce signal appartient à un lien de E'' qui est en conflit primaire avec le lien (u,v). Il y a interférence si un lien émet simultanément avec un lien avec lequel il est en conflit primaire ou si le modèle d'interférence additif n'est pas respecté. Or, il n'y a dans la restriction du problème d'admission que l'on vient de proposer, aucun risque d'interférence selon le modèle d'interférence additif car, si aucun conflit primaire n'existe, alors le SINR de chaque lien planifié $(u,v) \in E''$ lors de l'envoi de données (voir équation 59) ou lors de l'envoi de l'acquittement (voir équation 60) est toujours supérieur à β comme le prouve les équations suivantes :

$$\frac{P_{uv}}{P_{vv} + \sum_{\forall (i,j), e_y \in \Gamma} P_{iv}} = \frac{2}{0.5 + 0} > \beta \qquad \text{Équation 59}$$

$$\frac{P_{vu}}{P_{uu} + \sum_{\forall (i,j), e_y \in \Gamma} P_{ju}} = \frac{2}{0.5 + 0} > \beta \qquad \text{Équation 60}$$

avec Γ l'ensemble des liens E'' moins ceux qui sont en conflits primaires avec le lien (u,v). Avec la restriction de notre problème proposé, les nœuds du lien (u,v) peuvent recevoir le signal d'un nœud avec une puissance supérieure de 0, seulement si ce signal est émis par un nœud appartenant à un lien qui est en conflit primaire avec (u,v), ainsi $\sum_{\forall (i,j), e_y \in \Gamma} P_{iv} = 0$ et $\sum_{\forall (i,j), e_y \in \Gamma} P_{ju} = 0$. Un lien subit une interférence que si ce dernier est en conflit primaire. La fenêtre de planification possède N_p slots planifiables.

Cette instance du problème d'admission d'un flux est identique à un problème d'extension de la N_p-coloration des liens d'un graphe précolorié planaire extérieurebipartie. En effet, puisque le modèle d'interférence additif y est ignoré, un lien peut être panifié sur un slot, uniquement s'il n'est pas en confit primaire, c.à.d. si ses liens adjacents ont été planifiés sur un autre slot. Si l'on associe un numéro de slot à une couleur alors le graphe G(V'',E") avant l'admission du nouveau flux f' peut être considéré comme un graphe planaire extérieur biparti pré-colorié avec N_p couleurs ; un lien possède la couleur associé au slot pendant lequel il émet un flux. Le problème de l'admission du nouveau flux

consiste donc à trouver, pour l'ensemble des liens encore non planifiés, un numéro de slot pendant lequel chacun peut émettre sans conflit primaire et donc, à associer à chacun de ces liens un numéro de slot encore non utilisé par ses liens adjacents. Si, encore une fois, un numéro de slot planifiable est associé à une couleur, ce problème revient alors à trouver une N_p-coloration des liens d'un graphe pré-colorié planaire extérieur bipartie. L'instance du problème d'admission d'un flux présenté ci-avant est identique à un problème de N_p-coloration des liens d'un graphe pré-colorié planaire extérieur biparti. Ainsi, d'après (Garay et David 1990), le problème de k-coloration des liens d'un graphe pré-colorié planaire extérieur biparti peut être transformé via une fonction polynomiale en une instance de notre problème d'admission.

Théorème 1 : Le problème d'admission d'un flux dans un réseau mesh intégrant un système de planification est un problème NP-complet.

Preuve du théorème 1 : D'après le lemme 1 notre problème d'admission est NP. De plus, d'après le lemme 2, toute instance du problème de k-coloration des liens d'un graphe pré-colorié planaire extérieur biparti peut être transformée en temps polynomial en une instance du problème d'admission d'un flux en négligeant le délai. Ces deux lemmes prouvent donc que notre problème est NP-complet.

Le problème de contrôle d'admission d'un flux f lorsqu'on considère la demande d_f^{max} de délai maximum du nœud (cette dernière n'étant plus fixé à l'infini) est encore plus difficile, puisqu'il rajoute une contrainte supplémentaire qui n'est pas linéaire. Cette sixième contrainte vérifie que le délai du flux est inférieur au délai maximum requis par ce dernier :

$$d_f \leq d_f^{max} \qquad \text{Équation 61}$$

avec d_f obtenue via l'équation 51. Dans cette section, nous avons montré que le problème d'admission d'un flux est NP-complet, puisque toute solution du problème peut être vérifiée en temps polynomial et que, si l'on restreint notre problème, ce dernier est identique à un problème connu pour être NP-complet, la k-coloration d'un graphe biparti extérieur.

4.4. Notre solution de contrôle d'admission avec planification des liens

Notre solution offre un cadre complet pour la réalisation d'un système de contrôle d'admission pour un réseau mesh sans fil avec planification de liens. Le contrôle d'admission que nous proposons comprend :

- Un mécanisme de diffusion des requêtes d'admission d'un flux.
- Un algorithme d'admission et de planification d'un nouveau flux. Cet algorithme est effectué par un portail suite à la réception d'une demande d'admission d'un flux. Il établit si le flux peut être admis ou non sur le réseau ainsi que la planification de ce dernier en cas d'admission.
- Un mécanisme de diffusion de l'admission et de la planification d'un flux.

4.4.1. Algorithme d'admission et de planification d'un nouveau flux

Nous proposons dans cette section, un algorithme itératif d'admission et de planification d'un nouveau flux dans un réseau mesh planifié. Cet algorithme établit, suite à la demande d'admission d'un nouveau flux, une planification pour ce dernier qui respecte son délai et son débit ainsi que ceux des flux préalablement admis. Si l'algorithme échoue à établir la planification du nouveau flux, la demande d'admission de ce dernier est rejetée sinon elle est acceptée.

Chaque itération de l'algorithme se décompose en deux parties. Dans un premier temps, notre algorithme, pour une demande d'admission d'un flux, recherche une panification de ce dernier qui respecte ses contraintes en termes de bande passante et qui est donc solution du programme linéaire à variables binaires présenté précédemment. Dans un second temps, l'algorithme vérifie si cette planification respecte les contraintes en termes de délai du flux. Si elle les respecte, alors le flux est admis suivant la dernière planification retournée par l'algorithme, sinon la procédure est répétée afin de trouver une nouvelle solution au PLVB. L'algorithme s'arrête lorsqu'il n'existe plus de nouvelle solution au PLVB ou lorsque le nombre d'itérations effectué est supérieur à la valeur seuil $Nmax$ ou lorsqu'une solution est trouvée au PLVB qui respecte le délai demandé par le flux. Si aucune solution n'est trouvée alors le flux est rejeté, sinon le flux est accepté selon la dernière planification établit par l'algorithme. La valeur seuil $Nmax$ doit être fixée de telle manière que le

temps d'exécution de l'algorithme ne soit pas trop long. Sa valeur dépend donc principalement de la taille des données de l'algorithme et du CPU des routeurs portails d'accès.

Il existe déjà dans la littérature des algorithmes permettant de résoudre des programmes linéaires en variables binaires comme la méthode des plans sécants (Cornuéjols 2008), la génération de colonnes (Savelsbergh 2009) et la méthode de séparation et évaluation (Sierksma, Dam et Tijssen 1996).Ces algorithmes peuvent s'effectuer en un temps exponentiel pour certaines instances particulièrement complexes (un graphe avec de nombreux liens et sommets). Cependant, leur temps d'exécution est correct lorsque l'instance du problème possède une taille raisonnable, ce qui est généralement le cas d'un réseau mesh.

La méthode de séparation et évaluation (Sierksma, Dam et Tijssen 1996) permet d'obtenir une résolution exacte des programmes linéaires en nombres entiers mais également des programmes linéaires à variables binaires. Cette méthode effectue une énumération intelligente de l'espace des solutions afin d'accélérer la résolution du problème. L'espace des solutions de notre problème est de $2^{|E|*|N_p|}$ car il y a $|E|*|N_p|$ éléments binaires dans une matrice S_f.Appliquée à des problèmes NP-difficiles, la méthode de séparation et évaluation reste bien sûr exponentielle, mais sa complexité en moyenne est bien plus faible que pour une énumération complète. Elle pallie donc le manque d'algorithmes polynomiaux pour des problèmes de taille moyenne comme notre problème d'admission de flux dans un réseau mesh où la taille du réseau est généralement limitée.

Il existe plusieurs méthodes d'évaluation et séparation, dont l'une d'elle est la méthode arborescente de Dakin (Dakin 1965) (Dankin 1965). Cette méthode représente le problème à résoudre sous forme d'arborescence. Elle fait également appelle à un autre algorithme qui permet de résoudre des programmes linéaire à variables réelles (Guéret, Prins et Seveaux 2003), généralement, l'algorithme du simplexe (Dantzig 1990). Notre algorithme a pour but de trouver la matrice S_f, solution du problème qui a pour fonction objectif :

$$min \sum_{i=1}^{i=N_p} \sum_{j=1}^{j=|E|} s_{ij}^f \quad avec\ s_{ij}^f \in \{0,1\}$$ **Équation 62**

et pour contraintes, les six du PLVB présentés précédemment et formulées par les inéquations 53, 54, 55, 56, 57 et 58. Cette méthode initialise z', la meilleure solution obtenue pour la fonction objectif à $+\infty$. Elle construit ensuite une arborescence où chaque nœud n_i possède :

- un PLVB noté P_i.

- une matrice S_i^f dont les valeurs peuvent être binaires ou entières. Elle est obtenue par résolution via l'algorithme de simplexe du PLVB$P_i$relaxé du nœud via l'algorithme c.à.d. du PLVB P_i lorsqu'on considère que l'ensemble des variables de la matrice S_i^f peuvent prendre des valeurs non entières dans l'intervalle [0,1].
- une valeur objectif $\underline{z_i}$. $\underline{z_i}$ est le plus petit nombre entier supérieur ou égale à la valeur de la fonction objectif du PLVB P_i relaxé.
- un état, activé ou non activé. Si un nœud est dans l'état activé alors, il peut encore être sondé par l'algorithme sinon, il ne peut plus l'être.

L'arborescence est initialisée par la création d'un nœud n_0 dontle PLVB P_0 est le PLVB d'origine, la matrice S_f^0 est la solution du PLVB P_0relaxé et la valeur objectif $\underline{z_0}$ est le plus petit nombre entier ou égale à la valeur de la fonction objectif du PLVB P_0 relaxé. Le résultat de la fonction objectif de notre PLVB initiale non relaxé ne pourra jamais être inférieur à la valeur $\underline{z_0}$, car ce dernier possède une contrainte supplémentaire par rapport au PLVB initiale relaxé ; chaque élément de la matrice solution doit être binaire. Ainsi, l'algorithme fixe z^* la borne inférieure du résultat de la fonction objectif de notre PLVB à $\underline{z_0}$. L'algorithme place le nœud $n_0$dans l'état activé et crée ensuite l'arborescence itérativement à partir du nœud initial n_0 selon la méthode suivante composée de trois étapes :

1. Sélectionner parmi les nœuds récemment construits de l'arborescence qui sont actifs et qui n'ont pas encore été sondés le nœud n_i qui a la plus petite valeur objectif $\underline{z_i}$. Si c'est la première itération choisir le nœud initial n_0. Pour le nœud n_i sélectionné, l'algorithme choisit un élément s_{ij}^f de sa matrice S_i^f du nœud qui n'équivaut ni à 0 ou 1. Deux nœuds fils sont crées pour le nœud n_i que l'on note respectivement n_j et n_{j+1}. Chaque nœud fils est associé à un nouveau PLVB équivalent à celui du nœud père mais avec une contrainte supplémentaire. Pour l'un des nœuds, la contrainte supplémentaire est, s_{ij}^f=1 et pour le second nœud, s_{ij}^f=0.
2. Résoudre le PLVB relaxé de chaque nœud fils avec la méthode du simplexe. Si l'algorithme du simplexe a une solution pour un nœud fils n_j, alors ce dernier est associé à une matrice S_j^f solution de son PLVB relaxé et une valeur objectif $\underline{z_j}$ qui est le plus petit nombre entier ou égale à la valeur de la fonction objectif de son PLVB P_j relaxé. Si l'algorithme du simplexe n'a pas de solution pour l'un des nœuds fils, ce dernier est désactivé et ses branches ne seront pas sondées.

3. Etudier la valeur $\underline{z_j}$ de chaque nœud fils. Si la valeur objectif $\underline{z_j}$ de l'un des de nœuds fils est supérieure à notre meilleure solution z', alors le nœud est désactivé et ses branches ne seront pas explorées. Si l'un des nouveaux nœuds a une valeur objectif $\underline{z_j}$ inférieure à la meilleure solution z' et si sa matrice S_j^f possède uniquement des valeurs binaires, alors z' est fixé à $\underline{z_j}$ et le nœud est désactivé.

L'algorithme s'arrête lorsqu'un nœud n_i obtient une matrice S_i^f binaire où $\underline{z_i} = z^*$ ou, lorsque l'ensemble des nœuds de l'arborescence ont déjà été explorés ou sont désactivés. Si à la fin de l'algorithme z' équivaut à $+\infty$, alors l'algorithme retourne qu'il n'y a pas de solution au problème sinon, il retourne la matrice S_i^f du nœud n_i qui parmi l'ensemble des nœuds possédant une matrice dont l'ensemble des éléments sont binaires a la plus petite valeur objectif. La figure 28 présente l'arborescence de la méthode de Dakin pour notre problème.

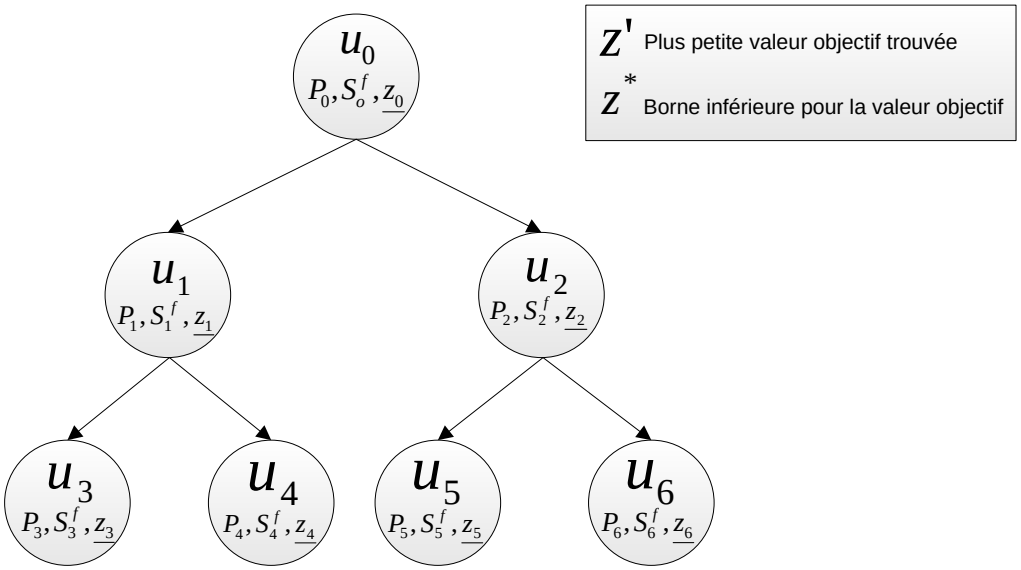


Figure 28 : Arborescence de la méthode de Dakin

Nous proposons un algorithme qui modifie l'étape 1 de l'algorithme de Dakin afin de retourner une matrice de planification solution au problème PLVB telle que le délai du flux engendré par cette planification soit minimum pour une sélection de N_f slots au niveau du premier lien e_0 du chemin du flux. En d'autres termes, pour une réservation $\Psi_{j'}^{f,e_0}$ il n'existe pas de matrice de planification du flux f assurant à ce dernier un plus petit délai que celui engendré par la planification retourné par notre algorithme. Lors de l'étape 1 de la méthode de Dakin, après qu'un nœud u_z ait été sélectionné, notre algorithme choisit l'élément s_{ij}^f de la matrice S_z^f tel que :

- s_{ij}^f n'est pas binaire.

- l'indice i de s_{ij}^f est également celui du premier lien e_i du chemin p_f qui n'a pas encore N_f slots planifiés, c.à.d. qui ne possède au niveau du PLVB de u_z, des contraintes, forçant la $i^{ème}$ ligne de la matrice S_z^f à avoir N_f éléments à 1.
- l'indice j de s_{ij}^f représente le numéro du premier slot planifiable du lien e_i qui, si le lien e_i a déjà été planifié sur z slots (c.à.d. si le PLVB de u_z possède des contraintes, forçant la $i^{ème}$ ligne de la matrice S_z^f à avoir z élèments à 1), est situé juste après le $z+1$ ème slot planifié par e_{i-1}, le lien précédent sur le chemin du flux.

L'idée est de réserver, pour un slot planifié par un lien e_{i-1}, le slot le plus proche de ce dernier afin que le temps d'attente d'un paquet au niveau d'un nœud u_i soit le plus petit possible. Par exemple, soit Ψ_j^i le numéro de slot planifié par e_i pour émettre les paquets reçus de e_{i-1} au slot numéro Ψ_z^{i-1}, alors tout paquet envoyé à Ψ_j^{i-1} a au niveau du nœud émetteur u_i du lien e_i a un délai de :

$$d = \begin{cases} \Psi_j^i - \Psi_z^{i-1} \; si \; \Psi_j^i > \Psi_z^{i-1} \\ \Psi_j^i + N - \Psi_j^{i-1} \; si \; \Psi_j^i < \Psi_z^{i-1} \end{cases} \qquad \text{Équation 63}$$

L'équation ci-dessus permet de calculer le délai du paquet au niveau du nœud u_i lorsque, u_i fait suivre le paquet pendant la fenêtre de planification où il le reçoit ($\Psi_j^i > \Psi_z^{i-1}$) et, lorsque u_i fait suivre le paquet pendant la fenêtre de planification suivante à celle où il le reçoit ($\Psi_j^i < \Psi_z^{i-1}$). En choisissant le slot numéro Ψ_z^i tel qu'il soit le premier slot planifiable de e_i situéaprès le numéro de slot Ψ_z^{i-1}, le délai du paquet envoyé sur lien e_i au slot numéro Ψ_j^i est minimisé. Cette modification de l'étape 1 est appliquée à chaque itération de l'algorithme. En minimisant le délai de tous les paquets au niveau de chaque nœud du lien, le délai du flux est ainsi minimisé. La modification apportée à la méthode de Dakin permet ainsi pour N_f slots choisit arbitrairement au niveau du premier lien du chemin d'obtenir une planification du flux ayant un délai minimal. La 29 présente la méthode de Dakin modifiée, avec en italique la modification apportée à la méthode originelle.

Entrée: Le PLVB d'origine
Sortie: La matrice de planification du flux S^f

1: Initialiser z' à $+\infty$
2: Initialiser un nœud u_0, lui associer le PLVB d'origine, la matrice de planification S_0^f de son PLVB relaxé et $\underline{z_0}$ le plus petit nombre entier supérieur ou égale à la valeur de sa fonction objectif.
3: Initialiser z^* à $\underline{z_0}$ et u_0 comme noeud actif
4: **Tant que** il y a des noeuds actifs et qu'aucun noeud a une matrice binaire avec une valeur objectif à $\underline{z_0}$ **Faire**
5: Sélectionner le nœud u_x non sondé ayant la plus petite valeur objectif $\underline{z_x}$
6: *Choisir une variable non binaire s_{ij}^f de la matrice S_x^f du nœud u_x tel que l'indice i de s_{ij}^f soit également l'indice du premier lien e_i du chemin qui n'a pas encore N^f slots planifiés et l'indice j soit le numéro du 1^{er} slot encore non planifié du lien e_i (sachant que e_i en a déjà $z-1$ de réservés) situé juste après $z^{ème}$ slot planifié de e_{i-1}.*
7: Créer deux noeuds fils actifs à u_x, les associer au PLVB de u_x avec pour contrainte supplémentaire pour le premier $s_{ij}^f = 1$ et pour le second $s_{ij}^f = 0$
8: **Pour** chaque noeud fils u_y **Faire**
9: **Si** son PLVB est soluble **Alors**
10: Calculer sa valeur objectif $\underline{z_y}$ et sa matrice S_y^f
11: **Si** $\underline{z_y} = z^*$ et S_y^f binaire **Alors**
12: **Retourner** la solution S_y^f
13: **Sinon Si** $\underline{z_y} > z'$ **Alors**
14: Désactiver le noeud u_y
15: **Sinon Si** $\underline{z_y} < z'$ et S_y^f binaire **Alors**
16: Fixer z' à $\underline{z_y}$
17: **Fin Si**
18: **Fin Si**
19: **Fin Pour**
20: **Fin Tant que**
21: **Si** $z' == +\infty$ **Alors**
22: **Retourner** $S_i^f = null$
23: **Sinon**
24: **Retourner** la matrice binaire S_i^f qui a la plus petite valeur objectif
25: **Fin Si**

Figure 29 : Algorithme de Dakin modifié, qui retourne une solution au PLVB d'entrée. Si l'algorithme retourne null,alors il n'a pas de solution.

Pour résoudre le problème de contrôle d'admission d'un flux en termes de délai et de bande passante, nous proposons une approche itérative, intégrant l'algorithme de Dakin modifié présenté ci-dessus, en deux étapes :

1. Trouver une solution au PLVB d'admission d'un flux en négligeant le délai avec la méthode de Dakin modifiée
2. Vérifier si la planification retournée par l'étape précédente respecte les contraintes en termes de délai du flux.

Si la deuxième étape échoue alors, le PLVB de départ est modifié et l'approche est réitérée et ce jusqu'à ce que la planification produite respecte les contraintes du flux en termes de délai ou que l'algorithme de Dakin ne trouve plus de solution au PLVB. A chaque itération le PLVB est modifié afin qu'une solution ne soit pas obtenue plusieurs fois. L'approche de la coupe binaire proposée par Balas et *al* (Balas et Jeroslow 1972) (Tsai, Ming-Hua et Yi-Chung 2006) permet d'éviter ce phénomène. Balas et *al* proposent de réécrire à chaque fois qu'une solution du PLVB est trouvée, un nouveau PLVB qui possède les mêmes solutions que le problème originel moins les solutions déjà trouvées. Ainsi, à chaque fois qu'une planification de flux est obtenue via l'algorithme de Dakin modifié ne respectant pas le délai maximum de notre flux, le PLVB est réécrit. Dans ce nouveau PLVB, l'ensemble des planifications qui commencent par les N_f slots sélectionnés pour le premier lien du flux de la dernière planification obtenue sont éliminées du PLVB. En effet, si l'algorithme modifié de Dakin retourne une planification pour un flux, il n'existe alors pas d'autre planification possédant les mêmes N_f slots réservés pour le premier lien assurant un délai inférieur au flux. Notre algorithme se termine, lorsqu'il n'existe plus de nouvelles solutions au PLVB, ou lorsque le nombre d'itérations atteint un certain seuil noté $Nmax$. La valeur de $Nmax$ doit être ajustée selon la taille du problème, la taille du réseau, la valeur de N_c, la puissance de calcul du nœud point d'accès à Internet. Si aucune planification n'est trouvée respectant à la fois la bande passante et le délai, alors le flux est rejeté sinon le flux est accepté. Notre approche est résumée par le diagramme de la figure 30.

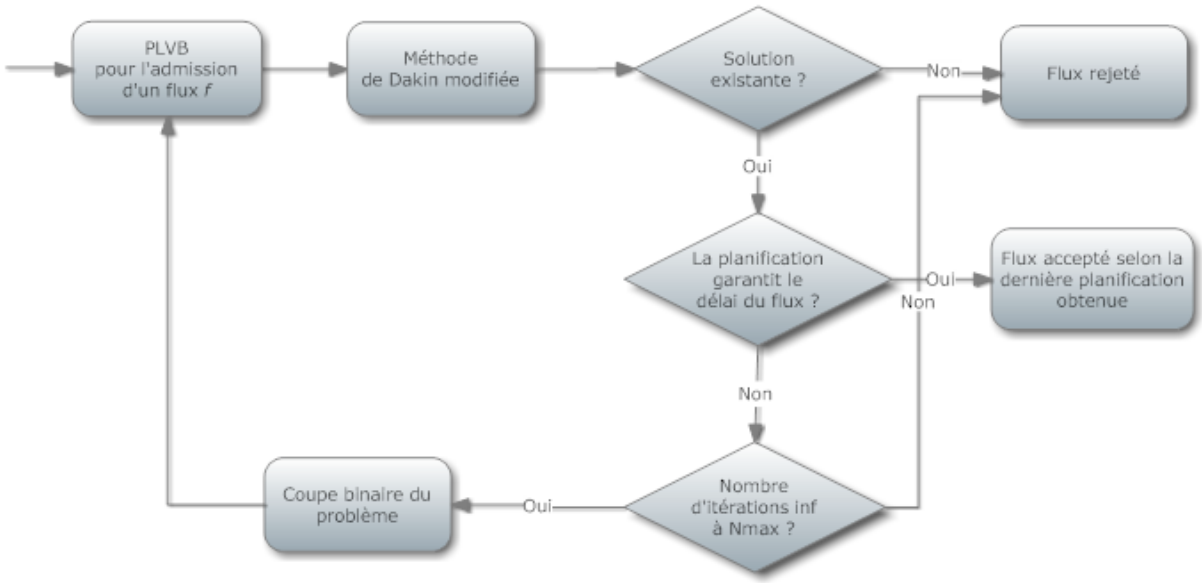


Figure 30 : Approche itérative de l'algorithme d'admission et de planification d'un flux

4.4.2. Contrôle d'admission d'un flux

Le contrôle d'admission proposé dans cette thèse repose sur un schéma de routage réactif tel que le protocole AODV (Perkins, Royer et Das 2003). Notre contrôle d'admission d'un flux se déroule en trois temps :

1. Dans un premier temps, la source envoie une requête d'admission pour le flux.
2. Dans un second temps, le portail recevant la requête effectue l'algorithme d'admission du flux présenté précédemment afin de décide si le flux est accepté ou non. Si le flux est accepté, il détermine la planification de ce dernier.
3. Finalement, le portail envoie une réponse à la demande d'admission du flux contenant, si le flux est accepté, sa planification.

4.4.3. Envoi d'une requête d'admission d'un flux

Chaque source, souhaitant émettre un flux, diffuse un paquet de découverte de route, RREQ. Ce paquet de découverte de route précise :

- le nombre de slots N_f minimum que chaque nœud de la route doit réserver par fenêtre de planification pour le flux.
- le délai d_f^{max} maximum requis pour le flux.
- un numéro de flux. Une source ne peut pas utiliser le même numéro pour deux flux dont elle est la source. Ainsi la source et le numéro d'un flux permettent d'identifier un flux de manière unique.
- le TTL du paquet.
- le portail d'accès à Internet qu'elle souhaite joindre.

Chaque nœud recevant le RREQ et qui n'est pas la destination vérifie si :

- le lien que vient de traverser le paquet est bidirectionnel. Il vérifie si le lien précédent est bidirectionnel afin, entre autre, de pouvoir transmettre la réponse d'admission du flux via la route empruntée par le RREQ.
- il n'a pas déjà reçu ce RREQ, c.à.d. si le nœud n'a pas déjà reçu récemment un RREQ en provenance de la même source avec un numéro de flux similaire.
- le TTL n'a pas expiré.

Si les vérifications échouent, alors le nœud détruit le paquet, sinon le nœud ajoute au paquet son identifiant et diffuse à son tour le paquet. La procédure continue jusqu'à ce que le nœud destination reçoive le RREQ. Le nœud destination vérifie alors si le dernier lien emprunté par le paquet est bidirectionnel ou non. S'il ne l'est pas alors le nœud élimine le paquet sinon il déclenche la procédure d'admission ou de rejet du flux. Pour rappel, chaque nœud émet le RREQ sur un slot non dédié à la planification car c'est un paquet de contrôle.

4.4.4. Procédure d'admission ou de rejet du flux

Le nœud destination est un portail d'accès à Internet. Tout portail d'accès à Internet dans le réseau connait l'ensemble des planifications de flux en cours sur le réseau. En effet, lorsqu'un portail d'accès établit une nouvelle planification pour un flux, il l'envoie aux autres portails du réseau mesh (via par exemple une ligne dédiée ou le réseau Internet). Ainsi, à chaque instant, les portails d'accès connaissant l'ensemble des planifications en cours sur le réseau puisqu'ils les ont soit eux-mêmes établies soit ils les ont appris d'un autre portail. On suppose que chaque nœud portail d'accès à Internet connait l'ensemble des puissances associées à chaque lien du réseau ainsi que l'ensemble des puissances thermiques des nœuds.

Lors de la procédure d'admission ou de rejet du flux, la destination extrait, du RREQ, le chemin emprunté par le flux p_f, ainsi que le nombre minimum de slots et le délai maximum demandés par le flux. Le portail a alors les informations suffisantes pour lancer l'algorithme d'admission et de planification du flux présenté précédemment et schématisé par la figure 30. Cet algorithme décide si le flux est admis ou non, et, s'il l'est, retourne la planification du flux. Si le flux est admis, alors la destination abandonnera les prochains paquets RREQ possédant le même numéro de flux et la même source du flux sinon, il continuera à traiter ces RREQ jusqu'à ce que le flux soit accepté ou qu'il n'en reçoive plus.

4.4.5. Envoi de la réponse d'admission ou de rejet d'un flux

Une fois la procédure d'admission terminée et si le flux est admis, la destination envoie à la source un paquet RREP où il précise, pour l'ensemble des liens composant le chemin du flux, les slots pendant lesquels ils doivent émettre le flux. Il envoie le RREP le long de la route empruntée par le RREQ, chaque nœud le recevant enregistre les numéros de slots pendants lesquels il doit émettre le

flux. Lorsque le paquet atteint la source, cette dernière peut commencer l'émission du flux. Chaque nœud du chemin connait alors les slots sur lesquels il doit transmettre le flux. La planification du flux au niveau d'un nœud devient inactive lorsque le nœud ne reçoit plus de paquets du flux depuis un certains laps de temps. Afin de valider notre solution et de vérifier ces performances, nous l'avons simulé sur le logiciel ns2. Les résultats de la simulation sont présentés dans la section suivante.

