



THÈSE

Pour l'obtention du grade de

DOCTEUR

Spécialité : *Informatique*

Présentée et soutenue le *Date (JJ/MM/AAAA)* par :

HASSAN FAOUZI

Amélioration de la Performance des Protocoles de Routage pour une
Meilleure QoS dans les Réseaux Mobiles Ad Hoc

JURY

.....
.....
.....
.....
.....
HICHAM MOUNCIF	PH FP-Beni Mellal	Directeur de thèse
MOHAMED LAMSAADI	PH FP-Beni Mellal	Directeur de thèse

Centre d'Études Doctorales :

Sciences et Techniques

Formation Doctorale :

Mathématiques et Physique Appliquées (MPA)

Laboratoire :

Laboratoire de Modélisation des Ecoulements et des Transferts

Avant Propos

Les travaux présentés dans ce mémoire ont été effectués au Laboratoire de Recherche en Modélisation des Ecoulements et des transferts, à la Faculté des Sciences et Technique de Beni Mellal sous la direction de Mr Mohamed LAMSAADI et Mr Hicham MOUNCIF.

C'est avec la plus profonde gratitude que je souhaite remercier Mr Mohamed LAMSAADI, Professeur d'enseignement supérieur à la faculté polydisciplinaire beni mellal. Je lui exprime ma profonde gratitude de m'avoir accepté au sien du laboratoire LAMET et d'avoir accepté de diriger mes travaux de recherche. Je le remercie infiniment pour tout l'intérêt, le suivi et l'implication inconditionnels portés à cette thèse, inlassablement durant toutes ces années et malgré son emploi du temps chargé. J'exprime ici ma profonde gratitude à son égard et l'estime respectueuse que je lui porte.

Merci aussi à Mr Hicham MOUNCIF, Professeur d'enseignement supérieur à la faculté polydisciplinaire beni mellal. Je tiens à le remercier pour ces années de soutien, pour ses précieux conseils scientifiques et pour son aide et sa capacité à simplifier les problèmes rencontrés dans le cadre du travail. Je le remercie infiniment pour sa compréhension dans toutes les circonstances et pour son habilité de toujours trouver les mots d'encouragement, qui ne manquaient pas d'augmenter ma motivation.

Je remercie tout particulièrement les membres de mon jury de thèse, qui ont accepté de juger ce travail et de participer au jury.

Je remercie également mes parents et toute ma famille pour leur encouragement et leur soutien tout au long de mes années d'études et je leur dédie cette thèse.

Mes remerciements vont enfin à toute personne qui a contribué de près ou de loin à l'élaboration de ce travail.

Abstract

Abstract — A MANET (Mobile Ad hoc networks) is a particular case of wireless network where each node can directly join its neighbors by the use of its radio interface and without the need for a pre-existing infrastructure. Their deployment is manifested in several areas of applications. However, these applications face several obstacles due to the dynamic topology, since the hosts are mobile and connected between them in an arbitrary way. Mobile Ad hoc networks therefore have a number of challenges which impose new requirements for design of routing protocols efficient. These include the constraint of energy consumption and management difficult to access to the medium.

The consumption of energy is a constraint very critical because generally the nodes consume their own energy in routing of the data for other nodes. Of this fact, the consumption of energy at the level of the nodes has a great influence on the life of the entire network.

Because of the limitation of the radio coverage and the sharing of the wireless interface, if a mobile phone emits a signal, all mobile in the area will not be able to transmit data simultaneously and the signal is received by all nodes, to powers variables, the noise so add to the interference, which increases the rate of error of the connection and decreases therefore the bandwidth.

To resolve these problem the work of this thesis are interested in the proposal of new methods of calculation of routing metrics (delay, size of the queue, residual energy batteries and hop count) there focusing our studies on the access method which multichannel reduce the contention as well as the probability of collision where several transmissions can be carried out simultaneously in the same area of transmission without overlap, thus greatly improving the throughput and the delay in the networks. We introduce new methods to anticipate the values of these metrics using dynamic and genetic algorithms through the development of an architecture Cross-layer taking into account the physical properties and MAC for routing.

Keywords : Mobile Ad hoc Networks (MANETs), network simulator (NS-2), Ad-hoc On-demand Distance Vector (AODV), Destination Sequenced Distance Vector (DSDV) , Multi-Channel Multi-Interface (MCMCI), Quality of Service (QoS), Markov chain Gilbert-Elliot model.