4.5. Evaluation du contrôle d'admission avec planification des liens

Notre solution a été validée sur le simulateur ns2 et ses performances ont été comparées avec le modèle classique composé de CSMA/CA pour le contrôle d'accès au médium et du protocole de routage AODV. Ils ont été étudiés en termes de perte de paquets, de capacité totale sur le réseau et de débit sur trois différentes topologies réseau, la topologie chainée, la topologie maillée et la topologie en croix (voir les figures 31, 32 et 33). Les nœuds rouges sur chaque figure, représentent un portail d'accès à l'Internet.

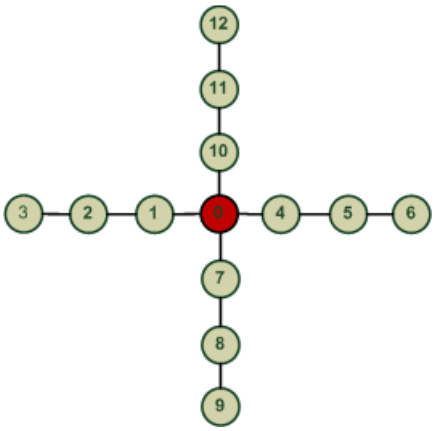


Figure 31 : Topologie en croix

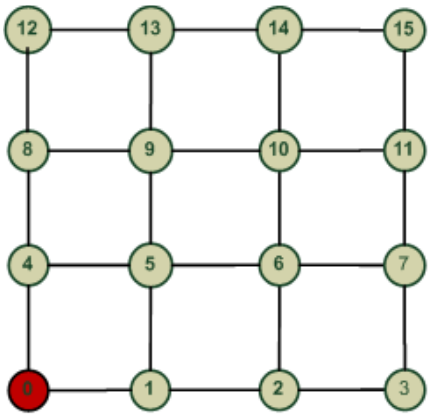


Figure 32 : Topologie maillée

Figure 33 : Topologie chainée

Les paramètres de la simulation sont présentés dans le tableau 7. Les valeurs des paramètres des couches MAC et physique reprennent celles de la norme IEEE 802.11g. La valeur du nombre N de

slots par fenêtre de planification est différente selon les topologies ; elle a été fixée pour chaque topologie de telle manière d'obtenir les meilleurs résultats possibles.

Tableau 7 : Tableau des paramètres de simulation du contrôle d'admission avec planifications des liens.

Niveau	Paramètre	Valeur
Propagation du signal	Modèle Two-ray-ground	
Modèle d'interférence	Modèle d'interférence additif	
Physique	Fréquence	54 Mbit/s
	PLCP préambule	20
Couche MAC	T_{slot}	260 μs
	N pour la topologie maillée	115
	N pour la topologie linéaire	116
	N pour la topologie en croix	116

Pour chaque topologie, le scénario suivant est réalisé : chaque nœud (sauf le portail d'accès) effectue une demande d'admission pour un flux vidéo à destination du nœud portail d'accès qui requiert un débit à 300 kbit/s et un délai inférieure à 150 ms. Les nœuds envoient une demande d'admission à une second d'intervalle les uns des autres et dans l'ordre croissant de leur identifiant. Par exemple, pour la topologie chainée, le nœud 0 effectue la première demande d'admission puis le nœud 1 puis le nœud 2, etc. Lorsqu'un nœud source reçoit un RREP qui précise que son flux est accepté, alors il commence à le transmettre. Chaque flux porte l'identifiant du nœud source qui l'émet.

Les figures 34, 35, 36, 37, 38 et 39 présentent le débit des flux en kbit/s dans le temps en seconde selon différentes topologies de réseau mesh (linéaire, en croix, maillée) et différents modèles (notre modèle et le modèle originel).

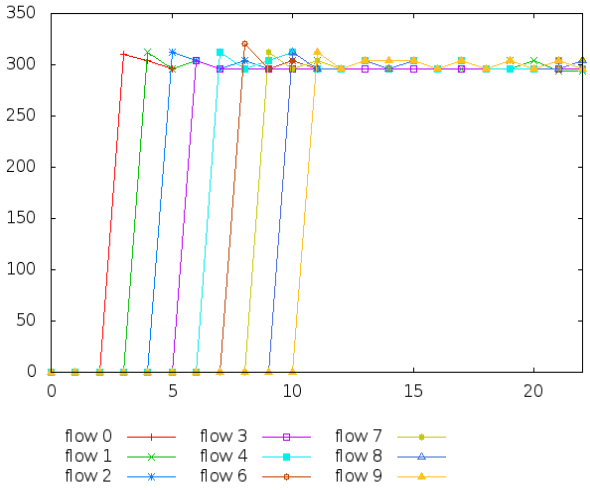


Figure 34: Débit des flux en kbit/s dans le temps (seconde) avec notre modèle sur la topologie chainée

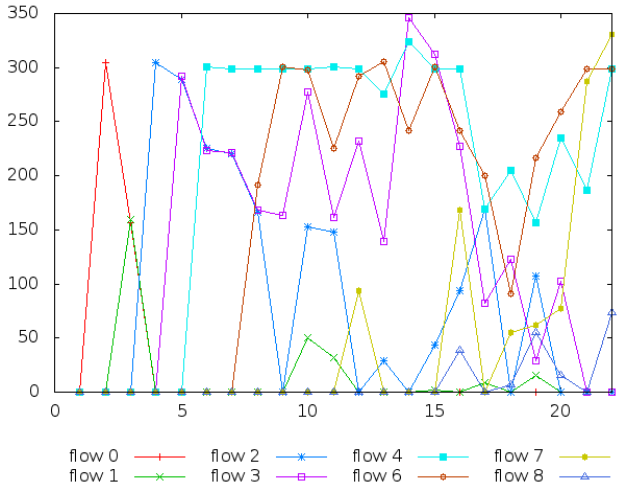


Figure 35 : Débit des flux en kbit/s dans le temps (seconde) avec le modèle originel sur la topologie chainée

Les figures 34 et 35 introduisent les résultats d'un réseau mesh à topologie chainée qui implémente respectivement notre modèle et le modèle originel. Sur le réseau à topologie chainée, notre modèle, admet 9 flux (flux 0, 1, 2, 3, 4, 6, 7, 8 ,9) et rejette un flux, le numéro 10 dont la demande d'admission est lancée par le nœud 10. Tous les flux admis obtiennent le délai qu'ils ont demandé de 300kbit/s. La demande d'admission pour le flux 10 est la dernière à être envoyée sur le réseau, celle-ci est refusée, car il n'y a alors plus suffisamment de slots disponibles et planifiables sur le réseau qui peuvent lui être attribués. Avec le modèle d'origine, tous les flux sont admis sur le réseau. Or, le flux 10 n'apparait pas sur la figure 35, on en déduit qu'aucun de ses paquets n'ont réussi à rejoindre le portail car le réseau était déjà saturé par les 9 autre flux déjà présent lorsque le nœud 10 effectue sa demande d'admission.

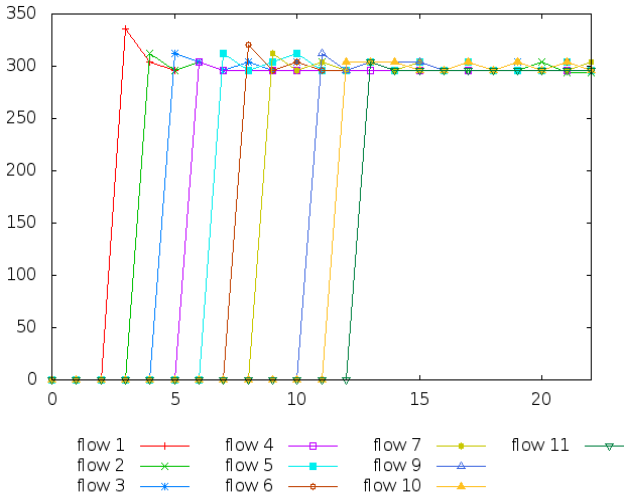


Figure 36 : Débit des flux en kbit/s dans le temps (seconde) avec notre modèle sur la topologie en croix.

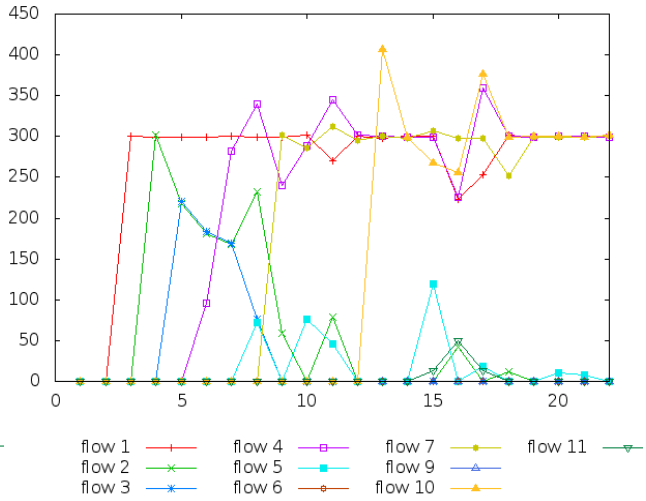


Figure 37: Débit des flux en kbit/s dans le temps (seconde) avec le modèle originel sur la topologie en croix

Les figures 36 et 37 introduisent les résultats d'un réseau mesh à topologie en croix qui implémente respectivement notre modèle et le modèle originel. Notre modèle, sur le réseau à topologie en croix admet 10 flux sur le réseau (flux 1, 2, 3, 4, 5, 6, 7, 9, 10, 11) et rejette deux flux, le flux 8et le flux 12 dont la demande d'admission est lancée par respectivement le nœud 8 et le nœud 12. Tous les flux admis obtiennent le délai qu'ils ont demandé de 300kbit/s. Avec le modèle d'origine, l'ensemble des flux sont admis sur le réseau. Or, les flux 12 et 8 n'apparaissent pas sur la figure 37, on peut en déduire que aucun de leurs paquets n'ont réussi à rejoindre le portail, car le réseau était déjà saturé par les autre flux déjà présents lorsqu'ils commencent à être émis. Aucun flux, dont au moins un paquet atteint le portail, n'obtient le débit qu'il requiert sur une longue période ; le débit des flux varie au cours du temps.

Les figures 38 et 39 introduisent les résultats d'un réseau mesh à topologie maillée qui implémente respectivement notre modèle et le modèle originel. Notre modèle dans le réseau à topologie en croix admet 10 flux sur le réseau (flux 1, 2, 3, 4, 5, 6, 7, 8, 9, 11) et rejette cinq flux, les flux 10, 12, 13, 14, 15 dont la demande d'admission est lancée par respectivement, le nœud 10, le nœud 12, le nœud 13, le nœud 14 et le nœud 15. Tous les flux admis obtiennent le délai qu'ils ont demandé de 300kbit/s. Avec le modèle d'origine, l'ensemble des flux sont envoyés sur le réseau. Or, les flux 14 et 15 n'apparaissent pas sur la figure 39, on en déduit qu'aucun de leurs paquets n'ont réussi à rejoindre le portail. Le débit des flux varient au cours du temps et aucun n'obtient le débit qu'il requiert.

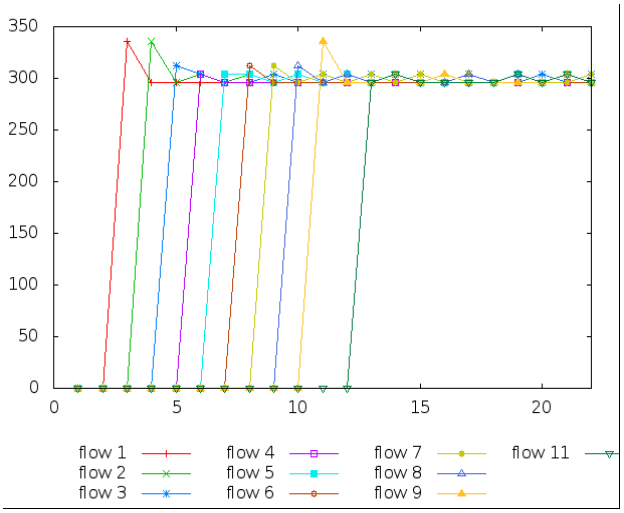


Figure 38 : Débit des flux en kbit/s dans le temps (seconde) avec notre modèle sur un réseau à topologie maillée.

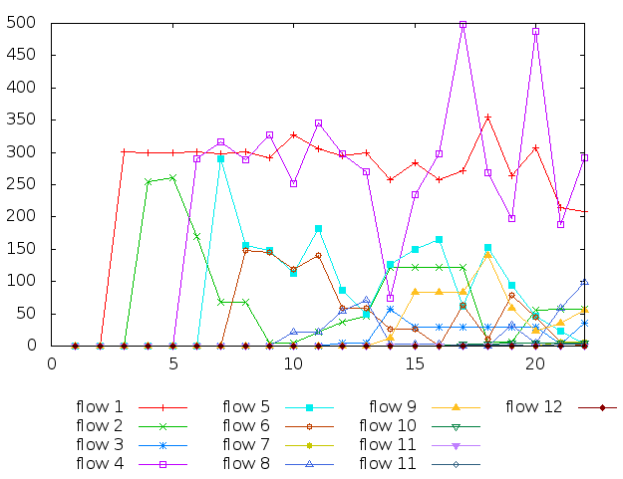


Figure 39 : Débit des flux en kbit/s dans le temps (seconde) avec le model originel sur un réseau à topologie maillée.

Les figures 40, 41 et 42 comparent via simulation le débit des flux obtenu avec notre solution, avec, le débit des flux obtenu avec le modèle originel.

La figure 40 présente les résultats obtenus avec un réseau mesh à topologie chainée. On remarque que tous les flux admis avec notre solution atteignent le débit qu'ils ont demandé, c.à.d. 300 kbit/s. Par contre, pour le protocole originel plus les nœuds sources sont loin du portail d'accès et plus leur débit est faible, en effet les deux nœuds les plus proches du portail (4 et 6) ont presque tous deux le débit de 300 kbit/s. Le réseau basé sur le protocole originel est saturé, et les nœuds situés loin du portail tel que le flux des nœuds 9 et 10, ont un débit nul.

La figure 41 présente les résultats obtenus avec un réseau dont la topologie est en croix. On remarque que notre solution accepte la demande de flux de tous les nœuds sauf celle du nœud 12. Tous les flux acceptés obtiennent dans notre modèle le débit demandé et le réseau évite la congestion en refusant un flux. Par contre, avec le modèle originel, seuls les flux dont les nœuds

sources sont situés près du portail d'accès (1, 4, 7, 10) ont un débit proche de 300 kbits/s, les autres sont en famine ; le réseau subit une congestion.

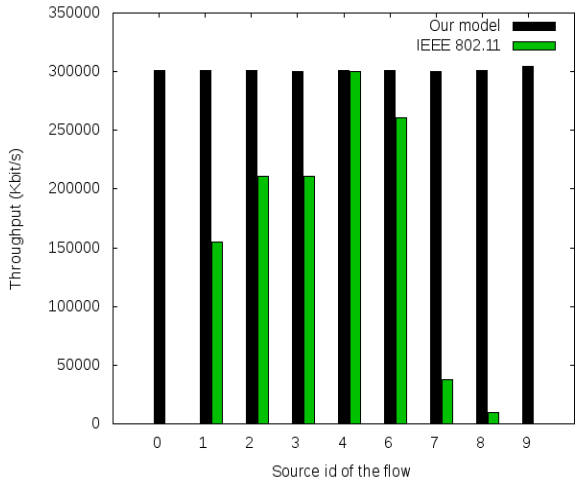


Figure 40 : Comparaison du débit des flux du réseau à topologie chainée

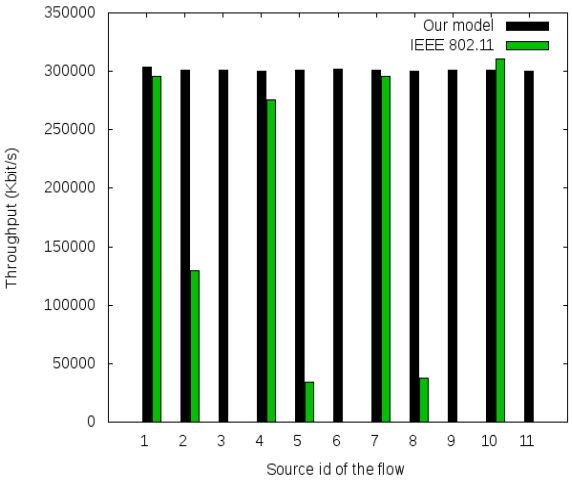


Figure 41 : Comparaison du débit des flux pour la topologie en croix

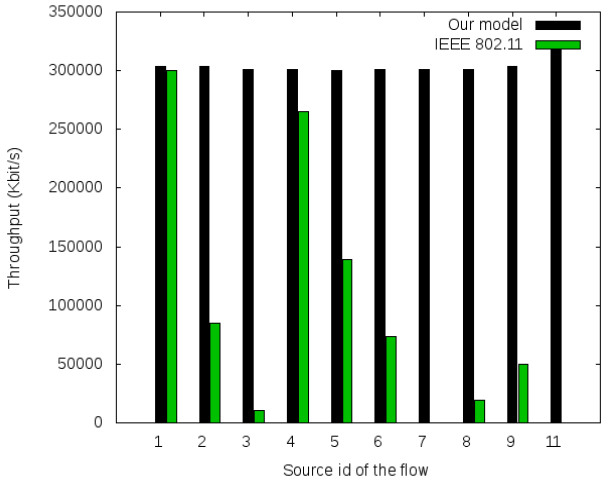


Figure 42 : Comparaison du débit des flux pour la topologie maillée

La figure 42 présente les résultats obtenus avec un réseau dont la topologie est maillée. Comme pour les autres topologies, tous les flux acceptés obtiennent dans notre modèle le débit demandé. Par contre, avec le modèle originel, seuls les flux dont les nœuds sources sont situés à un saut près du portail d'accès (1, 4) ont un débit proche de 300 kbits/s, les autres sont en famine. Sur l'ensemble des topologies réseau, lorsque le model originel est utilisé, seuls les flux dont la source est située à un saut du portail possèdent un faible taux de perte de paquets, ce phénomène est le problème de

famine des nœuds éloignés, problème déjà évoqué dans de nombreux articles (Lee, Yoon et Yeom 2010).

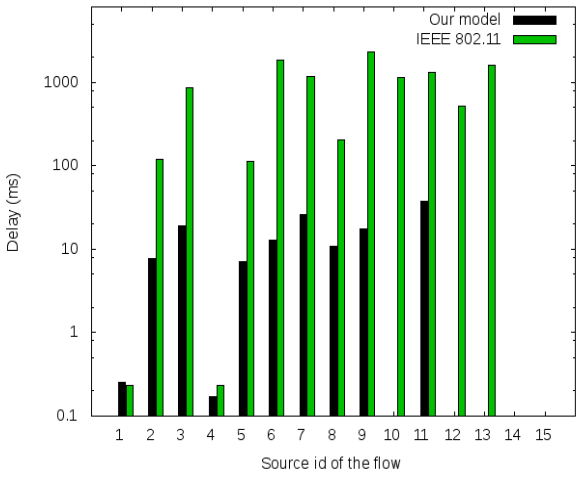


Figure 43 : Délai des flux dans le réseau mesh à topologie maillée

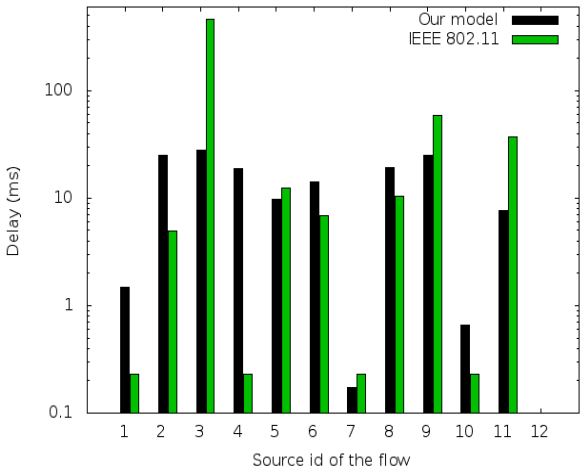


Figure 44 : Délai des flux dans le réseau mesh à topologie en croix

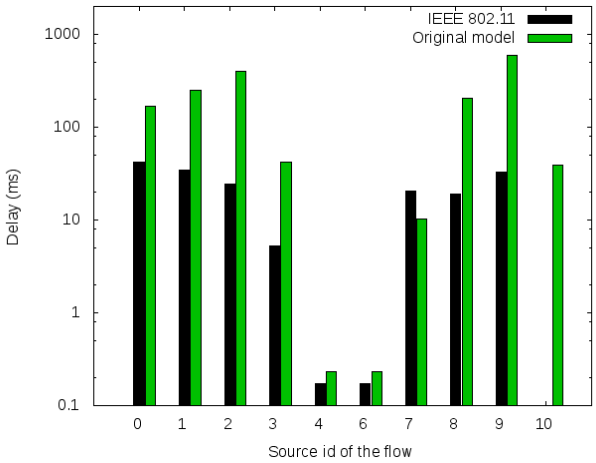


Figure 45 : Délais des flux dans un réseau mesh à topologie chainé

Les figures 43, 44, et 45 comparent le délai des flux entre notre modèle et le modèle originel dans, respectivement, le réseau mesh à topologie maillée, le réseau mesh à topologie en croix, le réseau mesh à topologie chainée. Le délai d'un flux est indiqué si au moins un paquet du flux a atteint le portail. Le délai du flux est le temps maximum nécessaire à un paquet du flux pour se rendre de la source au portail.

Notre modèle offre pour l'ensemble des flux admis dans le réseau des délais inférieurs au délai maximum requis pas les flux, qui est de 150ms. Le modèle originel, pour chacune des topologies, présente des délais bien supérieurs à notre modèle ; de nombreux flux possédant un délai supérieur

à 150ms. Dans notre modèle, comme dans le modèle originel les flux ont des délais d'autant plus petits que leur nœud source est proche du portail d'accès. Par exemple, sur la figure 43les flux 4 et 6 ont un faible délai car leur nœud source (4 et 6) sont proches du nœud portail d'accès qui porte l'identifiant 5.

Le simulateur ns2 repose sur un modèle d'interférence additif qui est le modèle d'interférence utilisé lors de la planification des liens de note CA, c'est pourquoi lors des simulations notre solution obtient presque aucune perte de paquets de données. Cependant, dans la réalité, le modèle d'interférence additif n'étant pas parfait comme le montre certaines expérimentations (Tan, Hu et Portmann 2012), notre modèle subirait un certain pourcentage de pertes de paquets. L'ensemble des flux avec le modèle originel sur l'ensemble de topologies, par contre admettent un taux de perte de paquets très importants, due entre autre :

- à des collisions sur le réseau, le contrôle d'accès CSMA/CA peut entrainer de nombreuses collisions
- à la surcharge du réseau ; les paquets peuvent être abandonnés par les nœuds suite à une surcharge de leur buffer.

D'après les résultats des évaluations, notre solution permet de respecter les contraintes de flux qu'elle admet sur le réseau en termes de débit et de délai des flux, contrairement au modèle originel où les exigences de la majorité des flux ne sont pas satisfaites. De plus, elle permet grâce à la planification des flux d'augmenter, par rapport au modèle originel, la capacité globale du réseau car elle évite les collisions ainsi et la perte de temps induite par les temps et paquets de contrôles liés au accès au support à compétition.

4.6. Conclusion

Dans ce chapitre, nous avons présenté un nouveau contrôle d'admission ayant pour objectif de respecter les exigences des flux en termes de délai et de débit ainsi que d'accroitre la capacité globale du réseau.

L'idée de notre solution est d'intégrer au sein d'un contrôle d'admission, un système de planification de liens afin d'accroitre la bande passante du réseau, de contrôler avec précision le débit que chaque nœud peut accorder à un flux et de maitriser le délai des flux. Le problème d'admission d'un nouveau flux est présenté sous forme d'un problème de programmation linéaire à variables binaires. Nous

démontrons ensuite que ce problème est NP complet car il est NP et réductible en temps polynomial en un autre problème NP-complet, le problème de la k-coloration d'un graphe précolorié. Afin de résoudre le problème d'admission d'un flux dans un réseau mesh avec planification des liens, nous proposons un algorithme itératif de planification des liens qui est une version modifiée de l'algorithme de séparation évaluation de Dakin. L'algorithme est intégré dans notre contrôle d'admission où chaque portail, lorsqu'il reçoit une demande d'admission d'un flux, vérifie s'il peut accepter le flux. S'il peut, il envoie alors à l'ensemble des nœuds de la route du flux la planification de ce dernier.

Les performances de notre solution ont été comparées, sur le simulateur ns2 à un réseau mesh classique basé sur le standard IEEE 802.11 avec routage réactif. Les résultats montrent que notre solution améliore les performances d'un réseau mesh en termes de débit, délai et perte de paquets. De plus, ils prouvent que notre solution atteint ces objectifs car, sur l'ensemble des simulations, les exigences en termes de délai et de débit des flux admis sont respectées et la capacité totale utile du réseau est améliorée.

Partie 1. Conclusion

Les réseaux mesh sont actuellement confrontés à des problèmes de qualité de service induit principalement par la faible capacité du réseau, la perte de paquets et le non respect régulier des contraintes de flux en termes de délai et de débit.

Cette partie, dans un premier temps, présente un état de l'art sur deux solutions permettant de résoudre les problèmes actuels des réseaux mesh ; le contrôle d'admission et la planification de liens. Tandis que le contrôle d'admission permet de respecter les contraintes de flux qu'il admet sur le réseau, la planification des liens, en assignant des slots d'émission à chaque lien du réseau, permet d'augmenter la capacité utile du réseau et d'éviter la perte de paquets. Cependant, les contrôles d'admission existants généralement surestiment ou sous-estiment la capacité du réseau entrainant soit des congestions ou une perte des ressources disponibles; de plus, ils sont limités dans le nombre de flux qu'ils peuvent admettre sur le réseau. Quant aux systèmes de planification, ils fixent la planification du réseau lors de son déploiement et ne gèrent pas la dynamique du réseau.

Dans un second temps, cette partie introduit notre système de contrôle d'admission avec planification des liens. Après avoir proposé une nouvelle méthode de calcul du délai des flux, posé le problème d'admission d'un flux comme un problème de programmation linéaire à variables binaires et prouvé qu'il était NP-complet, nous proposons un algorithme d'admission et de planification d'un flux. Cet algorithme lance itérativement un algorithme de séparation évaluation de Dakin modifié ; la méthode de séparation évaluation de Dakin permet de résoudre les problèmes de programmation linéaire en variables binaires, nous l'avons modifié afin qu'elle renvoie à chaque itération, si une nouvelle solution existe, une planification pour le flux dont le délai est optimum. Cet algorithme est introduit dans notre contrôle d'admission ; lorsqu'un nœud envoie une demande d'admission d'un flux, la destination utilise l'algorithme afin de décider si le flux est accepté ou non et si il l'est, de déterminer sa planification. Un message de retour est ensuite envoyé à la source précisant la planification du flux, si ce dernier est accepté.

Notre solution a été évaluée sur ns2 et comparée à un modèle classique basé sur un accès à compétition au réseau et sur le protocole AODV. Les résultats montrent que notre système atteint ses objectifs puisqu'il permet d'éviter les collisions, de respecter les contraintes des flux admis dans le réseau et d'augmenter la capacité utile du réseau.

Partie 2. Nouveau système de confiance avec détection des mauvais nœuds dans un réseau mesh sans fil

Partie 2. Introduction

Les mécanismes de QoS dans les réseaux mesh, assurent une utilisation fluide des services réseaux pour les utilisateurs. Cependant, ils permettent de garantir la qualité d'un réseau mesh uniquement lorsque le réseau ne possède pas de nœuds malveillants, égoïstes ou défaillants. Un nœud égoïste est un nœud qui cherche à maximiser son gain personnel tandis qu'un nœud malveillant a pour but de dégrader les performances du réseau (Nandiraju, et al. 2007). Par la suite, dans ce chapitre tout nœud ne respectant pas les règles du réseau, qu'il soit défaillant, malveillant ou égoïste est appelé un mauvais nœud. Au contraire, un nœud non mauvais est un bon nœud, c.à.d. honnête et non défaillant. De Les caractéristiques inhérentes (routage ad-hoc multi-sauts, liaison sans fil, manque de protection physique des routeurs) des réseaux mesh rendent ces réseaux particulièrement vulnérables aux attaques.

Pour assurer la qualité d'un réseau mesh sans fil, il est donc indispensable d'y intégrer un mécanisme de QoS, et une solution de sécurité. La majorité des solutions de sécurités ont basées sur de la cryptographie, or cette dernière permet de détecter principalement les nœuds externes au réseau, or un nœud mauvais peut également appartenir au réseau. Les systèmes de confiance permettent de pallier aux limites des solutions cryptographiques. En attribuant une valeur de confiance devant refléter le comportement des nœuds, ces systèmes détectent aussi bien les mauvais nœuds externes qu'interne au réseau. Le premier chapitre de cette partie dresse un état de l'art de ces systèmes. Chaque nœud dans un réseau mesh intégrant un mécanisme de confiance surveille ses voisins pour leurs attribuer une confiance reflétant leur comportement. Or, cette surveillance peut être tronquée selon l'état des liens entre le nœud et son voisin ; en effet, si ce lien subit beaucoup de pertes alors, le nœud peut penser que son voisin est malveillant alors qu'il ne l'est pas. Les solutions existantes considèrent généralement qu'un seul et unique mode de mauvaises actions, alors qu'il peut en exister plusieurs et n'envisagent pas les problèmes de bruit sur les liens du réseau mesh.