Résumé

Résumé — Un MANET (Mobile Ad hoc Networks) est un cas particulier de réseau sans fil où chaque nœud peut directement joindre ses voisins par l'utilisation de son interface radio et sans nécessité d'infrastructure préexistante. Leur déploiement se manifeste dans plusieurs domaines d'applications. Cependant, ces applications font face à plusieurs obstacles dus à la topologie dynamique, puisque les hôtes sont mobiles et connectés entre eux de manière arbitraire. Les réseaux mobiles Ad hoc possèdent donc un certain nombre de défis qui imposent de nouvelles exigences pour concevoir des protocoles de routage performants. Citons la contrainte de consommation d'énergie et la gestion difficile à l'accès au médium.

La consommation d'énergie est une contrainte très cruciale puisque généralement les nœuds consomment leur propre énergie en routant des données pour d'autres nœuds. De ce fait, la consommation d'énergie au niveau des nœuds a une grande influence sur la durée de vie du réseau en entier.

En raison de la limitation de la couverture radio et le partage de l'interface sans fil, si un mobile émet un signal, tous les mobiles dans un certain périmètre ne pourront pas transmettre de données simultanément et le signal est réceptionnée par tous les nœuds, à des puissances variables, les bruit donc s'ajoutent aux interférences, ce qui augmente le taux d'erreur de la liaison et diminue par conséquent la bande passante.

Pour pallier à ces problèmes les travaux de cette thèse s'intéressent à la proposition de nouvelles méthodes de calcul des métriques de routage (délai de bout-en-bout, taille de la file d'attente, énergie résiduelle des batteries et nombre de saut) on focalisant nos études sur la méthode d'accès multicanaux qui réduise la contention ainsi que la probabilité de collision où plusieurs transmissions peuvent s'effectuer simultanément dans une même zone de transmission sans chevauchement, améliorant ainsi considérablement le débit et le délai dans les réseaux mobiles Ad hoc. Nous introduisons de nouvelles méthodes pour anticiper les valeurs de ces métriques en utilisant des algorithmes dynamiques et génétiques à travers le développement d'une architecture cross-layer tenant compte des propriétés physiques et MAC pour le routage.

Mots clés : Réseaux sans fil, Réseaux mobiles Ad hoc, économie d'énergie, algorithmes dynamiques, algorithmes génétiques, Routage Ad hoc, qualité de service (QoS), MIMC.

Table des matières

Table des sigles et acronymes	viii
Introduction générale	1
1 Réseaux Mobiles Ad Hoc	4
1.1 Introduction	4
1.2 Caractéristiques des réseaux mobiles Ad Hoc	5
1.3 Domaines d'application	6
1.4 Stratégies de routages des données dans les réseaux mobiles Ad Hoc	6
1.5 Classification des protocoles de routages	7
1.5.1 Protocoles proactifs	7
1.5.2 Protocoles réactifs	7
1.5.3 protocoles Hybrides	8
1.6 Présentation de quelques protocoles de routage Ad Hoc	8
1.6.1 Ad-hoc On Demand Distance Vector (AODV)	8
1.6.1.1 Découverte de chemins	8
1.6.1.2 Entretien des routes	9
1.6.2 Dynamic Source Routing (DSR)	10
1.6.2.1 Découverte de routes	11
1.6.2.2 Maintenance de routes	12
1.6.3 Optimized Link State Routing (OLSR)	13
1.6.3.1 Découverte des nœuds de voisinage	13
1.6.3.2 Construction de la table topologique	14
1.6.3.3 Calcul des routes	14
1.6.4 Destination Sequenced Distance Vector (DSDV)	14

1.6.5	Zone Routing Protocol (ZRP)	15
1.7	Etat de l'art sur la QoS dans les réseaux mobiles Ad Hoc	16
1.7.1	Approches au niveau de la couche physique et MAC	16
1.7.1.1	Approches au niveau de la couche physique	17
1.7.1.2	Approches au niveau de la couche MAC	22
1.7.2	Approches au niveau de la couche réseau	30
1.7.2.1	Ticket based probing	31
1.7.2.2	First Fast Second Reliable	31
1.7.2.3	EA_AODV	32
1.7.2.4	Link Prediction	33
1.7.2.5	Energy saving Multi-path AODV (EM-AODV)	33
1.7.2.6	ELMA	35
1.7.2.7	LEACH	36
1.7.2.8	Hybrid Energy-Efficient Distributed Clustering	37
1.8	Conclusion	37
2	Evaluation de performances	38
2.1	Introduction	38
2.2	Présentation de network simulator NS2	39
2.3	Modèles de propagation radio	39
2.3.1	Modèle de propagation en espace libre (Free space model)	40
2.3.2	Modèle de propagation Two-ray ground model	40
2.3.3	Modèle de propagation Shadowing	41
2.4	Modélisation d'un canal radio 802.11	41
2.5	Modèle de mobilité	42
2.5.1	Modèles sans mémoire	43
2.5.1.1	Random Walk	43