C'est pourquoi nous proposons dans le cadre de cette thèse, un nouveau système de confiance présenté dans le chapitre 6 de cette partie. L'originalité de notre système réside dans l'utilisation d'un système de surveillance composé de trois modules de défense, chacun surveillant un type de mauvais comportement. De plus, chaque module considère la perte de paquets sur les liens afin d'assigner une confiance à ses voisins. Chaque module utilise une méthode statistique afin de comparer la perte de paquets « normale » sur un lien avec la perte de paquets observée sur ce lien et démasquer les nœuds mauvais qui abandonnent des paquets. Le dernier chapitre de cette partie valide notre système via des évaluations de chacun de nos modules.

Chapitre 5. Les modèles de confiance

5.1. Introduction

La nature sans fil du réseau mesh facilite l'écoute des messages du réseau par des nœuds malveillants ; en effet, il suffit qu'un nœud malveillant soit dans la zone de portée d'écoute du réseau pour entendre l'ensemble des communications qui passent par cette zone. De plus, la nature sans fil du canal permet à un attaquant de brouiller l'ensemble des communications du réseau mesh sans fil en émettant un signal à forte puissance sur l'ensemble des fréquences utilisées par le réseau mesh (Siddiqui et Seon 2007).

Les réseaux mesh sont basés sur un routage ad-hoc multi-sauts où les messages sont transférés de nœud en nœud afin d'atteindre leur destinataire. Le bon fonctionnement d'un réseau mesh nécessite une collaboration entre ses nœuds ; si un seul nœud ne fait pas suivre les messages toutes les communications passant par ce nœud échouent. Le routage ad-hoc multi-sauts rend le réseau particulièrement vulnérable puisqu'il suffit qu'un seul nœud soit malveillant et ne collabore pas au routage ou pire fausse le routage pour que tout le réseau soit endommagé (Iyer, Rosenberg et Karnik 2009) (Siddiqui et Seon 2007).De plus, capturer un nœud pour un attaquant peut souvent se révéler être une tâche particulièrement aisée, car les routeurs mesh sont généralement peu protégés (Iyer, Rosenberg et Karnik 2009).

Les routeurs mesh sont souvent des équipements à bas prix, installés dans des lieux publics et facilement accessibles ; ils peuvent être placés sur des lampadaires, des poteaux de signalisation (Iyer, Rosenberg et Karnik 2009), etc. Un routeur mesh, peut ainsi être facilement capturé par un attaquant et modifié ou remplacé. En capturant un routeur mesh, l'attaquant peut accéder aux clés d'un nœud et ainsi décrypter des messages. En modifiant ou en remplaçant le routeur, l'attaquant peut aussi contrôler les communications qui sont transférés par ce dernier, envoyer de faux messages, perturber le routage, etc (Iyer, Rosenberg et Karnik 2009).

Ainsi, les réseaux mesh, de part leurs caractéristiques (routage ad-hoc multi-sauts, nature sans fil du canal, manque de protection physique des routeurs) sont des réseaux particulièrement vulnérables (Iyer, Rosenberg et Karnik 2009). De nombreuses solutions de sécurité ont été proposées pour les réseaux mesh, la plupart repose sur des systèmes cryptographiques (Iyer, Rosenberg et Karnik 2009) (Kim et Tsudik 2009). L'accès au réseau d'un nœud n'est alors possible que si ce dernier possède le

matériel cryptographique adéquat. L'intégration d'un système cryptographique au réseau permet de garantir la confidentialité, la non répudiation, l'intégrité et l'authentification des messages tant que le système cryptographique est sécurisé. Cependant, les solutions basées sur la cryptographie ne protègent pas le réseau mesh d'un nœud malveillant autorisé à accéder au réseau et donc possédant le matériel cryptographique adéquat. Ainsi, un système cryptographique prévient des menaces externes au réseau mais non internes. Les mécanismes de confiance permettent de pallier aux limites des solutions basées sur un système cryptographique (Yu, et al. 2010) (Ning et Sun 2005), car ils peuvent détecter les mauvais nœuds appartenant au réseau et par exemple les isoler du réseau ou les inciter à adopter un bon comportement.

Le chapitre 5 présente un état de l'art sur les modèles de confiance dans les réseaux mesh, il liste leurs objectifs, décrit leurs structures, et introduit des modèles de confiance existants et leurs limites.

5.2. Objectifs et fonctionnement des modèles de confiance dans les réseaux mesh sans fil

Dans un réseau mesh implémentant un système de confiance, un nœud assigne à d'autres nœuds du réseau un niveau de confiance à partir d'observations obtenues via des échanges directes ou indirects qu'il a eu avec ce dernier. Le niveau de confiance qu'un nœud assigne à un autre nœud, détermine comment il interagit avec ce dernier ou avec les paquets de ce dernier.

La confiance qu'un nœud A a dans un nœud B est l'espérance subjective du nœud A que l'interaction avec le nœud B soit positive (Yu, et al. 2010). Les objectifs d'un système de confiance sont multiples:

- Détecter les nœuds qui agissent mal. Le niveau de confiance accordé à un nœud reflète les actions passées de ce dernier. Plus la confiance accordée à un nœud est haute et plus ce dernier est considéré comme non mauvais (honnête et non défaillant) et plus elle est basse et plus le nœud est considéré comme mauvais (défaillant, égoïste, malveillant). En général, le niveau de confiance qu'un nœud accorde à un autre nœud prend une valeur entre 0 et 1, plus cette valeur est proche de 1 plus la confiance est élevée, plus elle est proche de 0 et plus la confiance est basse.
- Isoler les mauvais nœuds. Plus la confiance en un nœud est faible et moins le nœud interagira avec lui, il ne l'inclura plus sur les routes des flux et ne fera plus suivre ses

données. Un mauvais nœud peut donc être isolé du réseau afin de limiter son impact négatif sur le réseau.

- Inciter les mauvais nœuds à bien se comporter. Afin d'inciter un mauvais nœud à agir honnêtement, les nœuds du réseau peuvent adopter un comportement où tout nœud du réseau a plus intérêt à se comporter de manière honnête que mauvaise. Les mécanismes d'incitation indut dans un système de confiance sont souvent basés sur la théorie des jeux et supposent un comportement rationnel des mauvais nœuds (Wang, et al. 2009).
- Remplacer ou réparer les mauvais nœuds. La détection d'un mauvais nœud peut permettre son remplacement par l'administrateur du réseau.

Dans un système de confiance, chaque nœud collecte des informations de première main et parfois également de seconde main sur les autres nœuds du réseau afin de calculer le niveau de confiance qu'il leur accorde. Les informations de première main que possède un nœud sur l'un de ses voisins proviennent d'interactions directes avec ce voisin. Si les informations directes qu'un nœud récolte se révèlent non suffisantes alors le nœud peut utiliser des informations indirectes dites de seconde main en récoltant des informations sur ses voisins ou des nœuds plus éloignés auprès d'autres nœuds du réseau. Selon le niveau de confiance qu'un nœud a en un autre nœud, le nœud décide de la façon dont il interagit avec lui ou traite ses paquets. Certains systèmes de confiance établissent également des systèmes de dissémination d'alertes pour prévenir qu'un nœud est mauvais (Buchegger et Le Boudec 2002) (Hasswa, Zulkernine et Hassanein 2005) . La figure 46 présente l'architecture du système de confiance intégré à un nœud.

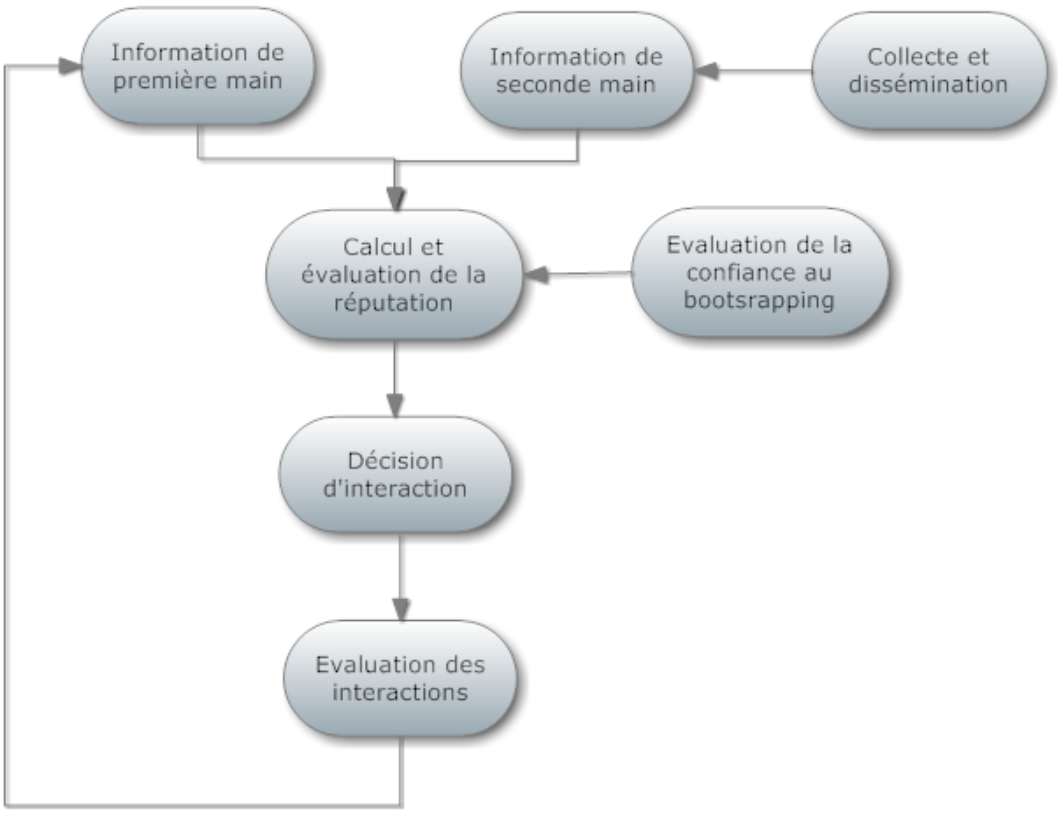


Figure 46 : Architecture de base d'un nœud dans un système de réputation

La réalisation d'un système de confiance nécessite que chaque nœud possède (voir figure 46) :

- un module de *bootstrapping* . Ce module permet au nœud d'assigner un niveau de confiance à un nœud nouvellement arrivé sur le réseau.
- un module de surveillance et d'évaluation des interactions. Ce module permet au nœud de collecter des informations de première main.
- un module de collecte et de dissémination des informations de seconde main, si nécessaire.
- un module de calcul de la confiance. Ce module permet à un nœud de calculer la confiance qu'il a en un autre nœud à partir des informations de première et de seconde main qu'il possède sur ce dernier.
- un module de décision des interactions. Ce module permet au nœud de déterminer, selon le niveau de confiance qu'il accorde à un nœud, les futures interactions qu'il aura avec ce nœud et la manière dont il traitera les paquets de ce nœud.

Il existe de nombreuses solutions de système de confiance qui diffèrent principalement dans leur implémentation de ces différents modules.

5.3. Modèles de confiance existants dans les réseaux mesh

Les modèles de confiance dans les MANETs peuvent facilement être implémentés dans un cœur de réseau mesh car, ce dernier, peut être considéré comme un MANET où l'ensemble des entités qui le compose sont fixes. Ainsi, cette section présente aussi bien des modèles de confiances existants pour les réseaux mesh que pour les MANETs. De nombreux systèmes de confiance utilisent comme méthode de détection des mauvais nœuds, le système de détection d'intrusion (IDS) Watchog (Marti, et al. 2000). Watchdog est un IDS qui permet de détecter les *greenholes* (Jhaveri 2013)c.à.d. les nœuds qui font suivre seulement certains paquets, les *blackholes* (Jhaveri 2013) les nœuds qui font suivre aucun paquet et les nœuds qui modifient les paquets. Ce système de détection d'intrusion permet à un nœud de détecter si un voisin est mauvais en écoutant si ce dernier fait suivre correctement ses messages. Chaque nœud implémentant Watchdog enregistre chaque paquet de données qu'il émet, et vérifie ensuite si le nœud suivant sur la route du paquet (si ce dernier n'est pas la destination), retransmet correctement le message. Ainsi, le nœud écoute tous les paquets que son voisin émet en mode *promiscuous*. Le mode *promiscuous* est un mode de configuration d'une carte réseau sans fil qui lui permet d'accepter tous les paquets qu'elle reçoit, même si ceux-ci ne lui sont pas adressés. Le nœud compare chaque paquet envoyé par son voisin avec l'ensemble des paquets qu'il possède dans son buffer et qu'il lui a envoyé pour faire suivre. Si un paquet dans son buffer correspond à celui que vient d'envoyer son voisin alors le nœud efface le paquet de son buffer et estime que son voisin a fait suivre correctement son paquet. Si au bout d'un certain laps de temps, il n'a toujours pas entendu son voisin faire suivre le paquet qu'il lui a envoyé, il efface le paquet de son buffer et incrémente le compteur d'échec de son voisin. Si le compteur d'échec d'un voisin est supérieur à un certain seuil, le nœud considère que ce dernier ne fait pas suivre intentionnellement les paquets et est mauvais. Ainsi, Watchdog permet aux nœuds de surveiller si leurs voisins font suivre les messages qu'ils leur envoient et s'ils ne les modifient pas. La figure 47 présente une illustration du fonctionnement de *Watchdog* où un nœud A souhaitant envoyer un paquet à un nœud C l'envoie au nœud B pour que ce dernier le transmette à C. Via *Watchdog* le nœud A vérifie que le nœud B transmette correctement son paquet à C en écoutant les messages envoyés par B.

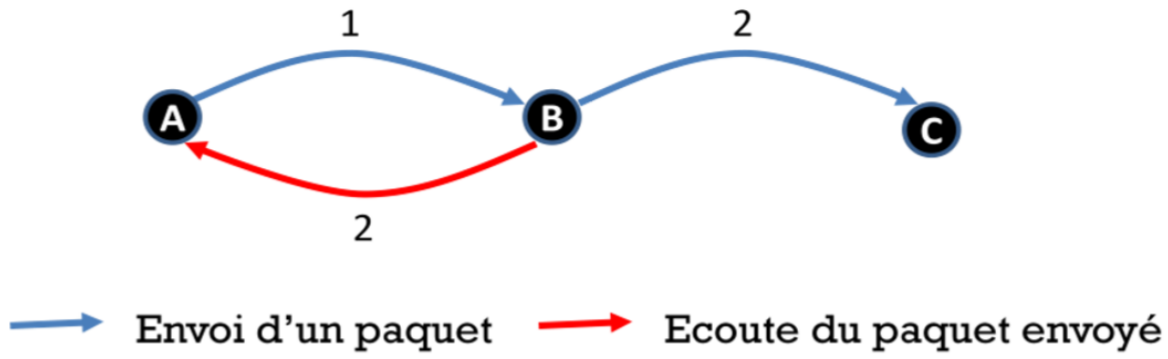


Figure 47 : Illustration de *Watchdog* : A envoie un paquet à B et vérifie que B le fait suivre à C en écoutant les paquets que B envoie.

Dans (Sen 2010), Sen *et al*. proposent un modèle de confiance distribué pour les MANETs, dont le module de collecte et de dissémination des informations de seconde main est à la fois à la demande et proactif. Dans leur solution, chaque nœud enregistre pour chacun de ses voisins, le nombre de paquets qu'il lui envoie et le nombre de paquets que ce dernier fait suivre correctement via *Watchdog*. Des informations de seconde main sont également collectées et disséminées aussi bien pro-activement ou à la demande. La méthode de dissémination proactive est déclenchée lorsqu'un nœud détecte que l'un des voisins est malveillant, il broadcaste alors la confiance qu'il a en ce nœud à l'ensemble de ses voisins. La méthode de collecte des informations à la demande est déclenchée lorsqu'un nœud souhaite calculer la confiance qu'il a dans un autre nœud, il effectue alors une requête auprès de ses voisins pour connaitre la confiance qu'ils accordent en ce nœud. Pour calculer la confiance qu'il a dans un nœud, un nœud considère à la fois l'historique du nœud et les informations de seconde main qu'il pondère. A l'arrivée d'un nouveau nœud sur le réseau, sa confiance est maximale et est fixée à 1 ; la confiance pouvant fluctuer entre 0 et 1. La réputation qu'un nœud A possède en un nœud B, $r(A,B)$ dépend du nombre de paquetsque le nœud B a retransmis après les avoir reçus de A sur le nombre de paquets que A à envoyé à B, elle est périodiquement calculée ainsi :

$$r(A,B) = \frac{Nb\ paquets\ transférés\ par\ B}{Nb\ de\ paquets\ envoyés\ à\ B} \qquad \text{Équation 64}$$

Chaque réputation est ensuite combinée avec la précédente afin de considérer lors du calcul de la confiance l'historique des interactions. Ainsi, en notant l'ancienne confiance que A a en B $r_{old}\,(A,B)$ et la confiance actuelle que A a en B $r_{current}\ \ (A,B)$alors la réputation que A possède de B est :

$$r(A,B) = (1-\alpha) * r_{old}\,(A,B) + \alpha * r_{current}\ \ (A,B) \qquad \text{Équation 65}$$

avec α une valeur entre 0 et 1. Lorsqu'après avoir envoyé une demande d'information sur un nœud B, un nœud A reçoit en provenance de P nœuds voisins, $N_1, N_2, \ldots N_i \ldots N_p$, la confiance qu'ils ont en B notée $r(N_i, B)$, le nœud A recalcule la confiance qu'il a en B en pondérant la confiance reçue par un nœud N_i par la confiance qu'il possède en ce nœud notée $r(A, N_i)$, ainsi :

$$r(A,B) = \frac{\alpha * r(A,B)}{\alpha + \sum_{i=1}^{P} r(A,N_i)} + \frac{\sum_{i=1}^{P} r(A,N_i) * r(N_i,B)}{\alpha + \sum_{i=1}^{P} r(A,N_i)} \qquad \text{Équation 66}$$

Les résultats de simulation de la solution montrent que le niveau de confiance d'un nœud est d'autant plus bas que ce dernier est mauvais. Cependant, cette solution ne propose aucun mécanisme pour isoler un nœud mauvais ou l'inciter à bien se comporter. De plus, un nœud peut faire suivre correctement les messages mais envoyer de fausses informations de réputation et cette solution ne gère pas une telle situation. L'envoi de fausses informations de confiance, si ces fausses informations ont pour but de diminuer la confiance des nœuds, s'appelle le *badmouthing*, si elles ont pour but d'augmenter la confiance d'un nœud, s'appelle le *ballot stuffing* (Yu, et al. 2010).

Dans (Y. Li 2011), les auteurs présentent un système de confiance dans un réseau mesh au routage multi-chemins où les mauvais nœuds situés prés des portails d'accès à l'internet sont plus sévèrement punis que les autres. En effet, le mauvais comportement d'un nœud situé prés d'un portail a plus de conséquence que le mauvais comportement d'un nœud situé loin du portail car ce dernier est traversé par beaucoup moins de flux. La collecte des informations de première main est effectuée par une extension de *Watchdog*. Cette extension permet à un nœud d'enregistrer à la fois le nombre de paquets :

- que son nœud voisin retransmet et qu'il lui a envoyé (figure 27.a), comme dans *Watchdog*.
- que son nœud voisin a reçu d'un nœud tiers ainsi que le nombre de paquet en provenance du nœud tiers que son voisin a retransmis (figure 27.b).

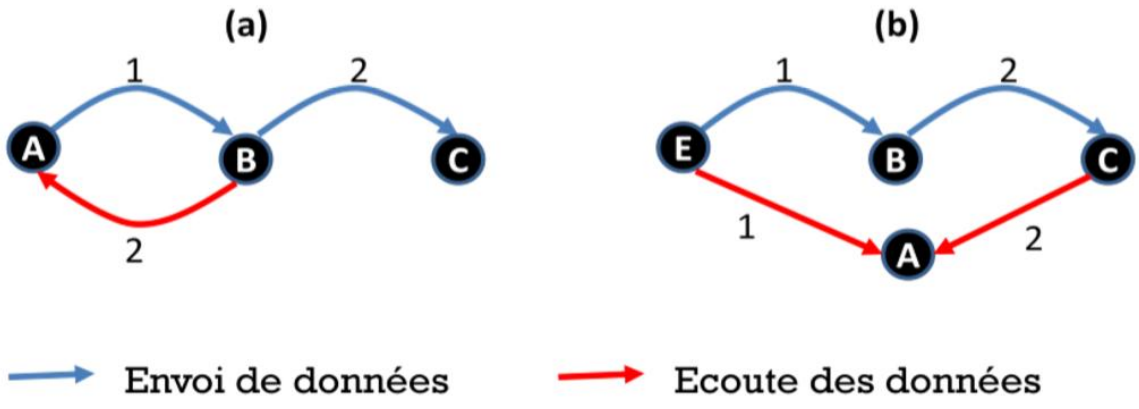


Figure 48 : Extension de *Watchdog*

(a) Le nœud A collecte le nombre de paquets que B retransmet à C

(b) Le noeud A enregistre le nombre de paquets que E envoie a B et le nombre que B renvoie à C

Chaque nœud arrivant sur le réseau a un niveau de confiance égal à la réciproque de son délai ; le délai d'un nœud étant le temps moyen nécessaire pour que le nœud retransmette un paquet qu'il a reçu. Le délai d'un nœud A est noté D_A. Selon le type d'information qu'un nœud récolte sur un autre voisin, il ne calcule pas la confiance qu'il a en ce dernier de la même manière. Si un nœud A calcule la réputation de l'un de ses voisins B à partir du nombre de paquets qu'un nœud tierce envoie à son voisin B et le nombre de paquet que ce voisin B retransmet, alors il utilise la formule suivante :

$$r(A,B) = \frac{1}{D_B} - K(1 - \frac{nb\ de\ paquets\ que\ B\ fait\ suivre}{nb\ de\ paquets\ envoyés\ par\ un\ tierset\ reçus\ par\ B}) \quad \text{Équation 67}$$

avec D_B le délaiau niveau du nœud B et K une constante dont la valeur doit être obtenue par expérience. Si un nœud A calcule la réputation de l'un de ses voisins B à qui il transfère des données sur une route, il connait alors le nombre de sauts de la route notée P_{hop}, la position du nœud sur cette route notée L_B. Le nœud A calcule alors la réputation du nœud B selon la formule suivante :

$$r(A,B) = \frac{1}{D_B} - K(1 + P_{hop} - L_B)(1 - \frac{nb\ de\ paquets\ que\ B\ fait\ suivre}{nb\ de\ paquets\ reçus\ par\ B\ envoté\ par\ A}) \quad \text{Équation 68}$$

Cette dernière formule associe plus rapidement une mauvaise confiance à un nœud vers la fin de la route et donc prés d'un portail à l'internet qu'à un nœud au début de la route. Lorsqu'un nœud possède une confiance en l'un de ses voisins, inférieure à un certain seuil, alors le nœud diffuse une

alerte dans tout le réseau, pour qu'aucun nœud ne fasse suivre de paquets dont le mauvais nœud est la source. Le but de cette punition est de motiver le nœud à agir honnêtement en l'empêchant de pouvoir émettre des données sur le réseau, mais en lui permettant de pouvoir faire suivre des données pour améliorer son niveau de confiance. Le problème de ce système de confiance est qu'il est vulnérable au *badmouthing* , un nœud peut donc envoyer des fausses alertes sur des nœuds honnêtes qui vont alors être injustement punis.

Dans (Safaei, Sabaei et Torgheh 2010), les auteurs présentent un système de confiance qui renforce la coopération entre les nœuds et intègre un système de priorité des paquets selon la réputation des nœuds sources. Chaque nœud utilise *Watchdog* afin de calculer la confiance qu'il a en ses voisins, cette dernière équivaut au nombre de paquets envoyés par le nœud sur le nombre de paquets qu'il a reçu. La diffusion des informations de confiance s'effectue lors de l'envoi des paquets de RREQ. Lorsqu'une source souhaite envoyer un paquet de données, elle diffuse une requête de route.

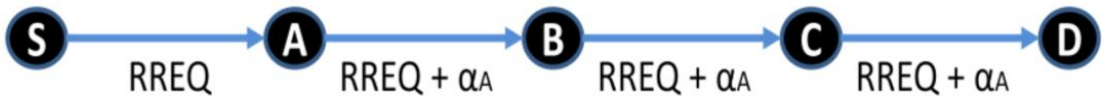


Figure 49 : S envoie une requête de route pour établir une route jusqu'au nœud D. Le nœud voisin A ajoute au RREQ la confiance α_A qu'il a en S. Chaque noeud intermédiaire non voisin de la source récupère la valeur de confiance que A a en la source S lorsqu'il reçoit la requête de route.

Chaque nœud voisin de la source, recevant la requête, ajoute au RREQ la confiance α qu'il a en la source avant de faire suivre le paquet (voir figure 49). Chaque nœud non voisin de la source et recevant ensuite le RREQ calcule la confiance qu'il a en la source du paquet selon la valeur de confiance α inscrite dans la requête (voir figure 49), mais également selon les valeurs de confiance qu'il a préalablement obtenu à propos de cette source via la réception de précédents RREQs. Ainsi, si le nœud a reçu n RREQs en provenance d'une même source dont il n'est pas le voisin, chacune possédant une valeur de confiance α_i pour cette source, alors il calcule la confiance qu'il a dans cette source avec la formule suivante :

Équation 69

$$\alpha = \frac{\sum_{i=1}^{n} \alpha_i * w_i}{n} \; avec \; w_i = \frac{s - c_i}{s} \; et \; s = \sum_{i=1}^{n} c_i$$

avec c_i le nombre de sauts effectués par chaque RREQ avant d'arriver au nœud. D'après la formule précédente, plus un paquet a traversé de nombreux nœuds plus le poids de la valeur de confiance contenue dans le paquet est faible. En effet, plus un paquet a effectué de sauts et plus la probabilité

qu'un nœud malveillant ait changé la valeur de confiance contenue dans le paquet est importante et plus le crédit accordé à la valeur de confiance contenue dans ce paquet est faible. La priorité d'un paquet au niveau d'un nœud dépend de la confiance que le nœud a en la source du paquet. Un nœud, selon le système de priorité des paquets, envoie un paquet, dont la source a un niveau de confiance noté α que si l'ensemble des paquets dont la source a un niveau de confiance supérieure à ce dernier ont déjà été envoyés. Les résultats de la simulation de la solution sur le simulateur GLOMOSIM (Xiang, Rajive et Mario 1998) montrent que cette approche permet de réduire le nombre de paquets détruits par rapport au système de confiance CONFIDANT (Buchegger et Le Boudec 2002). Cependant comme les nœuds ne pondèrent pas les informations de seconde main qu'ils reçoivent, cette solution est sensible au *bad mouthing* et au *ballot stuffing*.

Dans (Wang, et al. 2009), les auteurs proposent un système de confiance qui incite les nœuds à la coopération dans un MANET en utilisant la théorie des jeux et qui considère le problème de bruit sur le canal et donc de perte de paquets au niveau des liens. Dans ce réseau, les auteurs supposent que chaque nœud possède une certaine somme et qu'il perd un montant c à chaque fois qu'il émet un paquet dont il est la source ou un simple intermédiaire, et gagne un montant $g > c$ lorsque l'un de ses paquets est reçu par la destination. Lorsqu'un nœud détecte que son voisin ne fait pas suivre tous les paquets qu'il lui envoie, il l'isole et le punit pendant un certain nombre de slots. Lorsqu'il est isolé, un nœud ne peut pas émettre ni faire suivre de données, son gain, pendant une période d'isolement est nul. Lorsqu'il est puni, un nœud peut faire suivre des données mais ne peut pas en émettre. Afin d'inciter les nœuds à agir correctement estiment que le gain de la coopération doit être supérieur au gain de la période de mauvaises actions, d'isolement et de punition. Ainsi le nombre de slots$i$ et p pendant lesquels un mauvais nœud doit être respectivement isolé et puni doit être tel que :

$$\sum_{k=t}^{t=p+r+1} u(C) > u(M) + p * u(I) + i * u(P) \qquad \text{Équation 70}$$

où $u(C)$ est le gain d'un nœud lorsqu'il coopère pendant un slot, u (M) le gain d'un nœud lorsqu'il est mauvais pendant un slot, $u(I)$ le gain d'un nœud lorsqu'il est en isolement pendant un slot, $u(P)$ le gain d'un nœud lorsqu'il est puni pendant un slot. Si le nombre de slots d'isolement i et le nombre de slots de punition p sont fixés afin de vérifier la validité de l'équation 70, les nœuds du réseau ont alors plus intérêt à coopérer qu'à mal agir afin de maximiser leur gain. Les résultats de simulation de la solution montrent que le système incite les nœuds à agir honnêtement si ces derniers sont rationnels. Les auteurs expliquent comment ils intègrent la probabilité de perte de paquets au niveau d'un lien lorsque son gain est calculé mais non lors de la détection d'un nœud mauvais. Or, la perte

de paquets sur un lien a une incidence directe sur les informations collectées par *Watchdog* ; un nœud peut ne pas entendre l'un de ses voisins retransmettre son paquet à cause du bruit sur le canal et peut donc classer injustement son voisin comme mauvais. Il serait donc intéressant de considérer la perte de paquets sur un lien afin de détecter si un nœud est mauvais.

Dans (Li, Parker et Joshi 2012), les auteurs présentent un système basé sur une confiance multidimensionnelle et un algorithme de détection des nœuds malveillants. Dans leur solution, un nœud établit trois valeurs de confiance pour chacun des ses voisins :

- COLT (*Collaboration Trust*). COLT reflète le degré de collaboration d'un nœud pour faire suivre les données correctement. Cette valeur peut être utilisée pour déterminer si un nœud doit être inclus dans les activités de suivi de paquets.
- BET (Behavioral Trust). BET reflète le degré de comportement anormal d'un nœud tel que la modification de paquets, l'inondation du réseau avec des paquets RTS et le mauvais routage de paquets. BET est un indicateur de la malveillance d'un nœud.
- RET (Reference Trust). RET reflète la quantité d'opinions correctes qu'un nœud envoie. La valeur de RET peut être utilisée comme poids pour l'intégration des informations de seconde main d'un nœud.

L'ensemble des valeurs de confiance sont initialisées 1, ainsi, un nœud associe à un nouveau nœud une confiance maximale. L'algorithme de détection des nœuds malveillants permet d'obtenir le top k, des nœuds ayant la plus mauvaise réputation. A la fin de l'algorithme, chaque nœud possède le même top k de mauvais nœuds, c'est-à-dire que les k nœuds ayant la plus mauvaise réputation sont les mêmes pour l'ensemble des nœuds du réseau. Dans cet algorithme, chaque nœud envoie à l'ensemble de ses voisins sa vue locale, c'est-à-dire l'ensemble des valeurs de confiance qu'il possède. Chaque nœud recevant la vue locale de l'un de ses voisins met à jour les valeurs de confiance qu'il possède et diffuse à son tour sa vue locale. L'algorithme s'arrête lorsque tous les nœuds ont le même top k de mauvais noeuds. Pour mettre à jour la confiance qu'un nœud i a dans un nœud m, après avoir reçu la valeur de confiance que possédait son voisin j sur m, notée m_j, un nœud i utilise la formule suivante :

$$m_i = \frac{w_{ii} * m_i + w_{ij} * m_j}{w_{ii} + w_{ij}} \qquad \text{Équation 71}$$

avec w_{ij} le poids que donne le nœud i aux informations de j et w_{ii} le poids qu'il donne à ses propres informations. Le poids qu'un nœud accorde à ses propres informations est toujours égal à 1. Les

résultats de simulation de la solution montrent que leur système offre de bons résultats en termes de détection des mauvais nœuds et de faible surcharge du réseau. Cependant, les auteurs ne décrivent pas comment ils collectent l'ensemble des données telles que la quantité de fausses informations de seconde main, d'inondation du réseau et d'envoi de fausses indications de routage.

5.4. Conclusion sur l'état de l'art des systèmes de confiance dans les réseaux mesh

Les modèles de réputation existants présentent de nombreuses similarités, la plupart intègre un module de surveillance pour collecter les données, un module de calcul de la confiance des nœuds, une méthode d'interactions entre les nœuds, un module de *bootstraping*, un module de collecte des informations de seconde main et son corolaire de dissémination des informations de seconde main.

Les systèmes de confiance se différencient principalement dans l'implémentation de ces différents modules. La collecte des informations de première main s'effectue principalement avec l'IDS *Watchdog* (Marti, et al. 2000) ou des versions améliorées de *Watchdog* comme dans (Safaei, Sabaei et Torgheh 2010). Les méthodes de calcul de confiance sont plus ou moins complexes ; certaines considèrent l'historique des interactions (Sen 2010), d'autres le délai des nœuds (Y. Li 2011) ou la position du nœud source sur la route (Safaei, Sabaei et Torgheh 2010). Selon le niveau de confiance attribué à un nœud, les interactions avec ce dernier ou le traitement de ses paquets sont différents. Dans certaines solutions, le nœud est isolé et puni le temps minimum nécessaire pour qu'il coopère (Wang, et al. 2009), dans d'autres, chaque paquet possède une priorité au niveau d'un nœud selon la confiance que le nœud accorde à la source du paquet (Safaei, Sabaei et Torgheh 2010). Le module de *bootsraping* associe une confiance à un nœud lorsque ce dernier arrive sur ce réseau. Certaines solutions accordent à un nœud arrivant sur le réseau la confiance maximale (Li, Parker et Joshi 2012) (Sen 2010) tandis que d'autres peuvent lui accorder une confiance qui est l'inverse du délai au niveau du nœud (Y. Li 2011). Certaines solutions utilisent des informations de seconde main pour prendre des décisions ou calculer la confiance des nœuds. La collecte et la dissémination des informations de seconde main peut s'effectuer dans certaines solutions pro-activement ou à la demande (Sen 2010), ou via l'envoi de RREQs (Safaei, Sabaei et Torgheh 2010). Le tableau suivant compare les solutions présentées précédemment selon leur implémentation des différents modules d'un système de confiance.

Tableau 8 : Tableau de comparaison des solutions de système de confiance existantes

	(Wang, et al. 2009)	(Sen 2010)	(Safaei, Sabaei et Torgheh 2010)	(Y. Li 2011)	(Li, Parker et Joshi 2012)
Surveillance et collecte des informations de première main	Watchdog	Watchdog	Watchdog	Version améliorée de Watchdog	Non précisé
Calcul de la confiance	Pas de calcul de confiance	Considère l'historique et pondère les informations de seconde main	Considère le nombre de sauts effectués par le paquet apportant les informations de seconde main	Considère la position du nœud sur la route et le délai du nœud	Multiples confiances Pondère les informations de seconde main
Décision d'interactions	Isolement et punition afin d'inciter à la coopération	Non précisé	Priorité du paquet au niveau d'un nœud selon la confiance que le nœud accorde à la source du paquet	Non suivi des paquets issus d'un nœud dont la source a une confiance inférieur à un certain seuil	Non précisé
Bootstrapping	Non précisé	Confiance maximale	Non précisé	Confiance équivalent à l'inverse du délai du nœud	Confiance maximale
Collecte et dissémination des informations de seconde main	Pas d'information de seconde main	A la demande et ou pro-activement	Lors de l'envoi de RREQ	Pas d'informations de seconde main	Utilise un algorithme de diffusion permettant d'obtenir les K plus mauvais nœuds du réseau

Parmi les solutions présentées précédemment, seul l'article (Wang, et al. 2009) considère le problème de bruit sur le canal et la perte de paquets sur un lien. Cependant, il ne considère pas le bruit lors de la détection de la malveillance d'un nœud ou du calcul de la confiance d'un nœud. Or, le bruit sur un lien peut considérablement nuire aux informations collectées par *Watchdog* et doncdoit être considéré dans le calcul de la confiance d'un nœud afin de ne pas attribuer une confiance basse à un nœud honnête et non défaillant dont la perte de paquets sur ces liens est importante. En effet,

le bruit peut empêcher un nœud d'entendre si son voisin retransmet correctement les paquets qu'il lui a envoyés. Ainsi, un nœud peut considérer que l'un de ses voisins est mauvais parce qu'il ne peut pas entendre à cause du bruit que ce dernier retransmet correctement les paquets qu'il lui a demandé de faire suivre. Un système de confiance dans un réseau mesh basé sur *Watchdog* peut donc générer de nombreux faux positifs si le canal est fortement bruyant et donc le rejet de nœuds honnêtes du réseau. Dans le cadre de cette thèse, nous proposons un nouveau système de détection d'intrusion des mauvais nœuds dans un réseau mesh qui considère la perte de paquets sur les liens.

Conclusion

Ce chapitre présente un état de l'art sur les systèmes de confiance existants dans les réseaux mesh. Ces systèmes ont pour but de d'assigner à chaque nœud du réseau une valeur de confiance reflétant son comportement. La confiance attribuée à un nœud est basée sur les informations qui sont collectés sur ce nœud. Or, la majorité des systèmes reposent sur un unique module de surveillance *Watchdog* qui ne s'intéresse qu'à un seul type de mauvaise action, les mauvais nœuds utilisant d'autres modes d'actions ne sont donc pas repérés. De plus, *Watchdog* ne considère pas la perte de paquets sur les liens, or cette dernière biaise les résultats collectés par les nœuds et donc par la suite la valeur de confiance que les systèmes de confiance basés sur *Watchdog* attribuent aux nœuds.

Chapitre 6. Nouveau système de confiance de détection des mauvais nœuds

6.1. Introduction

Ce chapitre présente notre système de confiance dans un réseau mesh sans fil (Dromard, Khatoun et Khoukhi 2013). Notre solution attribue à chaque nœud une valeur de confiance reflétant son comportement (mauvais ou bon). Notre système a pour objectif de pallier aux limites des solutions de confiance existantes qui considèrent généralement un seul type de mauvais comportement et qui ne prennent pas en compte le bruit lors de la détection des mauvais nœuds. Notre système de confiance intègre un système de surveillance comprenant trois modules, chacun détectant un type de mauvais comportement et intégrant une méthode statistique afin de déterminer si les données observées proviennent d'un nœud mauvais ou d'un nœud qui subit une perte sur son réseau. Ce système de surveillance permet de différencier les mauvais des bons nœuds sur un réseau subissant une forte perte de paquets. Le chapitre se termine sur l'évaluation de notre solution qui montre qu'elle peut détecter trois types d'attaques différentes en présence de bruit sur le réseau, avec un faible taux de faux négatifs et de faux positifs.

6.2. Objectifs

La majorité des systèmes existants utilisent sur l'IDS *Watchdog* afin de collecter des données de première main. Cet outil, intégré à un nœud permet à ce dernier de vérifier si ses voisins font suivre correctement les paquets qu'il lui envoie en écoutant ses transmissions. Si un nœud n'entend pas un voisin faire suivre correctement ses données, alors le nœud considère que son voisin l'a probablement abandonné. Via *Watchdog*, un nœud enregistre à la fois pour chaque voisin :

- le nombre de paquets qu'il lui a transmis et que ce voisin doit faire suivre
- le nombre de paquets qu'il lui a transmis et que ce voisin a correctement fait suivre.

Or, en utilisant *Watchdog*, un nœud peut considérer injustement que son voisin ne fait pas suivre un paquet correctement si :

- le voisin n'a jamais entendu et reçu le paquet à cause du bruit sur le canal. On considère alors que le paquet a été perdu sur le lien allant du nœud à son voisin (voir figure 51).
- lorsqu'il écoute en mode *promiscuous* les transmissions de son voisin, il n'entend pas ce dernier faire suivre son paquet parce qu'il y a du bruit sur le réseau. On considère alors que le paquet a été perdu sur le lien allant du voisin au nœud (voir figure 50).

Les figures ci-dessous représentent les deux situations possibles qui peuvent mener un nœud à penser injustement que son voisin ne fait pas suivre intentionnellement (ou à cause d'une défaillance) ses paquets de données. Sur la figure 50, un nœud A pense que son voisin B ne fait pas suivre intentionnellement (ou à cause d'une défaillance) son paquet car ce dernier est perdu lors de sa transmission sur le lien (A,B). Sur la figure 51, le nœud A pense que son voisin B n'a pas fait suivre correctement son paquet au nœud C car A écoute en mode *promiscuous* les paquets envoyés par B et que le paquet a été perdu sur le lien (B,A).

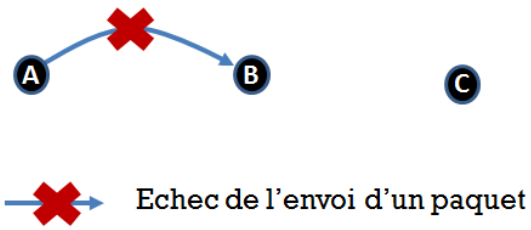


Figure 50 : Perte de paquet sur le lien (A,B)

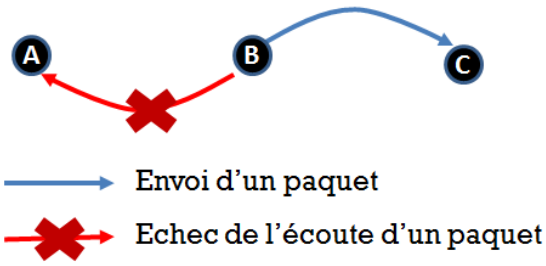


Figure 51 : Perte de paquet sur le lien (B,A)

Les nœuds utilisant l'IDS *Watchdog* peuvent injustement détecter leurs voisins comme mauvais. En effet, lorsqu'un nœud n'entend pas un voisin transmettre un paquet qu'il lui a envoyé, il en déduit que ce dernier a probablement abandonné le paquet alors qu'il peut en réalité être victime de perte de paquets sur l'un de ses liens. L'IDS *Watchdog* peut donc déclencher de nombreux faux positifs (nœuds détectés comme mauvais alors qu'ils ne le sont pas) d'autant plus que la perte de paquets sur les liens du réseau mesh est importante. A cause des faux positifs déclenchés par *Watchdog*, un système de confiance basé sur cet IDS peut accorder à des nœuds honnêtes et non défaillants une mauvaise confiance.

Des études récentes ont montré que la perte de paquets sur les liens d'un réseau mesh est non négligeable. Dans (Aguayo, et al. 2004), les auteurs présentent une étude sur la perte de paquets dans un réseau mesh de 38 nœuds. Les résultats de leur étude montrent que la majorité des liens ont une probabilité de perte de paquets d'environ 50% (voir figure 52). Il est donc indispensable de considérer cette problématique lors de l'utilisation de l'IDS *Watchdog* afin de limiter les faux positifs qu'il déclenche et d'obtenir des systèmes de confiance d'attribuant une confiance aux nœuds selon

s'ils font suivre correctement les paquets et non selon l'état de leurs liens. Le point rouge sur l'une des courbes de la figure 52, indique que 160 liens sur les 230 qui envoient des données à 11Mbit/s, ont une probabilité supérieure ou égale à 40% de transmettre avec succès un paquet.

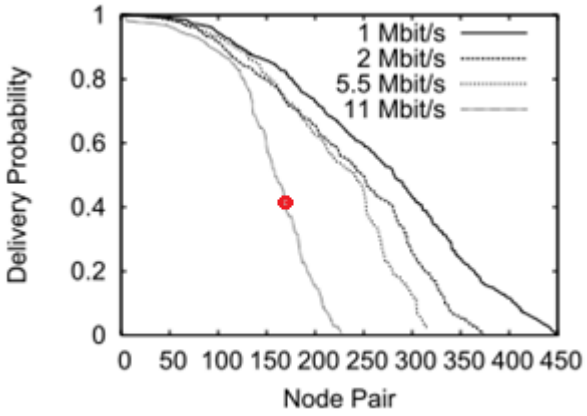


Figure 52 : Distribution de la probabilité de perte de paquets. Chaque point correspond à une paire émetteur/récepteur à un débit particulier. Seuls les pairs émetteur/récepteur qui ont envoyé au moins un paquet avec succès pendant l'expérience, sont représentées sur la figure (Aguayo, et al. 2004).

De plus, un nœud peut mal agir de différentes manières. Par exemple, il peut faire suivre les paquets de contrôle tels que les paquets de RREQ, mais ne pas faire suivre les paquets de données ; le but du nœud malveillant est alors de s'insérer sur les routes afin de nuire aux transmissions qui les empruntent. Un nœud peut au contraire, dans une logique de préservation de ses ressources, ne pas faire suivre les paquets de route afin de n'appartenir à aucune route et de ne pas avoir à faire suivre des paquets de données. Un mauvais nœud peut également fluctuer entre un bon et un mauvais comportement en alternant des périodes pendant lesquelles il fait suivre les données et des périodes pendant lesquelles il ne les fait pas suivre. Le but d'un tel comportement pour un nœud malveillant est d'être plus difficilement détectable. Nous proposons, dans ce chapitre, un système de confiance ayant pour objectifs de :

- limiter la quantité de faux positifs déclenchés par *Watchdog,*
- de détecter différents types de mauvais nœuds,
- d'assigner une confiance à un nœud reflétant le comportement réel du nœud.

Afin d'atteindre ces objectifs, nous proposons un système de confiance basé sur un IDS qui considère la perte de paquets sur les liens en comparant le modèle « normal » ou attendu de perte de paquets sur un lien lorsque son nœud récepteur est non mauvais avec la perte constatée sur ce lien. Cet IDS comprend plusieurs modules de détection afin de pouvoir surveiller différents types de mauvais nœuds. Chaque module considère différents paquets et utilisent un outil statistique pour vérifier si le taux de perte de paquets observé suit la distribution attendue. Finalement, cette IDS permet à notre

système de confiance d'assigner à chaque nœud une confiance reflétant le comportement réel du nœud.

6.3. Fonctionnement du système de confiance

Notre système permet à chaque nœud de collecter des données sur ses voisins via un IDS capturant différents types de mauvais comportements et considérant la perte de paquets sur les liens. Notre IDS se base sur un modèle de perte de paquets validé par des expériences afin de différentier les mauvais nœuds des nœuds dont les liens subissent une importante perte de paquet. Notre solution est composé de trois modules de surveillance inspirés de *Watchdog* : le premier détecte les mauvais nœuds qui ne font pas suivre les paquets de données mais envoient toujours des acquittements suite à la réception d'un paquet de données, le second détecte les nœuds qui s'inscrivent sur un maximum de routes mais ne font pas suivre les données et le troisième permet de détecter les mauvais nœuds qui modifient leur comportement, c.à.d. qui passent d'un bon comportement à un mauvais. Afin de considérer la perte de paquets sur un lien, ces modules comparent le modèle de perte de paquets sur ce lien lorsque le nœud récepteur n'est pas mauvais à celui observé via l'outil *Watchdog*. Cette comparaison s'effectue via des tests statistiques ou via un outil de détection de changement de comportement : CUSUM (Basseville et Nikiforov 1993) (*Cumulative Sum Control Chart*).

6.3.1. Modélisation de la perte de paquets

Dans cette section, nous proposons deux modèles de perte de paquets. Le premier modélise la perte de paquets sur un lien lorsque le nœud récepteur du lien est honnête et non défaillant et, le second, lorsque le nœud récepteur du lien est mauvais. Puis nous présentons les résultats d'une expérience qui prouve la validité de notre modèle de perte de paquets lorsque le nœud récepteur est bon.

6.3.1.1. Modélisation de la perte de paquets sur un lien où le récepteur est bon

Notre réseau mesh est composé d'un ensemble de nœuds noté V. Chaque paire de nœuds notée (i,j) forme un lien, le nœud i est l'émetteur et le nœud j est le récepteur. La variable aléatoire (V.A.) X_{ij} représente la probabilité de perte de paquets de données sur un lien (i,j) lorsque i envoie à j m paquets. Nous supposons que la variable aléatoire X_{ij} , lorsque m est suffisamment grand, suit une loi normale de moyenne μ_{ij} et d'écart-type σ_{ij} tronquée à l'intervalle [0,1] (voir figure 53).

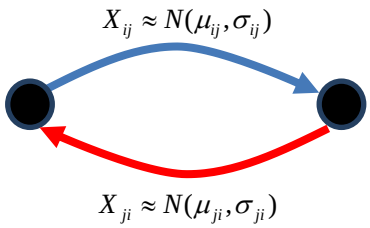


Figure 53 : Probabilité de perte de paquets sur les liens

Cette supposition a été validée expérimentalement, les résultats de l'expérience sont présentés par la suite dans ce chapitre. Soit la situation suivante ; un nœud i envoie à un nœud j m paquets pour qu'il les transmette à un nœud k, le nœud i écoute le canal en mode *promiscuous* pour vérifier que le nœud j transmette correctement les paquets qu'il lui a envoyé (voir figure 54). Un nœud i n'entend pas un nœud j transmettre le paquet qu'il lui a envoyé même si j est non mauvais si :

- le paquet est perdu sur le lien (i,j), c.à.d. lorsque le nœud i transfère à j le paquet.
- le paquet est perdu sur le lien (j,i) c.à.d. lorsque le nœud i écoute j transférer le paquet au nœud k.

On note X_{ij}^{o} le taux de paquets sur les m envoyés par i à j que le nœud i n'entend pas j transmettre et K_1 la variable aléatoire représentant le nombre paquets qui sont perdus sur le lien (i,j) parmi les m que i envoie à j. De plus, lorsque le nœud j est non mauvais (il n'abandonne aucun paquet par malveillance, égoïsme ou défaillance), on note K_2 la V.A. représentant le nombre de paquets que le nœud i n'entend pas j faire suivre à k parmi les $m - K_1$ qu'il lui envoie car ils sont perdus sur le lien (j,i) (voir figure 54).

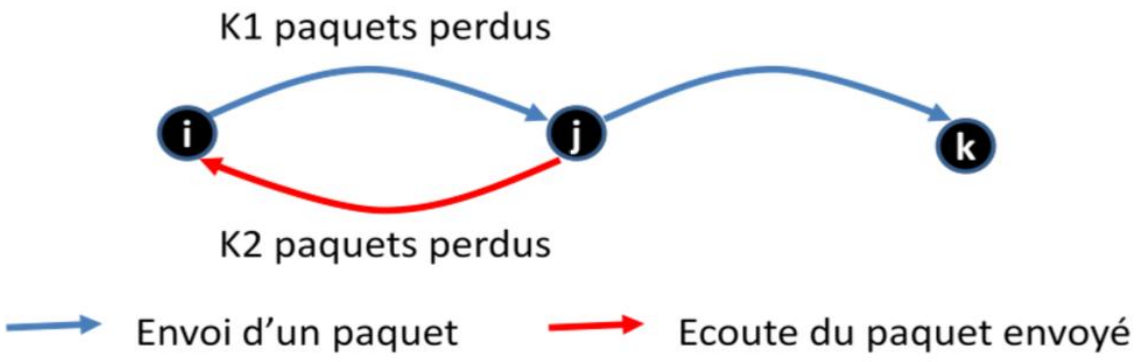


Figure 54: Le nœud i envoie m paquets à j, K_1 sont perdus sur le lien (i,j). Le nœud j s'il est non mauvais envoie $m - K_1$ paquets au nœud k et le nœud i entend seulement $m - K_1 - K_2$ d'entre eux.

La variable aléatoire X_{ij}^o, représentant le taux de paquets que le nœud i n'entend pas j transmettre sur les m qu'il lui a envoyé, a une probabilité d'être inférieure ou égale à x, lorsque j est non mauvais telle que :

$$\begin{aligned} p(X_{ij}^o \le x) &= p\left(\frac{K_1 + K_2}{m} \le x\right) \\ &= p(K_1 + K_2 \le mx) \\ &= \int_{k=0}^{k=mx} P(k + K_2 \le mx - k)\, f_{K_1}(k)\, dk \\ &= \int_{k=0}^{k=mx} P(K_2 \le mx - k)\, f_{K_1}(k)\, dk \\ &= \int_{k=0}^{k=mx} P\left(\frac{K_2}{m-k} \le \frac{mx-k}{m-k}\right) f_{K_1}(k)\, dk \end{aligned} \qquad \text{Équation 72}$$

Dans l'équation 72, $f_{k_1}(k)$ peut être substituée par $\frac{1}{m} * f_{X_{ij}}(\frac{k}{m})$, $P(\frac{K_2}{m-k} \le \frac{mx-k}{m-k})$ par $F_{X_{ji}}(\frac{mx-k}{m-k})$ avec $F_{X_{ij}}$ la fonction de répartition de la V.A. X_{ij}. Ainsi, à partir de l'équation 72, on peut obtenir la fonction de répartition de la V.A. X_{ij}^o telle que :

$$p(X_{ij}^o \le x) = \frac{1}{m} \int_{k=0}^{k=mx} F_{X_{ji}}\left(\frac{mx-k}{m-k}\right) f_{X_{ij}}\left(\frac{k}{m}\right) dk \qquad \text{Équation 73}$$

D'après l'équation 73, la variable aléatoire X_{ij}^o dépend des V.A. X_{ij} *et* X_{ji} et donc de leurs paramètres qui sont respectivement (μ_{ij}, σ_{ij}) *et* (μ_{ji}, σ_{ji}). Nous supposons que la variable aléatoire X_{ij}^o dont la fonction de répartition est donnée par l'équation 73 suit une loi normale. Pour le prouver notre hypothèse, nous proposons le test suivant. Via le logiciel R (Team 2008), un logiciel libre de traitement des données et d'analyse statistiques, 100 fois n réalisations notés $(x_1, x_2, \ldots x_n)$ de la

V.A. X_{ij}^{o} sont générées selon la fonction de répartition de l'équation 73 avec différentes valeurs de (μ_{ij}, σ_{ij}) et (μ_{ji}, σ_{ji}). Comme les variable μ_{ij} et μ_{ji} représentent des taux et les variables σ_{ij} et σ_{ji} les écart-types de taux, ils sont compris entre 0 et 1. Les valeurs de μ_{ij}, σ_{ij}, μ_{ji} et σ_{ji} utilisés pour générer une réalisation x_i de la variable aléatoire X_{ij}^{o} sont choisies aléatoirement dans l'intervalle [0,1]. Un test statistique de Kolmogorov-Smirnov (KS-test) est effectué sur chaque échantillon $(x_1, x_2, \ldots x_n)$ pour déterminer s'il suit une loi normale de moyenne μ_{ij}^{o} et d'écart-type σ_{ij}^{o}, les valeurs de μ_{ij}^{o} et σ_{ij}^{o} sont obtenues en calculant respectivement la moyenne et l'écart-type de l'échantillon est effectué le test. Le KS-test effectué sur un échantillon évalue l'hypothèse que l'échantillon suit une loi normale de paramètres μ_{ij}^{o} et σ_{ij}^{o} (l'hypothèse nulle) et l'hypothèse que l'échantillon ne suit pas une loi normale de paramètres μ_{ij}^{o} et σ_{ij}^{o} (l'hypothèse alternative).

La valeur n correspond au nombre d'élément par échantillon, elle est fixée à 1000. Le risque d'hypothèse de première espèce du test représente la probabilité de rejeter à tort l'hypothèse nulle, c.à.d. l'hypothèse que l'échantillon suive une loi normale de paramètres $(\mu_{ij}^{o}, \sigma_{ij}^{o})$, ce risque est fixé à une valeur usuelle, 0.05. Chaque KS-test effectué sur un échantillon retourne le taux de probabilité d'obtenir la même valeur pour l'échantillon de départ si l'hypothèse nulle est vraie, ce taux est appelé la valeur p ou *p-value*. Si p est supérieure au seuil α, l'hypothèse nulle est acceptée sinon elle est rejetée. La figure 55 présente les résultats de notre test. Chaque point représente la valeur p obtenue avec un KS-test sur un échantillon dont chaque élément à été généré selon la fonction de répartition de l'équation 73.

Toutes les valeurs p des tests sont supérieures à α, l'hypothèse nulle est donc acceptée pour les 100 KS-tests. On peut donc en conclure que, lorsque le nœud j est bon, la variable aléatoire X_{ij}^{o}, représentée par la fonction de répartition de l'équation 73, suit une loi normale.

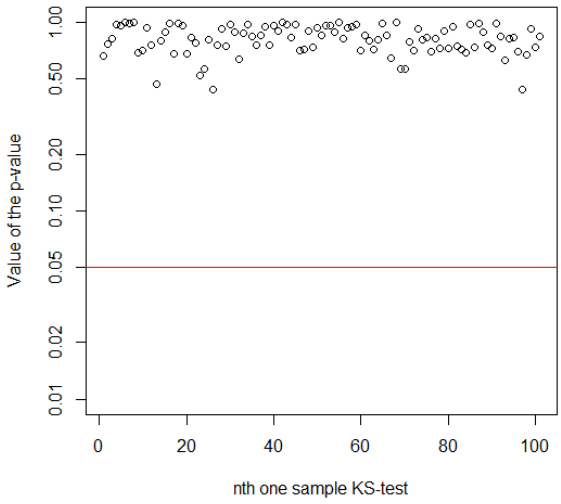


Figure 55 : Valeur de p retournée par chacun des 100 KS-tests. Chaque KS-test compare un échantillon de n réalisation de la variable X_{ij}^{o} avec une loi normale.

6.3.1.2. Modélisation de la perte de paquets sur un lien où le récepteur est mauvais

Un nœud j mauvais ne fait pas suivre l'ensemble des paquets qu'il reçoit de ses voisins ; il en abandonne certains. On note K_3 la variable aléatoire représentant le nombre de paquets que le nœud j abandonne sur les $m - K_1$ qu'il reçoit avec succès de i. Le nombre de paquets perdus sur le lien (j,i) sur les $m - K_1 - K_3$ que le nœud j envoie à i est représenté par la variable aléatoire K_4. Ainsi, la probabilité que, la variable aléatoire X_{ij}^{o}, représentant le taux de paquets que le nœud i a entendu j faire suivre sur les m paquets qu'il a envoyé à j lorsque j est un nœud mauvais, soit inférieure ou égale à x équivaut à :

$$p\left(X_{ij}^{o} \leq x\right) = p\left(\frac{K_1 + K_3 + K_4}{m} \leq x\right) \qquad \text{Équation 74}$$

Les équations 74 et 72 représentent respectivement la fonction de distribution X_{ij}^{o}, lorsque j est un bon nœud et lorsque j est un mauvais nœud. Comme elles sont différentes, on peut en conclure que la distribution de perte de paquets que le nœud i observe est différente selon si le nœud j est mauvais (équation 74) ou non (équation 72). Les modules de notre IDS se basent sur cette observation pour détecter les mauvais nœuds. Chaque module détecte si un nœud est mauvais en comparant la distribution du taux de perte de paquets observés pour un voisin avec le taux de perte

de paquets « normale » de ce dernier. Le taux de perte de paquets normale d'un voisin est celui observé lorsque le voisin est non mauvais ; il suit une loi normale.

6.3.1.3. Validation de notre modèle de perte de paquets lorsque le nœud voisin est bon

Nous avons précédemment fait l'hypothèse que la perte de paquets sur un lien suit une loi normale. En admettant cette hypothèse, nous avons montré que la variable aléatoire X_{ij}^{o} représentant le taux de paquet que le nœud i n'entend pas j retransmettre suit également une loi normale lorsque j n'est pas un nœud mauvais.

Afin de prouver que la variable aléatoire X_{ij}^{o} suit une loi normale lorsque le nœud j est non mauvais, nous proposons l'expérience suivante (voir figure 56). Un ordinateur représentant le nœud i envoie n paquets à un second ordinateur représentant le nœud j pendant une heure. A chaque fois que le nœud j reçoit avec succès le paquet, il l'acquitte puis le diffuse. Le nœud i enregistre, pour chaque paquet envoyé, s'il reçoit un acquittement et s'il entend le nœud j diffuser son paquet.