2.5.1.2	Random Waypoint	43
2.5.1.3	Modèle Random Direction	44
2.5.2	Modèles avec mémoire	44
2.5.2.1	Gauss Markov	45
2.5.2.2	Boundless	45
2.6	Évaluation des performances par simulation	46
2.6.1	Environnement de simulation	46
2.6.2	Critères d'évaluation	47
2.7	Résultats de simulations	48
2.7.1	Taux de paquets délivrés	48
2.7.2	Délai de bout en bout	49
2.7.3	Charge de routage normalisée	50
2.7.4	Débit moyen	52
2.8	Conclusion	53
3	Optimisation d'énergie dans AODV en environnement réel (E-AODV)	55
3.1	Introduction	55
3.2	Protocole E-AODV	56
3.2.1	Hypothèses	56
3.2.2	Paquets et structures de données de E-AODV	56
3.3	Description du E-AODV	56
3.3.1	Découverte de chemin dans E-AODV	57
3.3.1.1	Initialisation de la demande de route	57
3.3.1.2	Propagation de la demande de route	57
3.3.1.3	Propagation de la réponse de route	57
3.3.2	Sélection des routes	58
3.3.2.1	Mécanisme de l'E-AODV	58

3.3.2.2	Calcul du coût d'une route	59
3.3.3	Maintenance des chemins	61
3.4	Tests et résultats	61
3.4.1	Paramètres de simulation	61
3.4.2	Résultats des simulations	61
3.4.2.1	Charge de routage normalisée	61
3.4.2.2	Taux de paquets délivrés	63
3.4.2.3	Énergie consommée	63
3.4.2.4	Nombre de nœuds vivants	63
3.5	Conclusion	64
4	protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie	66
4.1	Introduction	66
4.2	Stratégie et outils d'implémentation	67
4.3	Description de la solution proposée	67
4.4	Architecture Cross-layer du protocole EMCMI-AODV	68
4.4.1	Approche au niveau de la couche physique	68
4.4.2	Approche au niveau de la couche MAC	71
4.4.2.1	Méthode d'accès multi-interface multicanaux	71
4.4.2.2	Principe de fonctionnement de la méthode d'accès multi-interface multicanaux	71
4.4.3	Approche au niveau de la couche réseau	73
4.4.3.1	Mécanisme de propagation de la demande de route	74
4.4.3.2	Métriques	75
4.4.4	Sélection des chemins	76
4.4.4.1	Modèle et hypothèses	76
4.4.4.2	Calcul des routes et acheminement des données	76

4.4.4.3	Parcours de réseau pour calculer le chemin le plus court . . .	78
4.5	Tests et résultats	80
4.5.1	Paramètres de simulation	80
4.5.2	Résultats de simulation	80
4.5.2.1	Charge normalisée du routage	80
4.5.2.2	Paquets perdus	81
4.5.2.3	Durée de vie du réseau	83
4.6	Conclusion	83
5	Optimisation multi-objectif des services dans les réseaux mobiles Ad Hoc	85
5.1	Introduction	85
5.2	Modélisation du système	86
5.2.1	Méthodes au fonctionnement couplé	86
5.2.2	Gestion de l'interface fixe et commutable	87
5.2.3	Gestion de la file d'attente	88
5.2.4	Estimation de l'énergie résiduelle	89
5.2.5	Estimation de délai disponible	89
5.3	Description du GA-AODV et DP-AODV	90
5.3.1	Découverte des chemins	90
5.3.2	Sélection des chemins	90
5.3.2.1	Algorithme génétique	90
5.3.2.2	Algorithme dynamique	95
5.4	Modèles de simulation & métriques de performance	96
5.4.1	Modèle de trafic	96
5.4.2	Modèle de mobilité	97
5.4.3	Modèle de propagation	97
5.4.4	Métriques de performance	97