La figure 56 représente le premier étage du bâtiment où se déroulent l'expérience et les 3 liens utilisés. Pour chaque lien, une extrémité représente le nœud i et l'autre extrémité le nœud j. La longueur des liens et les paramètres de l'expérience sont présentés dans le tableau 9.

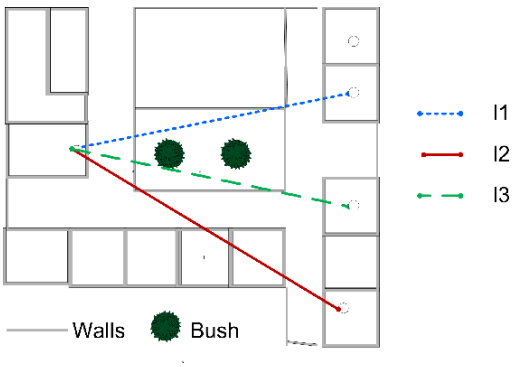


Figure 56: Carte du premier étage du bâtiment

Pour chaque lien, nous calculons, à partir des données collectées, un échantillon de données dénotés $(x_1, x_2, \dots x_z)$. Chaque élément x_i est une réalisation de la variable aléatoire X_{ij}^{o} obtenue en divisant le nombre de paquets que le nœud i a entendu j faire suivre sur les m paquets qu'il lui a envoyé. Un KS-test est effectué sur chaque échantillon $(x_1, x_2, \dots x_z)$ afin d'évaluer si ce dernier suit

une loi normale dont la moyenne et l'écart-type sont ceux de l'échantillon. Le risque de première espèce du test est fixé à 0.01, ainsi, si le résultat p du test est supérieur à 0.01, alors on accepte l'hypothèse que l'échantillon suit une loi normale sinon on la rejette. A partir des données collectées par le nœud i, on effectue plusieurs KS-tests. Chaque KS-test est effectué sur un échantillon de données où chaque élément est calculé avec un nombre de paquets m différent ; m représente le nombre de paquets que le nœud i doit envoyer au nœud j pour obtenir une réalisation de la variable aléatoire X_{ij}^{o} .

Tableau 9 : Paramètres de l'expérience

Fréquence	2.4Ghz
Débit	54 Mbit/s
Taille des paquets	1000 octets
Longueur du lien l1	11 mètres
Longueur du lien l2	12 mètres
Longueur du lien l3	11 mètres

Les figures 57, 58 et 59 présentent les $p-values$ des KS-tests effectués respectivement sur les données issues du lien 1, du lien 2 et du lien 3. La ligne rouge, sur chaque figure, représente la valeur du risque de première espèce α. Chaque point sur ces figures représente le $p-value$ obtenu avec un KS-test effectué avec une certaine valeur de m. On remarque que, pour l'ensemble des trois liens, p est toujours supérieure à 0.01 à partir de $m > 1000$. Ainsi, lorsque m est assez grand ($m \geq 1000$), on accepte l'hypothèse que les échantillons suivent une loi normale. Cette expérience valide notre hypothèse, que X_{ij}^{o} suit une loi normale pour tout lien (i, j) d'un réseau mesh, lorsque m est grand.

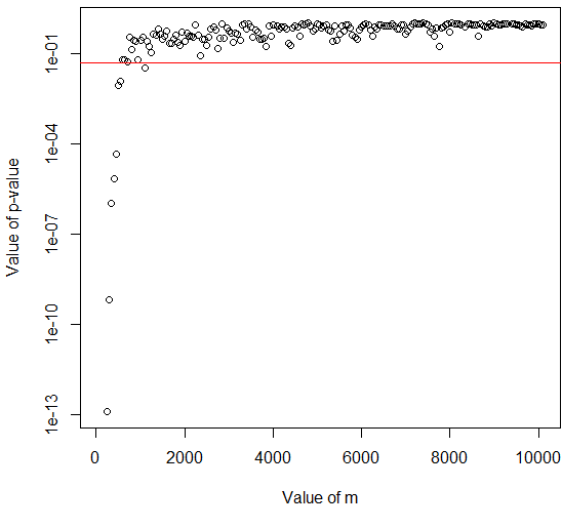


Figure 57 : $p - value$ pour chaque KS-test effectué sur des échantillons de données de différentes tailles m récoltées sur le lien 1

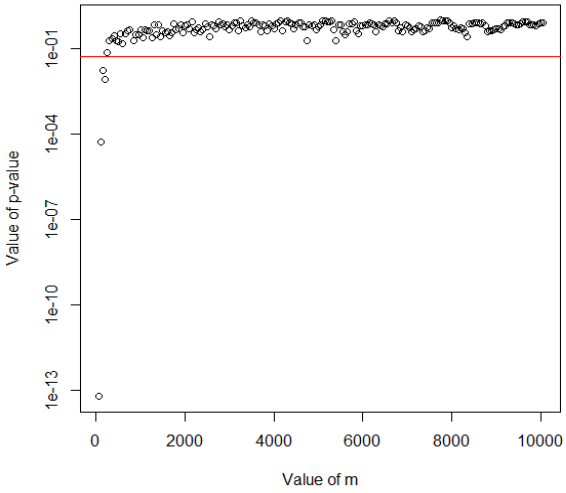


Figure 58 : $p - value$ pour chaque KS-test effectué sur des échantillons de données de différentes tailles m récoltées sur le lien 2

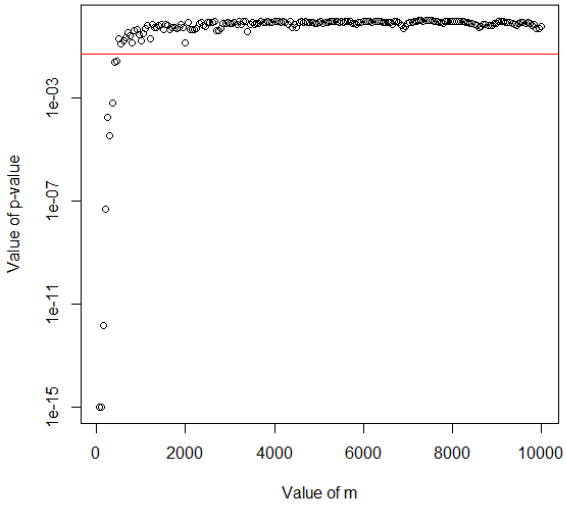


Figure 59 : $p - value$ pour chaque KS-test effectué sur des échantillons de données de tailles m récoltées sur le lien 3

6.3.2. Système de détection multiple des mauvais nœuds

Pour détecter un maximum de mauvais comportements, l'IDS de notre système de confiance est composé de trois modules, chacun surveillant un type de mauvaise conduite.

6.3.2.1. Premier module de détection des mauvais nœuds de l'IDS

Ce premier module a pour but de détecter les nœuds malveillants qui s'inscrivent sur un maximum de routes pour perturber de nombreuses communications. Pour s'inscrire sur un maximum de routes, les nœuds malveillants font suivre tous les paquets de requête de route (RREQ) ou de réponse de route (RREP). Par la suite, ces mauvais nœuds, pour perturber les communications sur les routes auxquelles ils appartiennent, abandonnent des paquets de données. Dans cette section, on utilisera le terme paquet de route pour désigner aussi bien un paquet RREP ou RREQ. Contrairement à un nœud mauvais, un bon nœud transfère tous les paquets de route et les paquets de données qu'il reçoit avec succès. Afin de détecter si l'un de ses voisins est mauvais, un nœud surveille s'ils font suivre correctement les paquets de route et les paquets de données qu'il leur envoie. Ainsi un nœud enregistre pour chacun de ses voisins :

- le nombre de paquets de route qu'il lui envoie et que ce dernier doit faire suivre ,
- le nombre de paquets de route qu'il lui a envoyé et qu'il entend son voisin faire suivre correctement,
- le nombre de paquets de données qu'il lui envoie et que ce dernier doit faire suivre ,
- le nombre de paquets de données qu'il lui a envoyé et qu'il entend son voisin fait suivre correctement.

Pour rappel, la variable aléatoire X_{ij}^{o} représente le taux de paquets de données que i entend j retransmettre correctement sur les m qu'il lui a envoyé. On note Y_{ij}^{o} , la variable aléatoire représentant le taux de paquets de route que i entend j retransmettre correctement sur les m qu'il lui a envoyé. La probabilité qu'un nœud i après avoir envoyé un paquet de route à un nœud j entende ce dernier le retransmettre équivaut, que le nœud j soit mauvais ou non, à la probabilité que le paquet de route ne soit ni perdu sur le lien (i,j) ni sur le lien (j,i). La probabilité qu'un nœud i après avoir envoyé un paquet de données à nœud j entende ce dernier le réémettre, lorsque le nœud j est non mauvais, équivaut à la probabilité que le paquet de données ne soit ni perdu sur le lien (i,j) ni sur le lien (j,i). Ainsi, si l'on suppose que la perte de paquets de données sur un lien est

la même qu'on y transfère un paquet de données ou de route, alors la probabilité que le nœud i, après avoir envoyé un paquet de route à j entende ce dernier le retransmettre équivaut à la probabilité que le nœud i, après avoir envoyé un paquet de données à un nœud j non mauvais entende j le retransmettre. Ainsi comme ces deux probabilités sont équivalentes, on en déduit que, si le nœud j est non mauvais, alors la distribution des variables aléatoires Y_{ij}^o et X_{ij}^o sont identiques.

Lorsque le nœud j est mauvais, ce dernier abandonne alors délibérément des paquets de données mais continue à envoyer à chaque paquet reçu un acquittement. Par conséquent la distribution de la variable aléatoire X_{ij}^o change et les distributions de X_{ij}^o et Y_{ij}^o deviennent alors différentes.

Pour détecter si le voisin j d'un nœud i est malveillant, le premier module compare la distribution des réalisations des variables aléatoire Y_{ij}^o et X_{ij}^o que le nœud i observe. Si les deux distributions sont équivalentes, il en conclut que le nœud j ne privilégie pas le suivi des paquets de route sur le suivi de paquets de données et que donc il ne tente pas de s'inscrire sur un maximum de routes pour les perturber.

Pour obtenir une réalisation y_i de la variable aléatoire Y_{ij}^o, le nœud i enregistre à chaque fois qu'il envoie m paquets de route, le nombre p de paquets de route qu'il a entendu j faire suivre correctement et calcule ensuite $y_i = \frac{p}{m}$. Pour obtenir une réalisation x_i de la variable aléatoire X_{ij}^O, le nœud i enregistre à chaque fois qu'il envoie m paquets de données le nombre p' de paquets qu'il a entendu j faire suivre correctement et calcule ensuite $x_i = \frac{p'}{m}$. Afin de comparer la distribution de n réalisations des variables aléatoires Y_{ij}^o et X_{ij}^o notées respectivement $(y_1, y_2 \dots, y_n)$ et $(x_1, x_2 \dots, x_n)$, le module utilise le test de Kolmogorov-Smirnov à deux échantillons. Ce test permet de déterminer si deux échantillons suivent la même distribution avec une probabilité de première espèce fixé. Le KS-test calcule la distance entre les fonctions de répartition du premier et du second échantillon et selon la valeur de cette dernière détermine s'ils suivent la même distribution. Ce test peut se décomposer de trois étapes. La première étape consiste à calculer la fonction de répartition de chacun des deux échantillons. La fonction de répartition pour un échantillon $(x_1, x_2 \dots, x_n)$ est obtenue avec la formule suivante :

$$F_{1,n} = \frac{1}{n} \sum_{i=1}^{n} I_{x_i \leq x} \qquad \text{Équation 75}$$

où $I_{x_i \leq x}$ est la fonction indicatrice ; elle équivaut à 1 si $x_i \leq x$ et 0 sinon. On note $F_{1,n}$ la fonction de répartition obtenue avec l'échantillon $(x_1, x_2 \dots, x_n)$ composé de n réalisations de X_i^o et $F_{2,n}$ la

fonction de répartition obtenue avec l'échantillon $(y_1, y_2 \ldots, y_n)$ composé de n réalisations de la variable aléatoire Y_{ij}^o. La seconde étape calcule la distance entre les fonctions de répartition des deux échantillons via la formule suivante :

$$D_{n,n} = \sqrt{\frac{n*n}{n+n}} \sup_x |F_{1,n}(x) - F_{2,n}(x)| \qquad \text{Équation 76}$$

La troisième étape du KS-test à deux échantillons vérifie si $D_{n,n}$ est inférieure ou égale à un certain seuil noté $K(\alpha)$. Si elle est inférieure, alors les distributions sont considérées comme équivalentes sinon elles sont considérées comme différentes. La distance entre deux fonction de répartition issues d'échantillons qui suivent la même loi suit une distribution bien connue ; la distribution de Kolmogorov. Il existe des tables qui, pour une variable aléatoire K suivant la distribution de Kolmogorov, donne le seuil $K(\alpha)$ à partir duquel la probabilité que K soit supérieure à $K(\alpha)$équivaut à 1-α. Ainsi, si Kest une variable aléatoire suivant une loi de Kolmogorov alors :

$$P[K \leq K(\alpha)] = 1 - \alpha \qquad \text{Équation 77}$$

Le risque de première espèce α de notre test est fixé à 0.05. Comme la variable aléatoire $D_{n,n}$ suit une distribution de Kolmogorov, le module estime que les deux échantillons suivent la même distribution avec un risque de première espèce 0,05 si :

$$D(n,n) \leq K(0{,}05) \qquad \text{Équation 78}$$

le seuil $K(0{,}05)$ est obtenu via les tables de la loi de Kolmogorov. Si l'équation 78 est vérifiée, l'hypothèse d'équivalence de la distribution des V.A. Y_{ij}^o et X_{ij}^o est acceptée et le premier module du nœud i considère le nœud j non mauvais et incrémente alors le nombre p_{ij}^1 d'interactions positives entre i et j. Si l'équation 78 n'est pas vérifiée alors le premier module du nœud i considère$j$ comme mauvais et augmente le nombre n_{ij}^1 d'interactions négatives entre les nœuds i et j.

6.3.2.2. Le second module de détection des mauvais nœuds de l'IDS

Lorsqu'un nœud reçoit avec succès un paquet de données, il envoie un acquittement puis fait suivre le paquet de données. Or, certains mauvais nœuds, pour ne pas éveiller les soupçons de leurs voisins envoient tous les paquets d'acquittement mais ne transfèrent pas tous les paquets de données afin de perturber les communications du réseau. Le second module a pour but de détecter ce type de mauvais nœuds. Pour les détecter, chaque nœud enregistre les données suivantes sur chaque voisin :

- le nombre de paquets de données qu'il envoie à son voisin et que ce dernier doit faire suivre
- le nombre d'acquittement qu'il reçoit en provenance de son voisin
- le nombre de paquets de données que son voisin a correctement retransmis et qu'il a entendu

La variable aléatoire Z_{ij}^{o} représente le taux de paquets d'acquittement que le nœud i a reçu du nœud j après lui avoir envoyé m paquets à transférer. En supposant que le taux de perte de paquets de données sur un lien est identique au taux de perte d'acquittement sur ce même lien alors Z_{ij}^{o}=X_{ij}^{o}, si le nœud j est non mauvais. Cette équivalence peut être démontré prouvée de la même manière que l'équivalence de la distribution entre les variables aléatoires Y_{ij}^{o}=X_{ij}^{o} du module 1 de notre IDS. Pour détecter si le nœud voisin j d'un nœud i est malveillant, le second module compare la distribution des réalisations des variables aléatoire Z_{ij}^{o} et X_{ij}^{o} obtenues par le nœud i. Si les deux distributions sont équivalentes, il en conclure que le nœud j ne privilégie pas l'envoi des acquittements sur le suivi de paquets de données et est non malveillant.

Pour obtenir une réalisation z_i de la variable aléatoire Z_{ij}^{o}, le nœud i enregistre, à chaque fois qu'il envoie m paquets de données à faire suivre, le nombre a d'acquittement que j lui envoie et calcule ensuite $z_i = \frac{a}{m}$. Il procède de même pour obtenir une réalisation x_i de la variable aléatoire X_{ij}^{o}. Afin de comparer la distribution de n réalisations des V.A. Z_{ij}^{o} et X_{ij}^{o} notées respectivement $(z_1, z_2 \dots, z_n)$ et $(x_1, x_2 \dots, x_n)$, le module utilise le test de Kolmogorov-Smirnov à deux échantillons avec un risque de première espèce fixé à 0.05. Si d'après le KS-test les deux échantillons suivent la même distribution, alors le second module du nœud i considère le nœud j non mauvais et incrémente le nombre p_{ij}^{2} d'interactions positives que le nœud i a eu avec le nœud j, sinon le module incrémente le nombre n_{ij}^{2} d'interactions négatives entre i et j.

6.3.2.3. Troisième module de détection des mauvais nœuds de l'IDS

Le troisième module nécessite, au préalable, que chaque nœud i connaisse la distribution de la variable aléatoire X_{ij}^{o} pour chacun de ses voisins j, lorsque ce dernier est non mauvais. Pour rappel, la variable aléatoire X_{ij}^{o} représente le taux de paquets que le nœud i entend j retransmettre sur les m paquets qu'il lui a envoyé. Cette V.A. suit une loi normale, lorsque le nœud j est bon, définie par une moyenne μ_{ij}^{o} et un écart-type σ_{ij}^{o}. Un nœud peut, par exemple, obtenir la distribution de la V.A. X_{ij}^{o} en effectuant des tests sur ses liens avant l'ouverture du réseau mesh aux utilisateurs. Ce module a pour but de détecter les nœuds qui deviennent mauvais en surveillant les changements de distribution de la variable aléatoire X_{ij}^{o}. Il détecte ainsi les nœuds qui font suivre :

- une proportion moins importante de paquets qu'avant. Un nœud peut faire suivre une proportion moins importante de paquets s'il devient défaillant, égoïste ou malveillant.
- une proportion plus importante de paquets qu'avant. Il est en effet possible qu'un nœud fasse suivre plus de paquets qu'avant si le routeur représentant le nœud a été remplacé par un routeur avec une meilleur puissance. Le but de l'attaquant, en remplaçant le routeur, est d'intercepter tous les messages que ce dernier doit transférer.

Afin de détecter les changements de comportement d'un voisin j, un nœud i en mode *promiscuous,* enregistre le nombre de paquets qu'il lui envoie et qu'il entend j faire correctement suivre. Il peut ainsi calculer les réalisations de la variable aléatoire X_{ij}^{o} notées $x_1, x_2 \dots, x_n$. Le module lance ensuite la méthode CUSUM (Le Boudec et Patrick 2001) afin de détecter s'il y a eu un changement dans le comportement de suivi de paquets de son voisin j. CUSUM est une méthode statistique et séquentielle de détection de changement qui surveille les changements dans la distribution d'une variable. Pour détecter ces changements, CUSUM calcule périodiquement deux sommes, la somme cumulée positive C^{+} et la somme cumulée négative C^{-}. La somme cumulée positive C^{+} est la somme des déviations entre chaque réalisation x_z de la variable aléatoire X_{ij}^{o} observée par le nœud i et la valeur $\mu_{ij}^{o} + K$, K est le coefficient de sensibilité et sera expliqué par la suite. Ainsi, la somme cumulée positive suite à l'obtention par le nœud i de la $z^{ème}$ réalisation de la variable aléatoire X_{ij}^{O} équivaut à :

$$C_z^{+} = \max[0, x_z - (\mu_{ij}^{o} + K) + C_{z-1}^{+}] \text{ et } C_0^{+} = 0 \qquad \text{Équation 79}$$

La somme cumulée négative C^{-} est la somme cumulée des déviations entre chaque réalisation x_z de la V.A. X_{ij}^{O} observée par le nœud i et la valeur $\mu_{ij}^{o} - K$ lorsque $x_z \leq \mu_{ij}^{o} - K$:

$$C_z^- = \max[0, (\mu_{ij}^o - K) - x_z + C_{z-1}^-] \text{ et } C_0^- = 0 \qquad \text{Équation 80}$$

Après avoir calculé C_z^+ et C_z^-, CUSUM compare leur valeur à l'intervalle de décision H. La variable de décision H représente le seuil à partir duquel une alerte est déclenchée, elle est expliquée plus en détail par la suite. Si $C_z^+ \geq H$ ou $C_z^- \geq H$ alors CUSUM déclenche une alerte sinon il n'en déclenche pas. Si CUSUM déclenche une alerte alors le module considère que la distribution de la variable aléatoire X_{ij}^o a changé et que le nœud j est mauvais et il incrémente le nombre d'interactions négatives n_{ij}^3 entre i et j sinon il augmente le nombre d'interactions positives p_{ij}^3 entre i et j. La méthode CUSUM nécessite de fixer trois paramètres :

- δ : le décalage que l'on souhaite détecter. Ce décalage est exprimé selon la valeur de l'écart-type σ_{ij}^o de la distribution attendue. La valeur de δ permet de calculer la valeur μ_t à partir de laquelle CUSUM considère que la distribution de la V.A. a changé :

$$\mu_t = \mu_{ij}^o \pm \sigma_{ij}^o \delta \qquad \text{Équation 81}$$

- K : le coefficient de sensibilité. Cette valeur est généralement choisie à mi-distance entre la valeur moyenne attendue μ_{ij}^o et la valeur μ_t à partir de laquelle CUSUM détecte un changement (Montgomery 2009).
- H : l'intervalle de décision. Lorsque la somme cumulée positive ou la somme cumulée négative est supérieure à l'intervalle de décision H, alors CUSUM détecte un changement dans la distribution de la variable aléatoire. Il existe des tables (Montgomery 2009) qui associent à une valeur de K, la valeur de H permettant d'obtenir de bonnes performances avec CUSUM.

La figure 60 illustre la méthode CUSUM. Cette figure représente la carte de contrôle obtenue avec CUSUM sur un échantillon de 40 données dont les 20 premières ont été générées selon une loi normale de paramètres $\mu_{ij}^o = 0{,}6$ et $\sigma_{ij}^o = 0{,}2$ et les vingt suivantes selon une loi normale de paramètres $\mu_{ij}^o = 0{,}4$ et $\sigma_{ij}^o = 0{,}2$. Dans cet exemple, la distribution de référence est utilisée pour générer les vingt premières données. Les paramètres de CUSUM ont été fixés selon les recommandations données dans le livre de Montgomery (Montgomery 2009). La carte de contrôle a été obtenue avec le logiciel R (Team 2008).

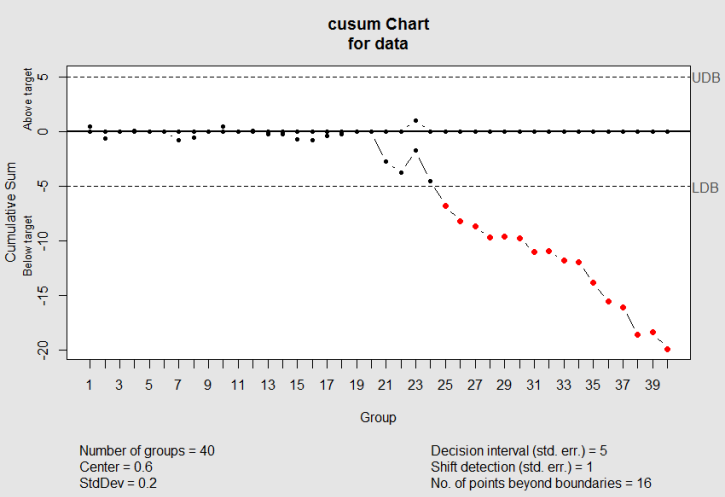


Figure 60: Carte de contrôle établie par la méthode CUSUM sur un échantillon de 40 données

La figure 60 est composée de 40 points chacune. La ligne supérieure est partiellement confondue avec la ligne en gras 0, le $i^{ème}$ point de cette ligne représente la somme cumulée positive obtenue avec la $i^{ème}$ donnée. La ligne inférieure est, également, partiellement confondue avec la ligne 0 jusqu'à l'obtention de la $21^{ème}$ donnée, première données générée avec une loi normale différentes des vingt premières données. La ligne en pointillé LDB représente la valeur -H, CUSUM lance une alerte lorsque la somme cumulée négative est en dessous de ce seuil. La ligne UDB représente la valeur +H, CUSUM lance une alerte lorsque la somme cumulée positive est au dessus de ce seuil. Sur la figure, la somme cumulée négative devient inférieure au seuil LDB à partir de la $25^{ème}$ donnée. Ainsi, dans cet exemple, CUSUM détecte un changement à la cinquième donnée générée par une distribution différente.

Appliquer à notre module, la méthode CUSUM établit une somme cumulée positive et négative à chaque fois qu'un nœud i obtient une réalisation de la variable aléatoire X_{ij}^{O} pour un de ses voisins j. Si CUSUM déclenche une alerte alors le nœud j est considéré comme malveillant et le nombre d'interactions négatives n_{ij}^{2} entre i et j est incrémenté, sinon j est considéré comme non mauvais et le nombre d'interactions positives p_{ij}^{2} entre i et j est incrémenté.

6.3.3. Système de calcul de la confiance

Notre IDS est composé de trois modules ; chacun surveille un type de mauvais comportement. Chaque module permet à un nœud i d'obtenir des informations sur chacun de ses voisins j, le

premier module renvoie la paire de valeurs $< n_{ij}^1, p_{ij}^1 >$, le second $< n_{ij}^2, p_{ij}^2 >$, le troisième $< n_{ij}^3, p_{ij}^3 >$. Chacune de ces paires de valeurs représente le nombre d'interactions positives et négatives que le module a pu observer entre le nœud i et le nœud j, ces valeurs sont périodiquement remises à zéro. Chaque nœud i calcule trois valeurs de confiance c_{ij}^1, c_{ij}^2 et c_{ij}^3 pour chacun de ses voisins j, chaque valeur de confianceest obtenue respectivement avecdes données du premiermodule, du secondmodule et du troisième. Une valeur de confiance considère à la fois les interactions passées et présentes du nœud, plus une interaction est ancienne et moins elle aura d'influence sur la valeur de la confiance. La confiance c_{ij}^z avec $z = \{1,2,3\}$, qu'un nœud i a en un nœud j via le z^{th} module de l'IDS est calculée via la formule suivante :

$$c_{ij}^z = \beta * c_{ij}^{z,old} + (1-\beta)\frac{p_{ij}^z}{p_{ij}^z + n_{ij}^z} \qquad \text{Équation 82}$$

avec $c_{ij}^{z,old}$ la valeur de l'ancienne confiance et β le facteur d'oubli dont la valeur est située entre 0 et 1. La valeur de β dépend du contexte, l'administrateur du réseau mesh fixe la valeur de β selon l'importance de l'historique ; plus les interactions passées sont importantes et plus la valeur de β est proche de 1.

Un nœud mauvais peut être détecté par un module mais pas par les autres. C'est pourquoi, il suffit qu'un seul des modules d'un nœud possède de nombreuses interactions négatives sur l'un de ses voisins pour que ce dernier soit considéré comme mauvais. En d'autres termes, il suffit qu'une des trois valeurs de confiance c_{ij}^1, c_{ij}^2 et c_{ij}^3 que le nœud i possède sur le nœud j soit basse, pour qu'il considère ce dernier comme mauvais. La confiance finale C_{ij} qu'un nœud i porte dans un nœud j est la valeur minimale de l'ensemble des confiances c_{ij}^1, c_{ij}^2 et c_{ij}^3 ainsi :

$$C_{ij} = \min\{c_{ij}^1, c_{ij}^2, c_{ij}^3\} \qquad \text{Équation 83}$$

La figure 61 résume le système de confiance implémenté dans chaque nœud permettant à ce dernier de déterminer la confiance qu'il a dans chacun de ses voisins.

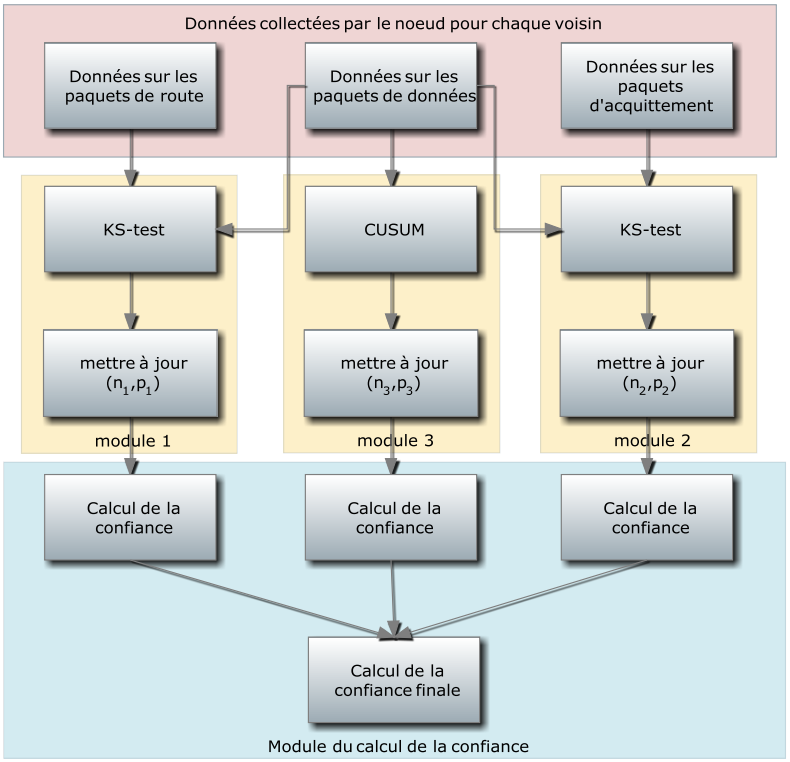


Figure 61 : Système de confiance installé sur chaque nœud du réseau mesh afin d'établir la confiance qu'il a en chacun de ses voisins

6.4. Evaluation du système de confiance

Afin d'évaluer notre système de confiance, nous avons testé, avec le logiciel statistique R, la capacité de chaque module de notre IDS à détecter des nœuds. Dans un second temps, nous avons comparé via le logiciel de simulation ns2, notre solution avec un système de confiance assez classique basé sur l'IDS Watchdog. La première évaluation a pour objectif de vérifier que :

- le premier module détecte les nœuds qui font suivre tous les paquets de route et pas tous les paquets de données,
- le second module détecte les nœuds envoyant tous les acquittements et pas tous les paquets de données,
- le troisième module détecte les nœuds qui modifient leur comportement d'envoi de données.

Les résultats de l'évaluation montrent que le taux de faux positifs et de faux négatifs de chaque module de l'IDS de notre système de confiance, est faible. Cependant, ce taux augmente avec l'accroissement de l'écart-type de la perte de paquets. La seconde évaluation a pour objectifs de

montrer que notre système contrairement aux systèmes de confiance existant assigne aux nœuds une valeur de confiance qui reflète leur comportement et non la qualité de leurs liens et permet ainsi de détecter avec plus de précision les mauvais des bons nœuds.

6.4.1. Evaluation du premier et du second module de l'IDS de notre système de confiance

Le premier et le second module utilisent le test de Kolmogorov-Smirnov afin de vérifier si un nœud est mauvais. Le premier module, pour détecter si le voisin j d'un nœud i est mauvais, évalue si la distribution des réalisations de la variable aléatoire Y_{ij}^o est identique à celle des réalisations de la V.A. X_{ij}^o tandis que le second module compare la distribution des réalisations de la V.A. Z_{ij}^o à celle des réalisations de la V.A. X_{ij}^o.

Pour valider le premier et second module, nous avons utilisé le logiciel statistique R. Les réalisations des variables aléatoires Z_{ij}^o et Y_{ij}^o qu'un nœud i collecte sur un nœud j ont été générées selon une loi normale de moyenne μ_{ij}^o et d'écart-type σ_{ij}^o tronquée en dessous de 0 et au dessus de 1. Les réalisations collectées par un nœud i sur son voisin j de la V.A. X_{ij}^o ne sont pas les mêmes selon si le nœud j est mauvais ou non et s'il est mauvais, selon le pourcentage de paquets qu'il abandonne. Ainsi, les réalisations de la V.A. X_{ij}^o, obtenues par un nœud i sur son voisin j ont été générées selon une loi normale de moyenne μ_{ij}^o et d'écart-type σ_{ij}^o tronquée en dessous de 0 et au dessus de 1, chaque élément a été ensuite multiplié par $1-p$, p représentant le taux de paquets que le nœud j abandonne quant il est mauvais. Si $p \leq 0{,}1$ alors les réalisations obtenues de la V.A. X_{ij}^o représentent celles d'un nœud j non mauvais. Par contre, si $p \geq 0{,}1$, alors les réalisations obtenues de la V.A. X_{ij}^o représentent celles d'un nœud j mauvais.

Afin de valider les modules 1 et 2, nous souhaitons montrer que le KS-test permet de détecter si l'échantillon de réalisations de la V.A. Z_{ij}^o (ou l'échantillon de réalisations de la V.A. de Y_{ij}^o) a la même distribution que celui des réalisations de la V.A. X_{ij}^o. En d'autres termes, si le KS-test est capable de détecter en comparant un échantillon de réalisations de Z_{ij}^o (ou de Y_{ij}^o) avec un échantillon de réalisations de X_{ij}^o, si l'échantillon représentant les réalisations de la V.A. X_{ij}^o a été généré avec $p \geq 0{,}1$.

Nous souhaitons également déterminer la taille idéale n des échantillons à partir de laquelle le nombre de faux positifs et de faux négatifs dédenchés par le module n'évolue presque plus. Les

figures 62, 63 et 64 représentent, pour une valeur donnée de σ_{ij}^{o} , le pourcentage de faux négatifs et de faux positifs obtenus par un module lorsque les échantillons ont une taille n. Chaque point d'une courbe est la moyenne des faux positifs et négatifs obtenus pour une taille n d'échantillon et une valeur de σ_{ij}^{o} sur 81 KS-tests. Chacun des 81 KS-tests réalisés avec une même taille n d'échantillon, est effectué sur des échantillons de données générées avec une paire différente de valeurs de μ_{ij}^{o} et p; μ_{ij}^{o} et p prennent des valeurs entre 0.1 et 0.9 par pas de 0.1.

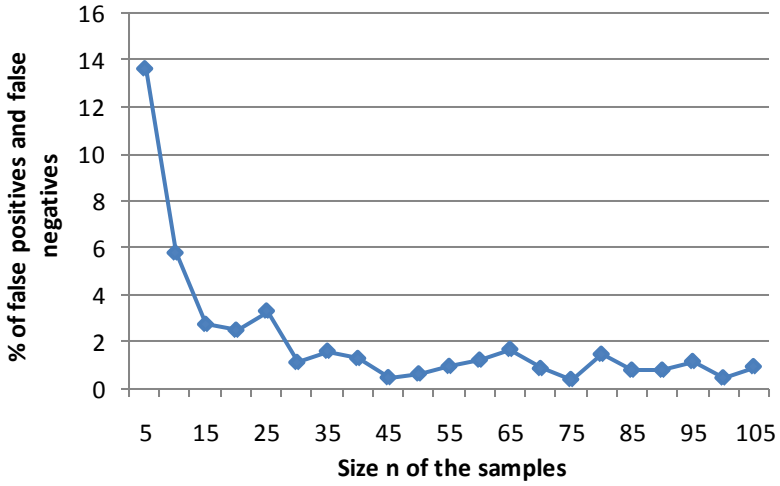


Figure 62 : Pourcentage de faux négatifs et de faux positifs lorsque les échantillons sont de taille n et que l'écart-type de la loi normale tronquée en 0 et en 1 générant les échantillons est de 0.1 ($\sigma_{ij}^{o} = 0.1$).

La figure 62 présente le taux de faux négatifs et de faux positifs pour différentes tailles n d'échantillon lorsque $\sigma_{ij}^{o} = 0.1$. A partir de $n = 50$, la courbe commence à converger. Ainsi, lorsque $\sigma_{ij}^{o} = 0.1$, on fixe n à 50, car augmenter la taille des échantillons au-delà de 50 n'augmente que très faiblement les performances du test. On remarque que le taux de faux positifs et de faux négatifs est faible, il est toujours inférieur à 2% lorsque $n \geq 30$.

La figure 63 présente, lorsque $\sigma_{ij}^{o} = 0{,}5$, le taux de faux négatifs et de faux positifs pour différentes tailles n d'échantillon. A partir de $n = 250$, la courbe converge, il est donc inutile de prendre des échantillons de taille supérieure à 250 lorsque $\sigma_{ij}^{o} = 0.5$. En comparant cette figure à la précédente, on remarque que le taux de faux négatifs et positif est supérieure lorsque $\sigma_{ij}^{o} = 0.5$ que lorsque $\sigma_{ij}^{o} = 0.1$. Cependant, il reste raisonnable puisqu'il est toujours inférieur à 7% lorsque $n \geq 250$.

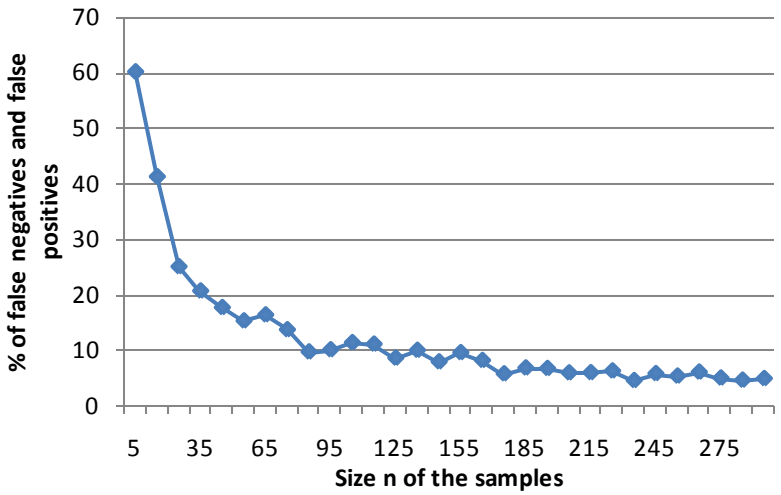


Figure 63 : Pourcentage de faux négatifs et de faux positifs lorsque les échantillons sont de taille n et que l'écart-type de la loi normale tronquée en 0 et en 1 générant les échantillons est de 0.5 ($\sigma_{ij}^{o} = 0.5$).

La figure 64 présente, lorsque $\sigma_{ij}^{o} = 0.9$, le taux de faux négatifs et de faux positifs pour différentes tailles n d'échantillon. A partir de $n = 300$, la courbe commence à converger. Ainsi, lorsque $\sigma_{ij}^{o} = 0.9$, on fixe $n = 300$. Le taux de faux négatifs et positifs est supérieur lorsque $\sigma_{ij}^{o} = 0.9$ que lorsque $\sigma_{ij}^{o} = 0.5$ ou $\sigma_{ij}^{o} = 0.1$. Cependant, il reste raisonnable puisqu'il est d'environ 10% lorsque $n \geq 300$.

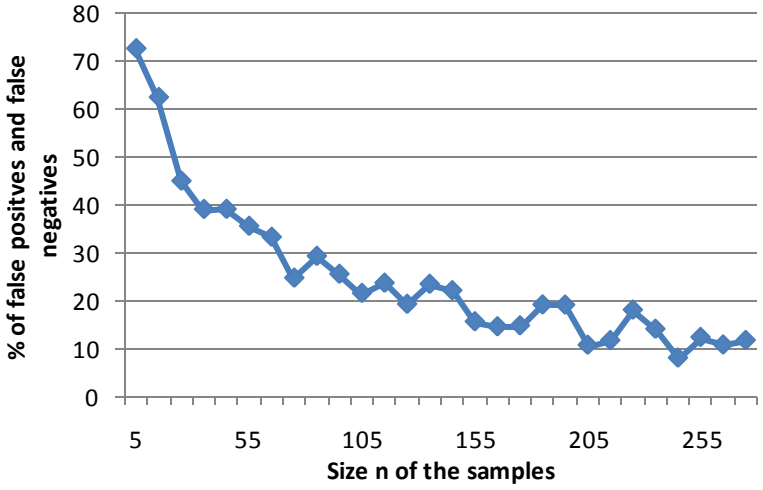


Figure 64 : Pourcentage de faux négatifs et de faux positifs lorsque les échantillons sont taille n et que l'écart-type de la loi normale tronquée en 0 et en 1 générant les échantillons est de 0.9 ($\sigma_{ij}^{o} = 0,9$).

Les figures 65, 67 et 69 présentent le pourcentage d'acceptation par le KS-test de l'hypothèse notée H_0 que les deux échantillons suivent la même distribution pour différentes valeurs de μ_{ij}^{o} et p, lorsque, respectivement, $\sigma_{ij}^{o} = 0{,}1$ et $n = 50$, $\sigma_{ij}^{o} = 0{,}5$ et $n = 250$ et $\sigma_{ij}^{o} = 0.9$ et $n = 300$. Lorsque le KS-test accepte H_0 alors que les échantillons du test ont été générés avec $p \geq 0.1$, un faux négatif

est dédenché. Si, par contre le KS-test n'accepte pas H_0 alors que les réalisations des échantillons du test ont été générées avec un $p \leq 0.1$, un faux positif est dédenché. Ces figures permettent de visualiser la répartition des faux négatifs et faux positifs lorsque la taille de l'échantillon est optimum.

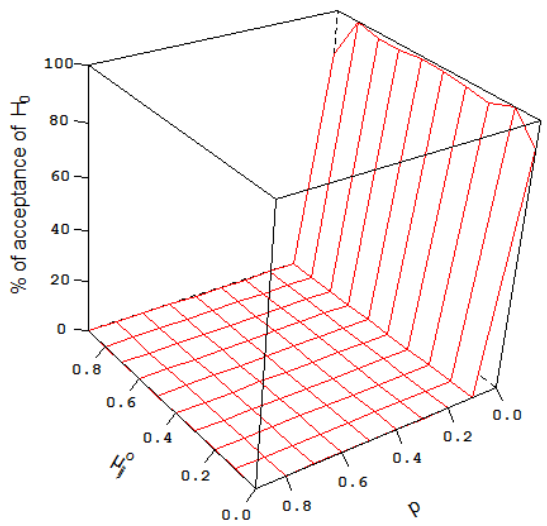


Figure 65 : Pourcentage d'acceptation de l'hypothèse H_0 lorsque σ_{ij}^o=0.1 et $n = 50$

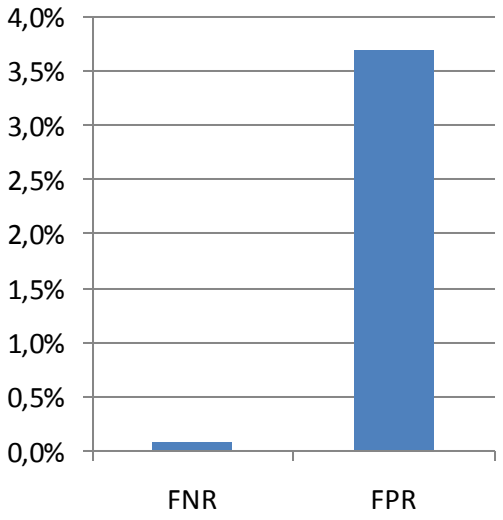


Figure 66 : Taux de faux négatifs et faux positifs lorsque σ_{ij}^o=0.1 et $n = 50$

D'après les résultats de la figure 65, lorsque σ_{ij}^o=0,1 et $n = 50$, si $p \leq 0{,}1$, l'hypothèse H_0 est acceptée 84% du temps, par contre, elle est presque toujours refusée si $p \geq 0{,}1$. Ainsi, le taux de vrais négatifs (TVN), c.à.d. le pourcentage de nœuds non mauvais détectés comme bons est de 86%, tandis que le taux de vrais positifs (TVP), c.à.d le pourcentage de nœuds mauvais détectés comme tels,est presque de 100%,. La figure 66 indique le taux de faux positifs (TFP) et de faux négatifs (TFN) générés lorsque σ_{ij}^o=0,1 et $n = 50$, ils sont tous les deux bas puisque inférieures à 4%. Les équations suivantes présentent les méthodes de calculs utilisées pour calculer le TVN, le TVP, le TFP et le TFN.

$$\text{TVN} = \frac{nb\ de\ noeuds\ non\ mauvais\ détectés\ comme\ tels}{nb\ de\ bon\ noeuds} \qquad \text{Équation 84}$$

$$TVP = \frac{nb\ de\ noeuds\ mauvais\ détectés\ comme tels}{nb\ de\ noeuds\ mauvais} \qquad \text{Équation 85}$$

$$TFN = \frac{nb\ de\ noeuds\ mauvais\ détectés\ comme\ non\ mauvais}{nb\ de\ \ noeuds\ détectés\ comme\ bons} \qquad \text{Équation 86}$$

$$TFP = \frac{nb\ de\ noeuds\ non\ mauvais\ détectés\ comme\ mauvais}{nb\ de\ noeuds\ détectés\ comme\ mauvais} \qquad \text{Équation 87}$$

D'après les résultats de la figure 67, lorsque σ_{ij}^o=0,5 et $n = 250$, si $p \leq 0{,}1$, l'hypothèse H_0 est acceptée 81% du temps, tandis qu'elle est presque toujours refusée si $p \geq 0{,}1$. Ainsi, le taux de vrais négatifs (TVN) est de 81% et le taux de vrais positifs (TVP)estde presque toujours 100% sauf lorsque

p=0,1 et $\sigma_{ij}^{o} = 0$. Les taux de faux positifs (TFP) et de faux négatifs (TFN) générés lorsque σ_{ij}^{o}=0,5 et $n = 250$, sont bas puisque inférieurs à 7%pour l'un et 1% pour l'autre (voir figure 68). En comparant les résultats obtenus avec $\sigma_{ij}^{o} = 0{,}1$ et ceux obtenus avec $\sigma_{ij}^{o} = 0{,}5$, on remarque que le TVP, le TFP, le TFN et le TFN augmentent très faiblement avec l'augmentation de σ_{ij}^{o}.

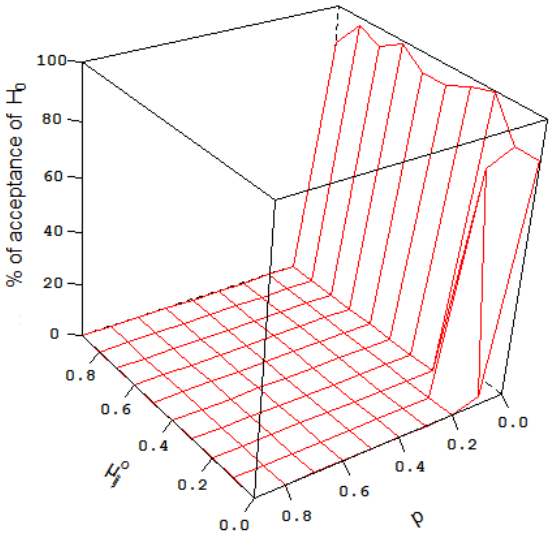


Figure 67 : Pourcentage d'acceptation de l'hypothèse H_0 lorsque σ_{ij}^{o}=0.5 et $n = 250$

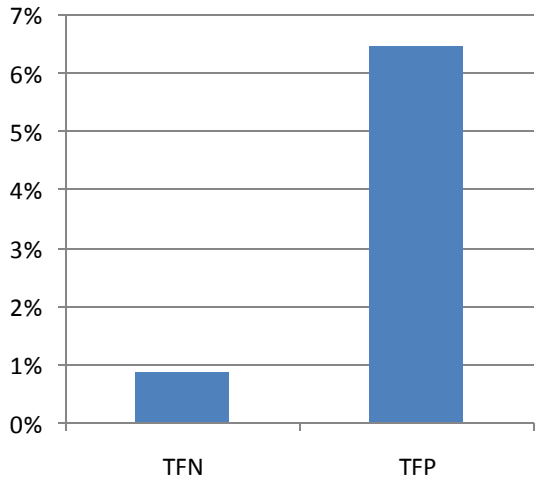


Figure 68 : Taux de faux négatifs et faux positifs lorsque σ_{ij}^{o}=0.5 et $n = 250$

D'après les résultats de la figure 69, lorsque σ_{ij}^{o}=0,9 et $n = 300$, si $p \leq 0{,}1$, l'hypothèse H_0 est acceptée 79% du temps tandis qu'elle est presque toujours refusée si $p \geq 0{,}1$. Le taux de vrais négatifs (TVN) est de 79% et le taux de vrais positifs (TVP) estde presque toujours 100%. Les taux de faux positifs (TFP) et de faux négatifs (TFN) générés lorsque σ_{ij}^{o}=0,5 et $n = 300$, sont tous bas puisque inférieurs à 10%pour l'un et 1% pour l'autre (voir figure 70).

Les résultats de l'évaluation prouvent que les modules 1 et 2 de notre IDS possèdent un taux de faux négatifs et de faux positifs faibles pour des tailles d'échantillon raisonnables (voir tableau 10) et peuvent ainsi détecter les mauvais nœuds en présence de perte de paquets sur les liens. Les résultats montrent également que plus la perte de paquets possède un écart-type faible plus les performances de nos modules sont bonnes pour des petites tailles d'échantillons (voir tableau 10).

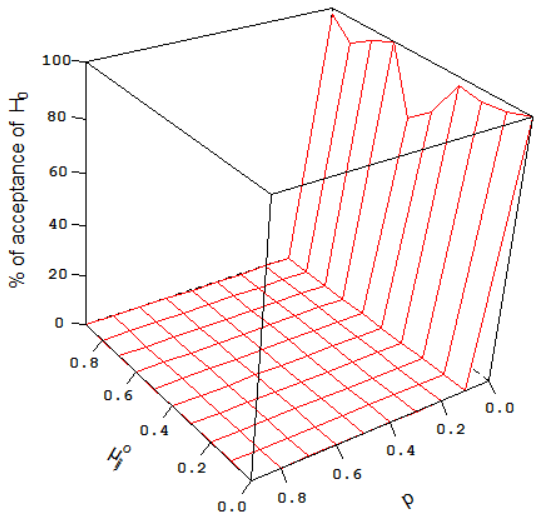


Figure 69 : Pourcentage d'acceptation de l'hypothèse H_0 lorsque σ_{ij}^o=0.9 et $n = 300$

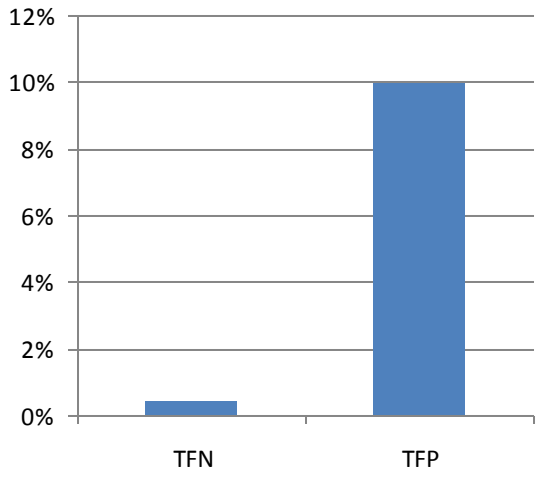


Figure 70 : Taux de faux négatifs et faux positifs lorsque σ_{ij}^o=0.9 et $n = 300$

Tableau 10 : Performances des modules 1 et 2 selon l'écart-type de la perte de paquets

	Taille n de l'échantillon	Pourcentage de vrais négatifs	Pourcentage de faux positifs	Pourcentage de faux négatifs
σ_{ij}^o=0 ,1	50	84%	0.1%	3.6%
σ_{ij}^o=0 ,5	250	81%	0.7%	6.5%
σ_{ij}^o=0 ,9	300	79%	0.5%	6.4%

6.4.2. Evaluation du troisième module de l'IDS de notre système de confiance

Le troisième module de notre IDS a pour but de détecter les mauvais nœuds qui modifient leur comportement dans le suivi des paquets de données.

A chaque fois qu'il obtient une réalisation de la variable aléatoire X_{ij}^o, le troisième module lance la méthode CUSUM. Cette dernière déclenche une alerte si elle détecte une modification de la distribution de la variable aléatoire X_{ij}^o par rapport à celle attendue. Le but de l'évaluation de ce module est de vérifier si CUSUM déclenche bien une alerte lorsque les réalisations de la V.A. X_{ij}^o ne suivent plus la distribution attendue.

Pour évaluer ce module dans une large variété de situations, on effectue le test suivant avec différentes valeurs de μ_{ij}^o, σ_{ij}^o et p. Via le logiciel R et pour une combinaison de σ_{ij}^o, de p et μ_{ij}^o, on

génère 100 réalisations de X_{ij}^{o}. Lorsque $p \geq 0.1$, ces réalisations représentent celles d'un mauvais nœud j et lorsque $p \leq 0.1$, ces réalisations représentent celles d'un nœud j non mauvais. On compare ensuite, via CUSUM, chaque réalisation avec le modèle normale X_{ij}^{o} qui est celui d'une loi normale tronquée de moyenne μ_{ij}^{o}, et d'écart-type σ_{ij}^{o}. Puis, on calcule le pourcentage d'alertes lancées parCUSUM. Une réalisation déclenche une alerte, si CUSUM considère qu'elle ne suit pas le modèle de référence ; le module estime alors que le nœud j est mauvais. Cette alerte est un faux positif si les réalisations ont été générées avec un paramètre $p \leq 0.1$. Si la méthode CUSUMne lance pas d'alerte suite au traitement d'une réalisation alors qu'elle a été générée avec p≥ 0.1, alors le module considère que le nœud j est non mauvais et génère un faux négatif puisqu'il ne détecte pas le mauvais comportement de j.

Pour la validation, les paramètres de CUSUM ont été fixés par rapport aux recommandations de Montgomery (Montgomery 2009), le décalage δ à détecter est fixé à 1, l'intervalle H de décision à $5 * \sigma_{ij}^{o}$ et la valeur de référence Kest fixée à $0.5 * \sigma_{ij}^{o}$.

La figure 71 présente le pourcentage d'alertes lancés par CUSUM lorsque $\sigma_{ij}^{o} = 0.1$ pour différentes valeurs de $p$etμ_{ij}^{o}. On remarque que CUSUM lance presque toujours une alerte lorsque $p \geq 0{,}1$ et rarement lorsque $p \leq 0{,}1$. Ainsi, les pourcentages de faux positifs et de faux négatifs sont très faibles, moins de 3% (voir figure 72).

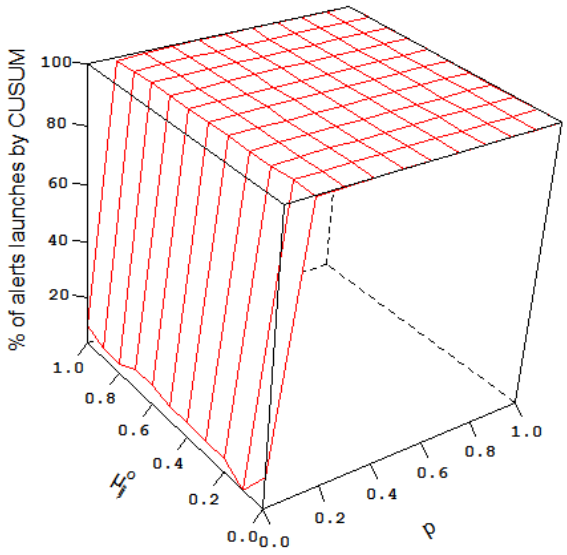


Figure 71 : Pourcentage d'alertes ave CUSUM lorsque σ_{ij}^{o}=0.1

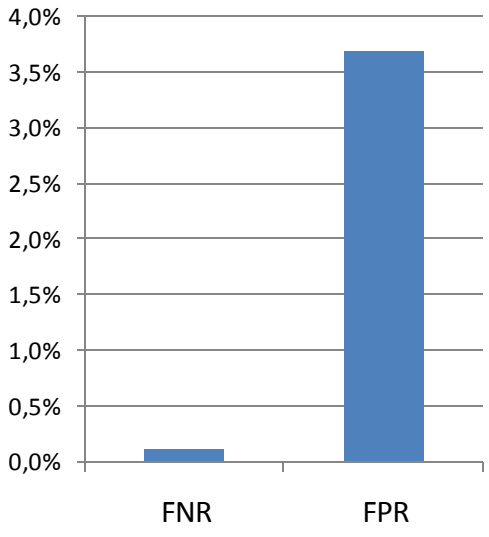


Figure 72 : Taux de faux négatifs et faux positifs du module 3 lorsque σ_{ij}^{o}=0.1

La figure 73 présente le pourcentage d'alertes lancées par CUSUM, lorsque $\sigma_{ij}^{o} = 0.5$, pour différentes valeurs de p et μ_{ij}^{o}. On remarque que, lorsque $p \leq 0{,}1$, CUSUM lance aucune alerte, le

nombre d'alertes augmentent progressivement jusqu'à $= 0.2$. A partir de $p = 0.2$, CUSUM lance pour toutes les réalisations des alertes. On remarque donc que CUSUM détecte les mauvais nœuds mais avec moins de précisions lorsque $\sigma_{ij}^{o} = 0.5$ que lorsque $\sigma_{ij}^{o} = 0.1$; un mauvais nœud est détecté avec une probabilité proche de 100 pourcent que s'il abandonne au moins 20% des paquets. Les pourcentages de faux positifs et de faux négatifs restent cependant faibles, inférieurs à 11% (voir figure 74).

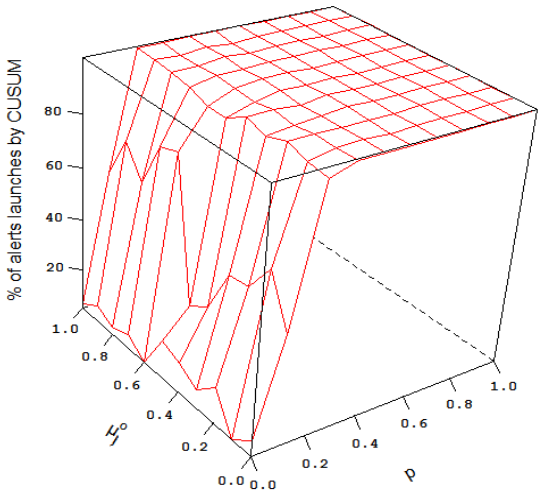


Figure 73 : Pourcentage d'alertes ave CUSUM lorsque σ_{ij}^{o}=0,5

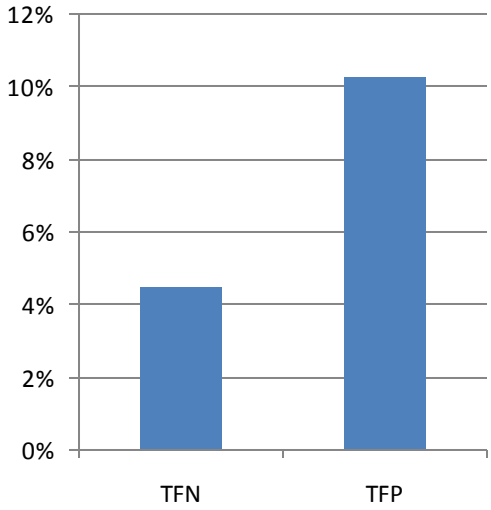


Figure 74 : Taux de faux négatifs et faux positifs du module 3 lorsque σ_{ij}^{o}=0,5

La figure 75 présente le pourcentage d'alertes lancées par CUSUM lorsque $\sigma_{ij}^{o} = 0.9$ pour différentes valeurs de p et μ_{ij}^{o}. Lorsque $p \leq 0{,}1$ CUSUM lance peu d'alertes, le nombre d'alertes augmentent progressivement, jusqu'à $p = 0.3$. A partir de $p = 0.3$, CUSUM lance presque pour toutes les réalisations une alerte. Ainsi, le module détecte les mauvais nœuds mais, avec moins de précisions lorsque $\sigma_{ij}^{o} = 0.9$ que lorsque $\sigma_{ij}^{o} = 0.5$. En effet, un mauvais nœud est détecté avec une probabilité proche de 100% que s'il abandonne au moins 30% des paquets,lorsque $\sigma_{ij}^{o} = 0.9$, alors qu'il est détecté à 100 % s'il abandonne 20% des paquets lorsque $\sigma_{ij}^{o} = 0.5$.Les pourcentages de faux positifs et de faux négatifs restent cependant faibles, inférieurs à 11% (voir figure 76).

Le module 3 permet de détecter les mauvais nœuds dans la majorité des situations avec un faible pourcentage de faux positifs et de faux négatifs (voir Tableau 11). Les résultats de l'évaluation montrent que plus l'écart-type de la perte est grande et moins bonne est la précision de détection. En effet, le module détectera, lorsque l'écart-type est grand, les mauvais nœuds que s'ils abandonnent un plus grand nombre de paquets. Cependant, la finesse de détection restent dans

toutes les situations raisonnable puisqu'à l'écart-type quasi maximum de 0.9, le module détecte presque à 100% tous les nœuds mauvais s'ils abandonnent au moins 30% des paquets.

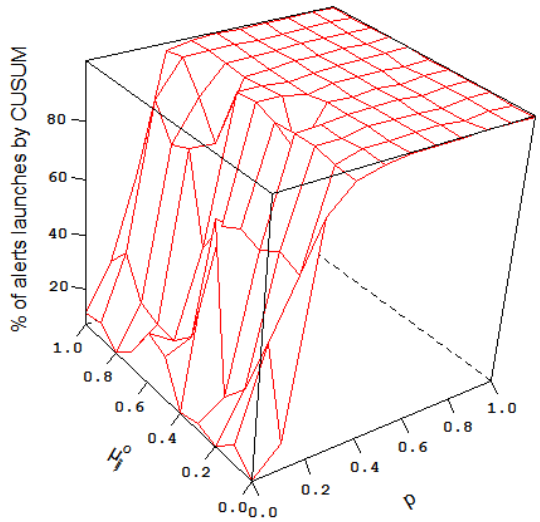


Figure 75 : Pourcentage d'alertes avec CUSUM lorsque σ_{ij}^{o}=0.9

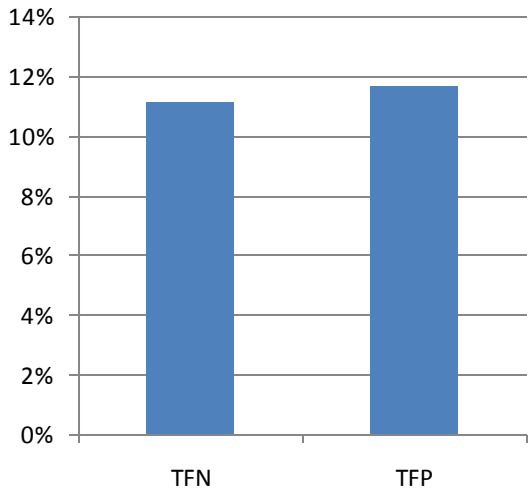


Figure 76 : Taux de faux négatifs et faux positifs du module 3 lorsque σ_{ij}^{o}=0.9

Tableau 11 : Table des résultats de validation du troisième module de l'IDS de notre système de confiance

	Pourcentage de faux négatifs	Pourcentage de faux positifs
$\sigma_{ij}^{o} = 0.9$	0,1%	2,7%
$\sigma_{ij}^{o} = 0.5$	4,7%	10,2%
$\sigma_{ij}^{o} = 0.9$	10,3%	11,7%

6.4.3. Evaluation comparative de notre système de confiance avec un système de confiance existant

Nous avons comparé notre solution avec un système de confiance assez classique basé sur l'IDS Watchdog et décrit dans (Sen 2010). Nous appellerons, par la suite la solution avec laquelle nous comparons notre proposition, solution de référence. Pour effectuer cette comparaison, nous avons simulé ces deux solutions sur le simulateur de réseau à événements discrets ns2. Le but de ces simulations est de montrer que notre solution assigne à chaque nœud du réseau une valeur de confiance qui reflète son comportement (mauvais ou bon) et non la qualité de ses liens contrairement à d'autres systèmes existants et qu'il est ainsi plus à même de détecter les mauvais nœuds.

6.4.3.1. *Protocole de l'évaluation comparative*

Notre solution a été comparé sur deux topologies mesh, une topologie en croix et une topologie maillée possédant chacune un seul portail d'accès représenté en rouge sur les figures 77 et 78. Chaque nœud, à part le portail, envoie un flux à 20 kbit/s sur le réseau et possède un numéro (voir figures Figure 77 : Topologie en croix77et 78). Le nœud numéro i commence à envoyer son flux à la $i + 1^{ème}$ seconde.

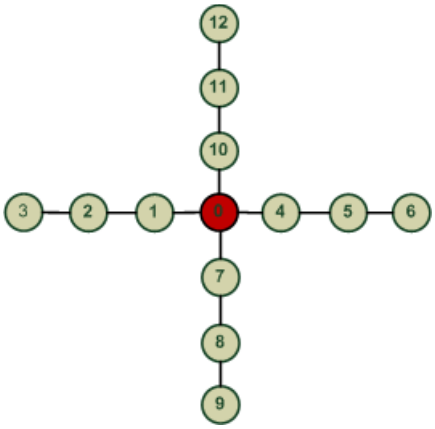


Figure 77 : Topologie en croix

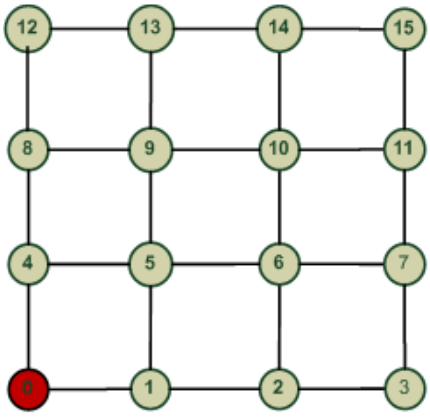


Figure 78 : Topologie maillée

Lors de la simulation, nous avons supposé que les liens du réseau mesh avaient été sélectionnés tels que la moyenne de la perte de paquets sur un lien ainsi que sa variance associée n'est pas trop importante. Ainsi, comme le montre le tableau 12 des paramètres de simulation, chaque lien est associé à une perte de paquets moyenne choisit aléatoirement selon une loi uniforme de paramètre 0.5 et 0, il en est de même pour la variance de cette moyenne. Les paramètres de simulation sont, à part ceux associés à la perte de paquets sur un lien, assez classiques et sont présentés sur le tableau.

Tableau 12 : Tableau des paramètres de simulation

Niveau	Paramètre	Valeur
Propagation du signal	Modèle Two-ray-ground	
Taux de paquets perdus sur un lien	Moyenne du taux choisie aléatoirement selon une loi uniforme	[0, 0.5]
	Déviation standard du taux choisie selon une loi uniforme	[0, 0.5]
Modèle d'interférence	Modèle d'interférence additif	
Physique	Fréquence	54 Mbit/s
	PLCP préambule	20
Couche MAC	CSMA/CA	
Poids de l'historique	β	0.5

Chaque topologie mesh possède un nœud malveillant dont le numéro est 7 pour le réseau à topologie en croix et le numéro est 4 pour le réseau à topologie maillée. Le nœud malveillant abandonne volontairement un certain pourcentage p de paquets de données sur l'ensemble des paquets qu'il reçoit avec succès.

6.4.3.2. Résultats de la simulation

Afin de ne pas surcharger les figures, nous avons, pour chaque simulation, représenté seulement la confiance de certains nœuds au cours du temps dont entre autre le nœud malveillant. Les figures Figure 79 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=179 et 80 présentent la valeur de confiance de quatre nœuds (nœud 1, 4, 7 et 10) du réseau dans le temps pour, respectivement, la solution de référence et notre solution lorsque le nœud malveillant (nœud 7) abandonne 100% des paquets de données qu'il reçoit.

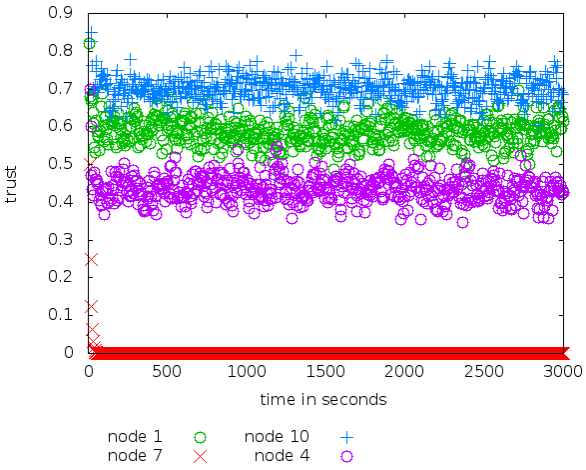


Figure 79 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=1

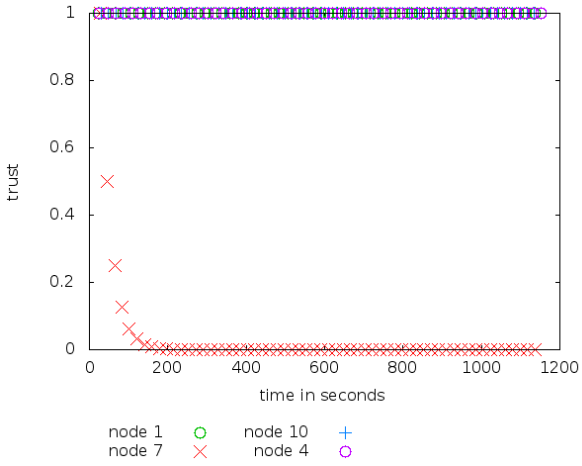


Figure 80 : Confiance des nœuds dans un réseau mesh à topologie en croix avec notre solution lorsque p =1

D'après les figures précédentes, notre solution, en considérant la perte de paquets sur les liens, assigne aux bons nœud (nœuds 1, 10 et 4) la valeur maximale de confiance qui est de 1, tandis que la solution de référence assigne à ces derniers une valeur de confiance qui est bien plus basse et qui reflète logiquement la qualité de leur liens. Avec notre solution comme avec celle de référence, la valeur de confiance du mauvais nœud, décroit rapidement dans le temps pour devenir nulle. Cependant, cette dernière décroit moins rapidement avec notre système. En effet, notre système nécessite que les nœuds observent un plus grand nombre de paquets avant d'assigner une valeur de confiance à ces voisins.

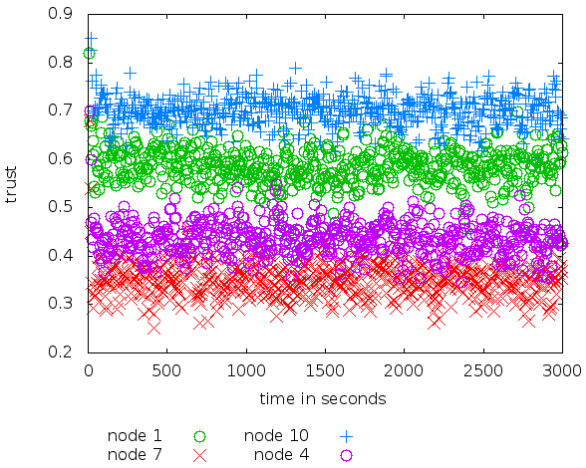


Figure 81 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=0.5

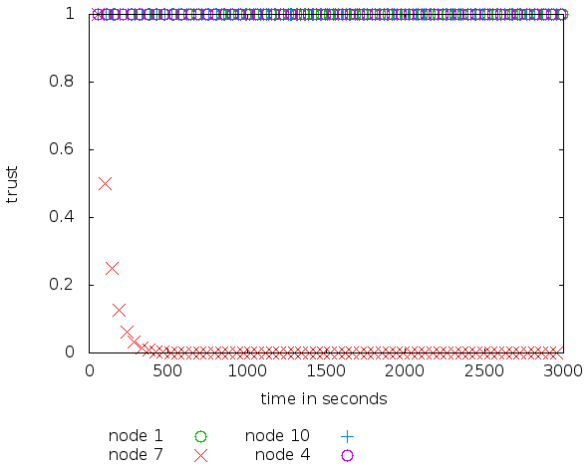


Figure 82 : Confiance des nœuds dans un réseau mesh à topologie en croix avec notre solution lorsque p =0.5

Les figures 81 et 82 présentent la valeur de confiance de quatre nœuds (nœud 1, 4, 7 et 10) du réseau dans le temps dans un réseau à topologie en croix pour, respectivement, la solution de référence et notre solution lorsque cette fois ci le nœud malveillant (nœud 7) abandonne 50% des paquets de données qu'il reçoit. Notre solution comme précédemment accorde aux bons nœuds la valeur de confiance maximale, tandis que le système de référence leur assigne une valeur de confiance qui reflète la qualité de leur liens. De plus, les deux solutions assignent une valeur de confiance basse au mauvais nœud, cependant cette valeur est très proche de celle de certains bons nœuds en ce qui concerne le système de référence alors que, avec notre solution, cette dernière est très différente de celle des bons nœuds ; elle devient quasi nulle avec le temps. Cependant, notre solution nécessite un certain temps avant d'assigner faible valeur de confiance au mauvais nœud, il lui faut 50 secondes pour lui assigner une valeur de confiance de 0.5 et 400 secondes pour une valeur de confiance proche de 0.

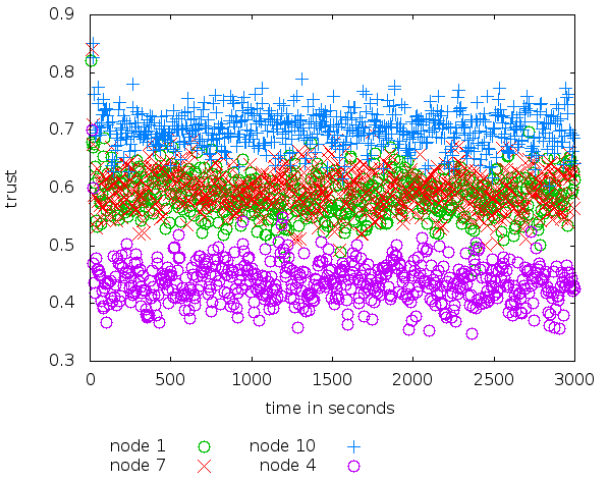


Figure 83 : Confiance des nœuds dans un réseau mesh à topologie en croix avec la solution de référence lorsque p=0.2

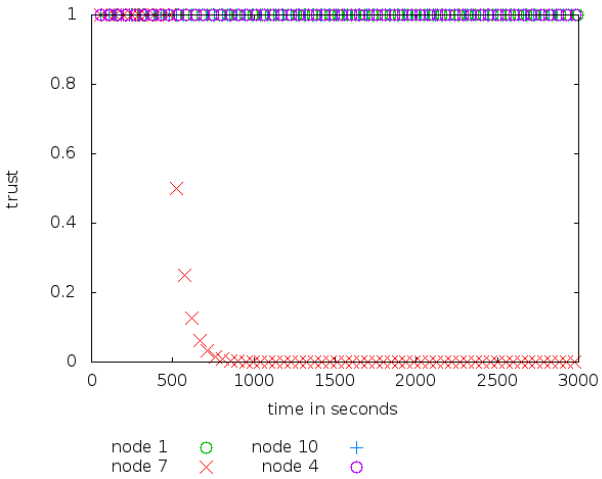


Figure 84 : Confiance des nœuds dans un réseau mesh à topologie en croix avec notre solution lorsque p =0.2

Les figures 83 et 84 présentent la valeur de confiance de quatre nœuds du réseau dans le temps dans un réseau à topologie en croix pour, respectivement, la solution de référence et notre solution lorsque le nœud malveillant (nœud 7) abandonne 20% des paquets de données qu'il reçoit. La solution de référence assigne au mauvais nœud une valeur de confiance qui est supérieure ou égale à celles de certains bons nœuds (nœud 1 et nœud 4). Ainsi, elle ne permet nullement de distinguer les bons des mauvais nœuds. Par contre, notre solution les distingue clairement car elle accorde aux bons nœuds une valeur de confiance maximale et au mauvais nœud une valeur de confiance qui dans le temps décroit jusqu'à la valeur nulle. Cependant, notre solution nécessite d'attendre un certain temps avant que mauvais nœud se distingue des autres (environ 500 s).

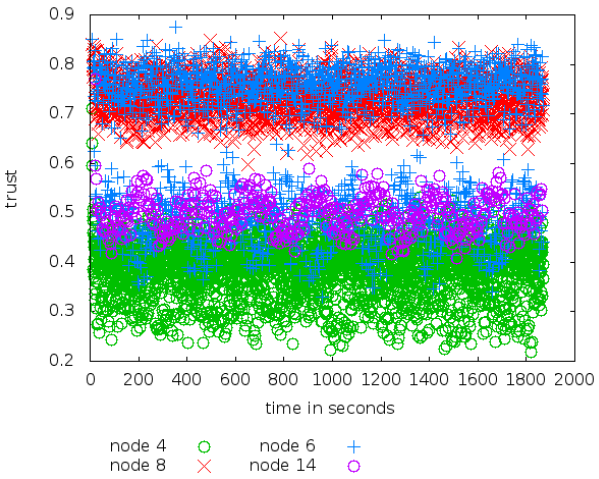


Figure 85 : Confiance des nœuds dans un réseau mesh à topologie maillé avec la solution de référence lorsque p=0.2

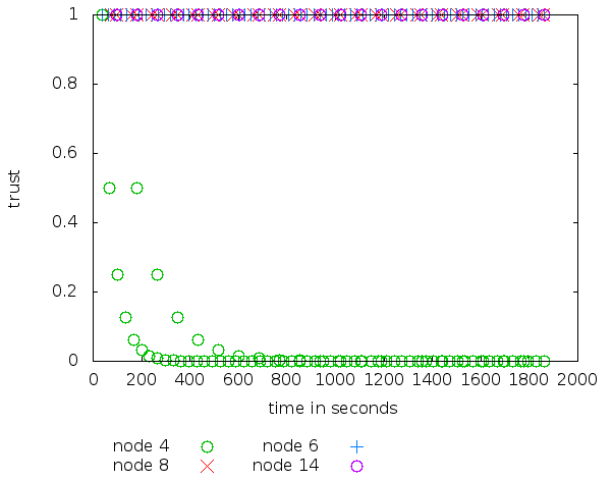


Figure 86 : Confiance des nœuds dans un réseau mesh à topologie maillée avec notre solution lorsque p =0.2

Nous avons également comparé notre système à celui de référence dans un réseau mesh à topologie maillée (voir figure 85 et 86). Le nœud malveillant est cette fois ci le nœud 4, il abandonne 20% des paquets qu'on lui envoie. On remarque deux courbes de confiance pour le nœud 4 via notre système de confiance. Cela s'explique par le fait que deux nœuds différents évaluent le nœud 4, le nœud 4 faisant suivre des paquets qu'il peut recevoir de deux différents voisins. Comme précédemment, le nœud malveillant via le système de confiance de référence a une valeur assez proche des bons nœuds ce qui n'est pas le cas avec notre système. Cependant, on remarque que notre solution assigne au nœud 4 une faible confiance qu'au bout d'un certain temps, environ 200 secondes pour la première courbe. Ainsi, via une topologie maillée le temps nécessaire pour détecter le mauvais nœud est bien plus court (deux fois plus) qu'avec une topologie en croix lorsque p=0.2. On peut expliquer ce phénomène par le fait que, probablement, le nœud 4 a plus de paquets a faire suivre que le nœud 7 sur la topologie en croix et donc que le voisin qui l'observe atteint plus rapidement le nombre d'observations nécessaires au déclenchement de l'algorithme de réputation.

Les résultats de ces simulations prouvent que notre système permet de mieux différencier les mauvais des bons nœuds que les systèmes de confiance traditionnels qui ne considèrent pas la perte de paquets sur les liens. La différence de qualité entre notre solution et ceux existants est d'autant plus grande que les mauvais nœuds abandonnent peu de paquets. Cependant, le temps d'assignation de la confiance à un mauvais nœud peut être avec notre solution assez long, car il nécessite que le voisin observe une large quantité de données avant de lancer l'algorithme de réputation.

6.5. Conclusion

Dans ce chapitre, nous avons présenté un nouveau système de confiance pour un réseau mesh sans fil qui a pour originalité de considérer plusieurs types de mauvais comportements ainsi que la perte de paquet sur les liens. Afin de considérer plusieurs types de mauvais comportements, chaque nœud possède, un système de surveillance composé de plusieurs modules. Chaque module d'un nœud lui permet de collecter des informations sur ses voisins sur un type de comportement malveillant. Ainsi, le niveau de confiance attribué à un nœud permet à ce dernier de savoir si l'un de ses voisins possède l'un des types de mauvais comportement surveillé par l'un des modules. De plus, chaque module considère le bruit sur les liens, il peut ainsi faire la différence entre un mauvais nœud et un nœud bon dont les liens ont une probabilité de perte de paquets importante. Pour se faire, chaque module compare le comportement de suivi des paquets d'un nœud observé avec son comportement de suivi « normale » des paquets ; le comportement normal de suivi de paquets d'un nœud est celui

que doit avoir le nœud lorsqu'il n'est pas mauvais. Si le comportement observé est le même que celui attendu alors le nœud est considéré comme bon sinon, il est considéré comme mauvais. Cette comparaison est effectuée selon les modules sur le suivi de différents types de paquets ; acquittements, paquets de route, paquet de données et aussi avec plusieurs méthode de comparaison : CUSUM ou le test de Kolmogorov-Smirnov. Notre solution a été validée par simulation sur R, elles montrent que pour chaque module de notre système de détection le taux de faux positifs et faux négatifs reste toujours raisonnable. Dans des contextes défavorables, ce taux ne dépasse jamais 12% et le nombre d'information nécessaires à une détection restent acceptables (environ 300 réalisations). Dans des contextes favorables, le taux de faux négatifs et le taux de faux positifs peuvent être très faibles (environ 3%) et le nombre d'information nécessaires à une détection peu important (environ 150 réalisations). De plus, la simulation de notre solution prouve que cette dernière assigne une valeur de confiance qui reflète le comportement des nœuds et non la qualité de leurs liens contrairement à certains systèmes existants. Il permet également de mieux différencier les mauvais des bons nœuds même lorsque le mauvais nœud abandonne peu de paquets. Cependant, la détection d'un mauvais nœud peut être avec notre solution un peu longue car elle nécessite pour assigner une valeur de confiance à l'un de ses voisins un nœud doit demander à son voisin de faire suivre une large quantité de données.

Ainsi notre solution remplit ses objectifs de limitation des faux positifs et négatifs déclenchés par *Watchdog*, de détecter les mauvais nœuds et d'assigner une valeur de confiance qui reflète réellement le comportement (mauvais ou bon) d'un nœud.

Partie 2. Conclusion

Les réseaux mesh permettent d'étendre la zone de couverture de l'Internet à des zones encore isolés. Cependant, leur déploiement est freiné par leurs nombreuses vulnérabilités qui permettent à des mauvais nœuds aussi bien internes ou externes au réseau de nuire à son bon fonctionnement.

Le premier chapitre de cette partie présente un état de l'art sur les systèmes de confiance qui permettent d'attribuer un valeur de confiance à chaque nœud selon si ce dernier a un mauvais comportement ou non, afin de pouvoir par la suite isoler, remplacer ou l'inciter les mauvais nœuds à devenir bon. Ce chapitre présente à la fois les objectifs, le fonctionnement et les limites de ces systèmes. Ces solutions généralement ne surveillent qu'un type de mauvais comportement pour attribuer une valeur de confiance à un nœud, cependant il existe plusieurs types de mauvaises actions. De plus, la valeur de confiance est basée sur des observations, or ces dernières peuvent être biaisées par la perte de paquets sur les liens. Un bon nœud peut avoir une confiance basse car ses nœuds voisins ne peuvent observer les paquets que ce dernier fait suivre à cause du bruit sur le lien.

Afin de pallier les limites des solutions existantes, nous proposons un nouveau système de confiance basé sur un mécanisme de surveillance comprenant trois modules dont chacun surveille un type de mauvaise action et considère le bruit sur les liens. Afin de différentier un mauvais d'un bon nœud qui souffrirait de pertes de paquets sur ces liens, ces modules comparent la distribution de la perte de paquets « normale » ou attendue sur un lien par rapport à celle observée avec des outils statistiques tels que CUSUM ou le test de Kolmogorov-Smirnoff. Pour se faire, nous avons modélisé la perte de paquets « normales » sur un lien, c.à.d. quant aucun des nœuds composant le lien n'abandonne volontairement ou à cause de défaillance des paquets, par une loi normale. La validité de notre modèle a été effectuée vie des expérimentations sur plusieurs liens. Par la suite, chaque module, est relié à un système de calcul de la confiance qui permet à un nœud d'attribuer à chacun de ses voisins une valeur de confiance ; il suffit qu'un nœud possède un seul type de mauvais comportement parmi les trois surveillés par notre solution pour qu'il possède une mauvaise réputation.

Notre solution a été ensuite validée via R et par simulation avec ns2. Les résultats montrent que chaque module détecte avec un faible nombre de faux positifs et de faux négatifs, moins de 12%, dans le pire des cas, si les observations collectées sur un nœud sont celles d'un nœud mauvais ou non. De plus, comme le montre le résultat des simulations, notre système est capable de détecter les bons nœuds des mauvais nœuds et d'assigner aux nœuds une valeur de confiance qui reflète réellement leur comportement.

Chapitre 7. Conclusion générale et perspectives

Dans ce chapitre nous concluons la thèse suivant deux parties: la première partie résume notre contribution et montre comment elle répond à notre problématique de départ et la seconde partie introduit les axes d'amélioration et d'extension de cette thèse.

7.1. Contribution

Les réseaux mesh sans fil sont des réseaux complémentaires aux réseaux sans fil avec infrastructure. Ce sont des réseaux privilégiés pour des zones isolées ou en voie de développement car ils peuvent être déployés à faible coûts. Ils sont également adaptés pour des zones accidentées ou lors de situation de crise car ils sont rapides à installer et permettent d'interconnecter de nombreux réseaux et équipements aux technologies différentes. Cependant, le déploiement des ces réseaux est limité par leurs nombreuses vulnérabilités et leurs faibles performances en termes de qualité de service. En effet, ces réseaux ne respectent que rarement les contraintes des flux car ils possèdent une faible capacité utile ainsi qu'un taux de perte de paquets important. De plus, leurs caractéristiques facilitent l'accès au réseau à de potentiels attaquants qui peuvent dès lors l'endommager. Un routeur mesh défaillant peut également nuire aux performances de son réseau car ce dernier est basé sur un routage multi-saut où chaque nœud est un point potentiel de rupture de chemins.

L'objectif de départ de cette thèse est de proposer une architecture de qualité de service sécurisée permettant de résoudre les problèmes actuels des réseaux mesh (non-respect des exigences des flux, perte de paquets, faible capacité, vulnérabilité aux mauvais nœuds) afin de pouvoir à terme offrir à de nombreux utilisateurs la possibilité de profiter de leurs avantages.

7.1.1. Contrôle d'admission avec planification des liens

Afin de répondre à la problématique de qualité de service des réseaux mesh, nous avons proposé une solution de contrôle d'admission dont l'originalité repose sur l'intégration d'un mécanisme dynamique de planification de liens. Notre solution repose sur un cadre théorique : le problème d'admission d'un flux est représenté sous forme d'un programme linéaire à variables binaires que nous prouvons être NP-complet. Une nouvelle méthode de calcul du délai des flux est également proposée. Afin de résoudre le problème d'admission d'un flux, nous présentons un algorithme

modifié de la méthode de séparation évaluation de Dakin. La méthode de Dakin permet de résoudre en un temps polynomial des problèmes NP-complets lorsque l'instance du problème est raisonnable. Notre version modifiée de cette méthode calcule itérativement, pour chaque demande d'admission d'un flux, une planification respectant la bande passante requise du flux dont le délai est optimum. Tant que le délai engendré par la planification est supérieur au délai maximum demandé par le flux, l'algorithme est réitéré et ce jusqu'à ce qu'il n'y ait plus de planification possible ou que le nombre d'itération soit supérieur à un certain seuil. L'algorithme est intégré au contrôle d'admission. Chaque nœud souhaitant émettre un flux envoie une requête de demande d'admission pour le flux à la destination. La destination effectue le contrôle d'admission et retourne, si le flux est accepté, la planification du flux obtenue avec notre algorithme. La validation de notre solution a été effectuée avec ns2 sur plusieurs topologies de réseau mesh. Elle prouve que notre système respecte les contraintes des flux en termes de délai et de débit, et qu'il permet de diminuer la perte de paquets et d'augmenter la capacité utile du réseau.

7.1.2. Nouveau système de confiance avec détection des mauvais nœuds dans un réseau mesh

Le contrôle d'admission permet d'assurer la qualité de service d'un réseau mesh tant que ce dernier n'est pas attaqué. Il est donc indispensable de réfléchir également au contrôle d'admission en termes de sécurité pour assurer la fluidité des services. Afin de détecter les mauvais nœuds du réseau nous proposons, dans cette thèse, un nouveau système de confiance permettant d'éviter les mauvais nœuds lors du routage. L'originalité de ce travail réside dans une détection de multiples mauvais comportements considérant la perte de paquets sur les liens afin que chaque nœud puisse assigner à chacun de ses voisins une valeur de confiance reflétant son comportement réel. Notre solution est basée sur un système de surveillance composé de trois modules, chaque module surveillant un type de mauvais comportement. Le premier surveille si les voisins d'un nœud ne tentent pas de s'inscrire sur un maximum de route pour par la suite les perturber. Le second vérifie si les voisins d'un nœud acquittent tous leurs paquets pour dissimuler des mauvaises actions. Le dernier vérifie si les voisins d'un nœud modifient leur comportement et plus particulièrement le taux de paquets qu'ils abandonnent ou font suivre. Afin que les observations collectées d'un nœud sur un voisin ne soient pas biaisées par la perte de paquets, chaque module vérifie avec un outil statistique, comme par exemple le test de Kolmogorov Smirnov ou la méthode CUSUM) si la perte de paquets observée suit une distribution « normale », c.à.d. si la distribution de la perte de paquet est celle d'un nœud non

mauvais. Ces modules ont pour but de détecter de multiples mauvais comportements même en présence de perte de paquets sur les liens, cette dernière pouvant être très importante. La validation de notre solution prouve que chaque module différencie avec un faible taux de faux positifs et de faux négatifs les bons des mauvais nœuds, ce taux pouvant monter jusqu'à 12% dans les contextes les plus défavorables. Comme le montre les résultats de simulations de notre solution et contrairement aux systèmes de confiance basés sur Watchdog qui ne considèrent pas la perte de paquets sur les liens, notre solution attribue à un nœud une valeur de confiance à partir de données non biaisées qui reflète ainsi son comportement et non la qualité de ses liens.

7.2. Perspectives

Le travail mené dans cette thèse est une première étape vers la réalisation d'une solution de contrôle d'admission sécurisée. Il ouvre plusieurs perspectives de travaux futurs. Dans la continuité de ce travail nous souhaitons, à court terme, approfondir l'évaluation de nos travaux, à moyen terme associer notre contrôle d'admission avec notre système de confiance et à long terme, dissocier deux types de confiance, la confiance en un nœud en termes de qualité de service et la confiance en un nœud en termes de sécurité.

7.2.1. Etudes d'évaluation approfondie

Nous envisageons d'améliorer l'évaluation de nos deux propositions par les actions suivantes :

- l'approfondissement de l'évaluation de notre contrôle d'admission avec planifications des liens en comparant ses résultats avec d'autres contrôles d'admission existants,
- l'évaluation de notre système de confiance via des simulations sur ns2 et comparaisons des résultats avec celles obtenus avec des solutions existantes de système de confiance.

7.2.2. Intégration de notre système de confiance au contrôle d'admission

A moyen terme, nous souhaiterions intégrer le paramètre de confiance dans le contrôle d'admission d'un flux. Ainsi, lorsque la destination recevra la demande d'admission d'un flux pour un chemin

donné, elle prendra sa décision d'admission d'un flux selon le niveau de confiance des nœuds intermédiaires et des paramètres de qualité de service demandés par le flux tel que le délai et la bande passante. Ainsi, le contrôle d'admission refusera tout flux dont le nœud source est mauvais ou dont le chemin intègre un mauvais nœud ; les mauvais nœuds seront ainsi isolés du réseau. Le réseau sera ainsi protégé des mauvais nœuds qui pourront à plus long terme être remplacés par l'administrateur réseau.

7.2.3. Différenciation de la confiance

A plus long terme nous souhaiterions également attribuer à un nœud deux confiances, afin d'améliorer l'architecture de qualité de service sécurisée. La première confiance, que l'on appellerait la confiance de QoS refléterait si le nœud a confiance dans son voisin pour faire suivre correctement ses données que ce voisin soit malveillant ou non. Par exemple, un mauvais nœud envoyant de faux messages de routage mais transférant tous les paquets de données aurait une bonne valeur de confiance de QoS, car ce serait un nœud intermédiaire sûr pour transmettre les paquets de données. La seconde confiance, que l'on appellerait la confiance de sécurité refléterait si le nœud est mauvais ou non. Ainsi lors de l'admission d'un flux, un flux ne pourrait être admis que si son nœud source n'est pas mauvais, le but étant alors d'inciter le nœud à agir correctement. De plus, un flux sera accepté le long d'un chemin que si la confiance de QoS des nœuds du chemin est bonne et que ce chemin respecte les exigences du flux.

L'idée de cette extension serait d'intégrer la sécurité à la qualité de service et de ne pas rejeter un nœud mauvais du chemin des flux s'il peut offrir aux flux un meilleur débit qu'un nœud honnête et non défaillant. En interdisant les nœuds mauvais à émettre sur le réseau, on inciterait également les mauvais nœuds à agir correctement. Ces derniers pourront être remplacés si, à long terme, leur comportement n'évolue pas. Cette solution permettrait d'assurer la qualité de service des flux admis, de détecter les mauvais nœuds, d'augmenter la capacité du réseau, d'isoler les mauvais nœuds et pourrait inciter ces derniers à agir correctement.

Publications

Revues internationales :

- J.Dromard, L. Khoukhi, R.Khatoun, An Efficient Admission Control Model Based on Dynamic Links Scheduling in Wireless Mesh Networks, EURASIP Journal on Wireless Communications and Networking, 2014

- J.dromard, R.khatoun, L.Khoukhi, A computation reputation system for Wireless Mesh Networks Considering Packet Loss Dynamic, International Journal of Information Security, 2014, (en cours de review)

Conférences internationales avec acte :

- J. Dromard, R. Khatoun, L. Khoukhi , A Watchdog Extension Scheme Considering Packet Loss for a Reputation System in Wireless Mesh Network, 20th International Conference on Telecommunications (ICT)-6-8 May 2013, Casablanca-Maroc

- J.Dromard, L.Khoukhi, R. Khatoun, An Admission Control Scheme Based on Transmission Scheduling for Wireless Mesh networks, Performance Analysis and Enhancement of Wireless Networks (PAEWN), 25-28 Mars 2013, Barcelone-Espagne

- J. Dromard, L. Khoukhi, R. Khatoun, An Admission Control Scheme Based on Links' Activity Scheduling for Wireless Mesh Networks. ADHOC-NOW 2012, p. 399-412, Belgrade- Serbie

- Nabet, R. Khatoun, L. Khoukhi, J. Dromardet D. Gaiti. Towards Secure Route Discovery Protocol in MANET. Global Information Infrastructure Symposium (GIIS), 4-6 aout 2011, Da Nang – Vietnam

Bibliographie

Abouaissa, Abdelhafid, Mohamed-el-Amin Brahmia, et Pascal Lorenz. «Increasing end-to-end fairness over IEEE 802.11e-based wireless mesh networks.» *International Journal of Communication Systems*, 2013: 1--12.

Aguayo, Daniel, John Bicket, Sanjit Biswas, Glen Judd, et Robert Morris. «Link-level measurements from an 802.11b mesh network.» *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004: 121--132.

Akyildiz, Ian, Xudong Wang, et Weilin Wang. «Wireless mesh networks: a survey.» *Computer Networks*, 2005: 445 - 487.

Aoun, B, et R Boutaba. «Max-Min Fair Capacity of Wireless Mesh Networks.» Édité par IEEE Computer Society. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 2006: 21-30.

Atelin, Philippe. *Wi-Fi - Réseaux sans fil 802.11.* Édité par Eni. Eyrolles, 2008.

Bakhouya, Mohamed, Jaafar Gaber, et Pascal Lorenz. «An adaptive approach for information dissemination in Vehicular Ad hoc Networks.» *Network and Computer Applications*, 2011: 1971-1978.

Balas, E, et R Jeroslow. «Canonical cuts on the unit hypercube.» *SIAM Journal of Applied Mathematics*, 1972: 61-69.

Basseville, Michèle, et Igor Nikiforov. *Detection of abrupt changes: theory and application.* Édité par Inc. Prentice-Hall. Prentice-Hall, Inc., 1993.

Belbachir, Redouane, Mekkakia Maaza Zoulikha, Ali Kies, et Bernard Cousin. «Bandwidth reservation in mobile adhoc networks.» *IEEE Wireless Communications and Networking Conference (WCNC)²*, 2012: 2608-2613.

Ben Salem, Naouel, et Jean-Pierre Hubaux. «Securing Wiireless Mesh Networks.» *IEEE Wireless Communications* 13, n° 2 (2006).

Boukerche, A, B Turgut, N Aydin, Mohammad Ahmad, Ladislau Bölöni, et D and Turgut. «Routing protocols in ad hoc networks: A survey.» *Computer Networks*, 2011: 3032 - 3080.

Brar, Gurashish, Douglas Blough, et Paolo Santi. «Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks.» *Proceedings of the 12th annual international conference on Mobile computing and networking* (ACM), 2006: 2--13.

Buchegger, Sonja, et Jean-Yves Le Boudec. «Performance analysis of the CONFIDANT protocol.» *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002: 226--236.

Calafate, Carlos Miguel Tavares, José Olivier, Juan Carlos Cano, Pietro Manzoni, et Manuel P. Malumbres. «A distributed admission control system for MANET environments supporting multipath routing protocols.» *Microprocessors and Microsystems*, 2007: 236--251.

Cappanera, Paola, Luciano Lenzini, Alessandro Lori, Stea Giovanni, et Gigliola Vaglini. «Link scheduling with end-to-end delay constraints in Wireless Mesh Networks.» *World of Wireless, Mobile and Multimedia Networks Workshops*, 2009: 1-9.

Chen, Yan, Toni Farley, et Nong Ye. «QoS Requirements of Network Applications on the Internet.» Édité par IOS Press. *Inf. Knowl. Syst. Manag.*, january 2004: 55--76.

Cheng, Xiaolin, Mohaprata Prasant, et Sung-Ju Lee. «Performance evaluation of video streaming in multihop wireless mesh networks.» *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2008: 57-62.

Clausen, T, et P Jacquet. *Optimized Link State Routing Protocol.* IETF RFC 3626, 2003.

Conti, M., et S Giordano. «Multihop Ad Hoc Networking: The Reality.» *Communications Magazine, IEEE*, 2007: 88-95.

Cornuéjols, Gérard. *Valid inequalities for mixed integer linear programs.* Springer-Verlag, 2008.

Dakin. «A tree-search algorithm for mixed integer programming problems.» *The Computer Journal*, 1965: 250-255.

Dankin, R. J. «A tree search algorithm for mixed integer linear programming problems.» *Computer Journal*, 1965: 250±255.

Dantzig. *Origins of the simplex method.* A History of Scientific Computing, 1990.

Das, Saumitra, Dimitrios Koutsonikolas, et Charlie Hu. «Measurement-based characterization of a wireless mesh network.» *Handbook of Wireless Mesh and Sensor Networking. McGraw-Hill International*, 2008.

David, Johnson, et Maltz David. «Dynamic source routing in ad hoc wireless networks.» *Mobile Computing* (Kluwer Academic Publishers), 1996: 153--181.

Deb, K, A Pratap, S Argawal, et T. Meyarivan. «A fast and elitist multiobjective genetic algorithm: NSGA-II.» Édité par IEEE Press. *Trans. Evol. Comp*, 2002: 182--197.

Djukic, Petar, et Shahrokh Valaee. «Delay Aware Link Scheduling for Multi-Hop TDMA Wireless Networks.» *IEEE/ACM Transactions on Netwrking* 17, n° 3 (Juin 2009): 870-883.

Dromard, Juliette, Lyes Khoukhi, et Rida Khatoun. «An Admission Control Scheme Based on Links' Activity Scheduling for Wireless Mesh Networks.» *ADHOC-NOW*, 2012: 399-412.

Dromard, Juliette, Lyes Khoukhi, et Rida Khatoun. «An Admission Control Scheme Based on Transmission Scheduling for Wireless Mesh networks.» *Performance Analysis and Enhancement of Wireless Networks*, 2012.

Dromard, Juliette, Rida Khatoun, et Lyes Khoukhi. «A Watchdog Extension Scheme Considering Packet Loss for a Reputation System in Wireless Mesh Network.» *International Conference on Telecommunications (ICT)*, 2013.

Garay, Michael, et Johnson David. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman & Co., 1990.

Gore, A.D., A Karandikar, et S Jagabathula. «On High Spatial Reuse Link Scheduling in STDMA Wireless Ad Hoc Networks.» *Global Telecommunications Conference*, 2007: 736--741.

Gore, A.D., et A Karandikar. «Link Scheduling Algorithms for Wireless Mesh Networks.» *Communications Surveys Tutorials, IEEE* 13, n° 2 (2011): 258-273.

Goussevskaia, Olga, Yvonne Anne Oswald, et Rogert Wattenhofer. «Complexity in geometric SINR.» Édité par ACM. *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, 2007: 100--109.

Goussevskaia, Olga, Yvonne Anne Oswald, et Rogert Wattenhofer. «Complexity in geometric SINR.» Édité par ACM. *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, 2007: 100--109.

Guéret, Christelle, Christian Prins, et Marc Seveaux. *Programmation linéaire.* Paris: Eyrolles, 2003.

Guimares, Rafael, Cerda Llorenz, José Barcelo, Jorge Gardia, Michael Voorhaenb, et Chris Blondia. «Quality of service through bandwidth reservation on multirate ad hoc wireless networks .» *Ad Hoc Networks*, 2009: 388-400.

Gupta, P, et P.R. Kumar. «The capacity of wireless networks.» *IEEE transactions on information theory*, mars 2000: 388-404.

Hanzo, Lajos, et Rahim Tafazolli. «Admission Control Schemes for 802.11-Based Multi hop Mobile Ad Hoc Networks: A survey.» *IEEE Communications survey and Tutorials*, 2009.

Hartenstein, H., et K P Laberteaux. «A tutorial survey on vehicular ad hoc networks.» *Communications Magazine, IEEE* 46 (2008): 164 - 171.

Hasswa, Ahmed, Mohammad Zulkernine, et Hossam Hassanein. «Routeguard: an intrusion detection and response system for mobile ad hoc networks.» *International Conference on Wireless and Mobile Computing, Networking and Communications.*, 2005: 336--343.

IEEE. «Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification.» n° 802.11. 1997.

I-STAMS: Scalable, modular tactical mesh wireless network for classified and unclassified voice, video, and data. 2013. http://www.telos.com/secure-networks/wlan-solutions/tactical-mesh/.

Iyer, Aravind, Catherine Rosenberg, et Adita Karnik. «What is the right model for wireless channel interference?» Édité par IEEE Press. *IEEE Transactions on Wireless Communications* 8, n° 5 (2009): 2662--2671.

Jhaveri, R.H. «MR-AODV: A Solution to Mitigate Blackhole and Grayhole Attacks in AODV Based MANETs.» *Advanced Computing and Communication Technologies (ACCT)*, 2013: 254--260.

Jia, Fang, Yanqiu Zhou, Weibing Luo, et Bo Zhang. «Research and Application of WMN-Based Emergency Communication.» *Wireless Communications, Networking and Mobile Computing*, 2008: 1-3.

Jian, Chen, Jia Jie, et Wen Ying-you. «Efficient link scheduling for TDMA based WMN using multi-objective genetic algorithm.» *Bio-Inspired Computing: Theories and Applications (BIC-TA),*, 2010: 923--927.

Jiri, Fiala. «NP completeness of the edge precoloring extension problem on bipartite graphs.» *J. Graph Theory*, 2003: 156-160.

Jun, Jangeun, et M Sichitiu. «The nominal capacity of wireless mesh networks.» Édité par IEEE Press. *Wireless Commun.*, 2003: 8--14.

Jung, Jong-kwan, Ki-Yeol Ryu, et Byeong-h Roh. «Mission-critical packet transfer with explicit path selection in WMN-based tactical networks.» *Wireless Telecommunications Symposium (WTS), 2010*, 2010: 1--5.

Kapse, V.S., et U.N. Shrawanakar. «Survey of channel assignement schemes in wireless mesh network.» *Electronics Computer Technology (ICECT), 2011 3rd International Conference on* 3 (2011): 103--107.

Khoukhi, Lyes. *Gestion Intelligente de Qualite de Service Dans Les Reseaux Ad Hoc Mobiles Sans Fil.* Sherbrooke Canada: ProQuest, 2006.

Kim, Jihye, et Gene Tsudik. «SRDP: Secure route discovery for dynamic source routing in MANETs.» *Ad Hoc Netw.* 7, n° 6 (2009): 1097-1109.

Le Boudec, Jean-Yves, et Thirian Patrick. *Network calculus: a theory of deterministic queuing systems for the internet.* Berlin: Springer-Verlag, 2001.

Lee, Janghwan, Hyunsoo Yoon, et Ikjun Yeom. «Distributed Fair Scheduling for Wireless Mesh Networks Using IEEE 802.11.» *Vehicular Technology, IEEE Transactions on* 59, n° Vehicular Technology, IEEE Transactions on (2010).

Li, Wenjia, James Parker, et Anupam Joshi. «Security Through Collaboration and Trust in MANETs.» *Mob. Netw. Appl.*, 2012: 342--352.

Li, Yu. «A reputation system for wireless mesh network using multi-path routing protocol.» *Proceedings of the 30th IEEE International Performance Computing and Communications Conference*, 2011: 1-6.

Lohier, Stéphane, Yacine Ghamri-Doudane, et Guy Pujolle. «Link Available Bandwidth Monitoring for QoS Routing with.» *IFIP/IEEE International Conference on Management of Multimedia*, 37-48: 37--46.

Lu, Wang, Wu Kaishun, et M Hamdi. «Combating Hidden and Exposed Terminal Problems in Wireless Networks.» *IEEE Transactions on Wireless Communications*, 2012: 4204-4213.

Luo, L, M Gruteser, H Liu, et D Raychaudhuri. «A QoS Routing and Admission Control Scheme for 802.11 Ad Hoc.» *Proc. Int. Conf. Mobile Computing and Networking,*, 2006.

Maheshwari, Ritesh, Shweta Jain, et Samir Das. «A measurement study of interference modeling and scheduling in low-power wireless networks.» Édité par ACM. *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2009: 141--154.

Marti, Sergio, T Giuli, Kevin Lai, et Mary Baker. «Mitigating routing misbehavior in mobile ad hoc networks.» *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000: 255-265.

Marx, Daniel. «NP-completeness of list coloring and precoloring extension on the edges of planar graphs.» *Journal of Graph Theory*, 2005: 313-324.

Mccanne, S, S Floyd, et K Fall. «ns Network simulator.» *http://www.isi.edu/nsnam/ns/.*

Mobile Mesh Networks for Military, Defense and Public Safety. 2013. http://www.meshdynamics.com/military-mesh-networks.html.

Mogre, Parag, Matthia Hollick, et Ralf Steinmetz. «QoS in wireless mesh networks: challenges, pitfalls, and roadmap to its realization.» Édité par ACM. *Network and Operating System Support for Digital Audio and Video*, 2007.

Montgomery. *Introduction to statistical quality control.* 6. John Wiley & Sons, 2009.

Nandiraju, N., D Nandiraju, L Santhanam, He Bing, Wang Junfang, et D.P. Agrawal. «Wireless Mesh Networks: Current Challenges and Future Directions of Web-In-The-Sky.» *Wireless Communications, IEEE* 14 (2007).

Naouel Ben, Salem, et Jean-Pierre Hubaux. «Securing wireless mesh networks.» *Wireless Communication*, 2006: 50--55.

Ning, Peng, et Kun Sun. «How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols.» *Ad Hoc Netw.* 3, n° 6 (2005): 795--19.

One Laptop per children association. 2005. http://one.laptop.org/ (accès le juillet 10, 2013).

Ouni, A, H Rivano, et F Valois. «Capacity of Wireless Mesh Networks: Determining Elements and Insensible Properties.» *Wireless Communications and Networking Conference Workshops*, 2010: 1-6.

Pathak, P.H., et R Dutta. «A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks.» *Communications Surveys Tutorials, IEEE* 13, n° 3 (2011): 396-428.

Peppas, Nikolaos, et D Turgut. «A Hybrid Routing Protocol in Wireless Mesh Networks.» *Military Communications Conference, 2007. MILCOM 2007. IEEE*, 2007: 1--7.

Perkins, C, E Royer, et S Das. «Ad hoc On-Demand Distance Vector (AODV) Routing.» IETF RFC 3561, 2003.

Perkins, Charles, et Pravin Bhagwat. «Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers.» Édité par ACM. *SIGCOMM Comput. Commun. Rev.*, 1994: 234--244.

Portmann, M, et A.A. Pirzada. «Wireless Mesh Networks for Public Safety and Crisis Management Applications.» *Internet Computing, IEEE*, 2008: 18--25.

Rappaport, Theodore. *Wireless Communications: Principles and Practice.* Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

Reis, Charles, Ratul Mahajan, Maya Rodrig, David Wetherall, et John Zahorjan. «Measurement-Based Models of Delivery and Interference in Static Wireless Networks.» *Special Interest Group on Data Communication (SIGCOMM)*, 2006: 51--62.

Rezgui, J, A Hafid, et M. Gendreau. «A distributed admission control scheme for Wireless Mesh Networks.» *Broadband Communications, Networks and Systems, 2008. BROADNETS 2008. 5th International Conference on*, 2008: 594-601.

Safaei, Z, M Sabaei, et F Torgheh. «An efficient reputation-based mechanism to enforce cooperation in MANETs.» *Proceedings of the 4th international conference on Communications and information technology*, 2010: 33-38.

Savelsbergh, Martin. «Branch and Price: Integer Programming with Column Generation.» *Encyclopedia of Optimization*, 2009: 328-332.

Sen, Jaydip. «A Distributed Trust and Reputation Framework for Mobile Ad Hoc Networks.» *CoRR*, 2010.

Sharma, Gaurav, Ravi Mazumdar, et Ness Shroff. «On the complexity of scheduling in wireless networks.» Édité par ACM. *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, 2006: 227--238.

Shen, Qiang, Xuming Fang, Li Pan, et Fang Yuguang. «Admission Control Based on Available Bandwidth Estimation for Wireless Mesh Networks.» *Vehicular Technology, IEEE Transactions on*, 2009: 2519-2528.

Sichitiu, M.L. « Wireless mesh networks: opportunities and challenges.» *Proceedings of World Wireless Congress*, 2005.

Siddiqui, Muhammad Shoaib, et Chong Seon. «Security Issues in Wireless Mesh Networks.» *IEEE Computer Society*, 2007: 717-722.

Sierksma, Gerard, Peter van Dam, et G Tijssen. *Linear and integer programming : theory and practice.* New York: Dekker, 1996.

Sivakumar, D, B Suseela, et R Varadharajan. «A survey of routing algorithms for MANET.» *Advances in Engineering, Science and Management (ICAESM)*, 2012: 625-640.

Stallings, William. *Data and computer communications (8th edition).* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007.

Subramanian, V.G., et D.J. Leith. «Utility fairness in 802.11-based wireless mesh networks.» *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 2010: 961-968.

Suli, Zhao, et Raychaudhuri Dipankar. «Multi-Tier Ad Hoc Mesh Networks with Radio Forwarding Nodes.» Édité par IEEE. *GLOBCOM*, 2007: 1360-1364.

Tafazolli, Lajos Hanzo II and Rahim. «Admission Control Schemes for 802.11-Based Multi hop Mobile Ad Hoc Networks: A survey.» *IEEE Communications survey and Tutorials*, 2009.

Tan, Wee Lum, Peizhao Hu, et M Portmann. «Experimental evaluation of measurement-based SINR interference models.» *World of Wireless, Mobile and Multimedia Networks*, 2012: 1-9.

Team, R Development Core. «R: A Language and Environment for Statistical Computing.» Vienne: R Foundation for Statistical Computing, 2008.

Tsai, Jung-Fa, Lin Ming-Hua, et Hu Yi-Chung. «Finding multiple solutions to general integer linear programs.» *European Journal of Operational Research*, 2006: 802-809.

Tzu-Jane, Tsai, et Chen Ju-Wei. «IEEE 802.11 MAC Protocol over Wireless Mesh Networks: Problems and Perspectives.» Édité par IEEE Computer Society. *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)* 2 (2005): 60--63.

Wan, Peng-Jun, Xiaohua Xu, et Ophir Frieder. «Shortest Link Scheduling with Power Control under Physical Interference Model.» Édité par IEEE Computer Society. *2010 Sixth International Conference on Mobile Ad-hoc and Sensor Networks*, 2010: 74-78.

Wan, Yao, Jia Xiaohua, et F Frances. «Maximum Independent Set of Links under Physical Interference Model.» Édité par Springer. *Wireless Algorithms, Systems, and Applications, 4th International Conference*, n° 978-3-642-03416-9 (2009): 169-178.

Wang, Dongbin, Mingzeng Hu, Hui Zhi, et Jianwei Ye. «Cooperation Enforcement Among Selfish Nodes in Ad Hoc Networks under Noise.» *Information Technology Journal*, 2009: 757-763.

Wifi-alliance. «IEEE Standard for Information Technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Am.» *IEEE Std 802.11s-2011*, 2011: 1-372.

Xiang, Zeng, Bagrodia Rajive, et Gerla Mario. «GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks.» *in Workshop on Parallel and Distributed Simulation*, 1998: 154--161.

Xu, Kaixin, M Gerla, et Sang Bae. «How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks.» *Global Telecommunications Conference*, 2002: 72-76.

Yang, Yaling, et Robin Kravets. «Contention-Aware Admission Control for Ad Hoc Networks.» Édité par IEEE Educational Activities Department. *IEEE Transactions on Mobile Computing*, 2005: 363-377.

Yick, Jennifer, Biswanath Mukherjee, et Dipak Ghosal. «Wireless sensor network survey.» *Computer Networks*, 2008: 2292 - 2330.

Yu, Han, Zhiqi Shen, Chunyan Miao, Cyril Leung, et Dusit Nyato. «A Survey of Trust and Reputation Management Systems in Wireless Communications.» *Proceedings of the IEEE* 98, n° 10 (October 2010): 1755-1772.

Zemin, Wu, et Qiu Zhenglun. «A Survey on Directional Antenna Networking.» *Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2011: 1-4.

Liste des abréviations

CA	Contrôle d'admission
ACA	Admission Control Algorithm
AODV	Ad-hoc On-demand Distance Vector
BRAWN	Bandwidth reservation in ad hoc wireless networks
BP	Bande Passante
CACP	Contention-aware Admission Control Protocol
CUSUM	Cumulative Sum Control Chart
CPU	Central Processing Unit
CTS	Clear To Send
DACME	Distributed Admission for Manets Environment
DCF	Distributed Coordination Function
DSDV	Destination-Sequenced Distance-Vector Routing
DSL	DSL-Digital Subscribe Line
DSR	Dynamic Source Routing
FIFO	First In First Out
HWMP	Hybrid Wireless Mesh Protocol
IDS	Intrusion Detection System ou système de detection d'intrusions
KS-test	Kolmogorov-Smirnov
MAC	Medium Access Control
MANETs	Mobile Ad-hoc Networks
MKVMP	Maximum K-Valid Matching Problem

PLCP	Physical Layer Convergence Procedure
PLVB	ProgrammationLinéaire en Variables Binaires
RREQ	Route Request
RREP	Route Reply
RTS	Ready To Send
SAE	Simultaneous Authentication of Equals
SINR	Signal to Interference plus Noise Ratio
TDMA	Time Division Multiple Access
TTL	Time To Live
TSF	Timing Synchronization Function
V.A.	Variable aléatoire
VANETs	Vehicular Ad-hoc Network
VoIP	Voix sur IP
WiMAX	Worldwide Interoperability for Microwave Access
WMN	Wireless Mesh Networks

Juliette DROMARD

Doctorat : Réseaux, Connaissances, Organisations

Année 2013

Vers une solution de contrôle d'admission sécurisée dans les réseaux mesh sans fil

Les réseaux mesh sans fil (Wireless Mesh Networks-WMNs) sont des réseaux facilement déployables et à faible coût qui peuvent étendre l'Internet dans des zones où les autres réseaux peuvent difficilement accéder. Cependant, plusieurs problèmes de qualité de service (QoS) et de sécurité freinent le déploiement à grande échelle des WMNs. Dans cette thèse, nous proposons un modèle de contrôle d'admission (CA) et un système de réputation afin d'améliorer les performances du réseau mesh et de le protéger des nœuds malveillants. Notre système de CA vise à assurer la QoS des flux admis dans le réseau en termes de bande passante et de délai tout en maximisant l'utilisation de la capacité du canal. L'idée de notre solution est d'associer au contrôle d'admission une planification de liens afin d'augmenter la bande passante disponible. Nous proposons également un système de réputation ayant pour but de détecter les nœuds malveillants et de limiter les fausses alertes induites par la perte de paquets sur les liens du réseau. L'idée de notre solution est d'utiliser des tests statistiques comparant la perte de paquets sur les liens avec un modèle de perte préétabli. De plus, il comprend un système de surveillance composé de plusieurs modules lui permettant détecter un grand nombre d'attaques. Notre CA et notre système de réputation ont été validés, les résultats montrent qu'ils atteignent tous deux leurs objectifs.

Mots clés : contrôle d'admission des connexions - système de communication sans fil - accès multiple par répartition dans le temps – réseaux d'ordinateurs, mesures de sûreté.

Towards a Secure Admission Control in a Wireless Mesh Networks

Wireless mesh networks (WMNs) are a very attractive new field of research. They are low cost, easily deployed and high performance solution to last mile broadband Internet access. However, they have to deal with security and quality of service issues which prevent them from being largely deployed. In order to overcome these problems, we propose in this thesis two solutions: an admission control with links scheduling and a reputation system which detects bad nodes. These solutions have been devised in order to further merge into a secure admission control. Our admission control schedules dynamically the network's links each time a new flow is accepted in the network. Its goal is to accept only flows which constraints in terms of delay and bandwidth can be respected, increase the network capacity and decrease the packet loss. Our reputation system aims at assigning each node of the network a reputation which value reflects the real behavior of the node. To reach this goal this reputation system is made of a monitoring tool which can watch many types of attacks and consider the packet loss of the network. The evaluations of our solutions show that they both meet their objectives in terms of quality of service and security.

Keywords: network performance (telecommunication) - wireless communication system - time division multiple access - computer networks, security measures.