5.5	Résultats et discussion	97
5.5.1	Taux de livraison normalisé (Packet delivery ratio)	97
5.5.2	Délai de bout en bout	99
5.5.3	Charge normalisée du routage	100
5.5.4	Durée de vie du réseau	101
5.6	Conclusion	102
Conclusion et perspectives		103
A Implémentation et Simulation		105
A.1	Introduction	105
A.2	Aperçu du logiciel	106
A.2.1	Principe de base de simulation	106
A.2.2	Arborescence des classes compilées dans NS2	106
A.2.3	Liaisons entre l'interpréteur et le simulateur	106
A.2.4	Caractéristiques d'un nœud sous NS-2	108
A.2.4.1	Classifier	108
A.2.4.2	Lien	108
A.2.4.3	Agent	109
A.3	Implémentation	109
A.3.1	Implémentation du modèle Gilbert-Elliot	110
A.3.2	implémentation multi-interfaces multicanaux	110
A.3.2.1	Changement au niveau du code OTCL	110
A.3.2.2	Changement au niveau du code c++	115
A.3.2.3	Changement du protocole AODV pour s'adapter au nouveau modèle	116
A.3.3	Implémentation de l'algorithmes génétiques	116
A.4	Simulation	118

A.4.1	Génération de trafic	118
A.4.2	Modèle de mobilité	118
A.5	Conclusion	119
Bibliographie		124

Table des figures

1.1	Modèle des réseaux mobiles Ad Hoc	5
1.2	Message de demande de route (RREQ)	9
1.3	Propagation du paquet RREQ	10
1.4	Chemin pris par RREP	10
1.5	Message Route Reply (RREP)	10
1.6	Message d'erreur (RERR)	11
1.7	Mécanisme d'entretien des routes	11
1.8	Découverte de route dans DSR	12
1.9	Exemple de RREP (Route REPLY) du protocole DSR	12
1.10	Diffusion pure et diffusion en utilisant les MPRs dans OLSR	13
1.11	Découverte de route dans DSDV	15
1.12	Principe de fonctionnement du protocole ZRP	16
1.13	Réseau wLAN : (1) en mode ad-hoc (2) en mode infrastructure [11]	17
1.14	Norme du protocole 802.11[12]	17
1.15	Processus d'étalement de spectre dans le domaine temporel	18
1.16	Processus d'étalement de spectre dans le domaine fréquentiel	18
1.17	Principe du modulateur OFDM	21
1.18	Graphe de Delaunay : Diagramme de Voronoi (lignes discontinues) et graphe de Delaunay associé	22
1.19	Exemple d'accès au canal avec la méthode DCF dans IEEE 802.11	24
1.20	Problème de la station cachée	24
1.21	Transmission des données dans CSMA/CA (mode RTS/CTS)	25
1.22	Interfaces séparées pour la signalisation et les données	27
1.23	Bénéfice de la communication multi-canaux	27

1.24	Approche sans commutation (nombre d'interfaces = nombre de canaux) . . .	28
1.25	Approche dynamique (une seule interface)	29
1.26	Affectation hybride (deux interfaces)	30
1.27	Limitation des diffusions (Ticket based probing)	31
1.28	Routage dans EA_AODV	33
1.29	Algorithme de la recherche de route EA_AODV	34
2.1	Principe de Two-ray ground	40
2.2	Diagramme d'état de transition du modèle de Gilbert-Elliot	42
2.3	Trajectoire d'un mobile générée à partir du modèle Random Walk [46]	43
2.4	Trajectoire d'un mobile générée à partir du modèle Random Waypoint	44
2.5	Taux de paquets délivrés : canal avec modèle Gilbert-Elliot	48
2.6	Taux de paquets délivrés : canal idéal	49
2.7	Délai de bout en bout : canal avec modèle Gilbert-Elliot	49
2.8	Délai de bout en bout : canal idéal	50
2.9	Charge de routage normalisée : canal avec modèle Gilbert-Elliot	51
2.10	Charge de routage normalisée : canal idéal	51
2.11	Débit moyen : canal avec modèle Gilbert-Elliot	52
2.12	Débit moyen : canal idéal	52
3.1	Découverte de routes dans E-AODV.	58
3.2	Charge de routage normalisée (NRL) vs densité et la mobilité des nœuds . . .	62
3.3	Taux de paquets délivrés vs densité et la mobilité des nœuds	63
3.4	Energie consommée après la simulation vs densité et la mobilité des nœuds .	64
3.5	Nombre de nœuds vivants après la simulation vs densité et la mobilité des nœuds	64
4.1	Partage de l'information entre les trois couches physique, Mac et Réseau. . .	68
4.2	Architecture d'un nœud mobile dans NS2.	69

4.3	Nouvelle architecture d'un noeud mobile dans NS2	70
4.4	Paquet Hello diffusé indiquant le canal fixe utilisé par les nœuds voisins . . .	72
4.5	Etat de convergence du tablevoisins	72
4.6	Nœud 3 change son canal fixe à 2 puis diffuse l'information aux voisins pour qu'ils mettent à jour leurs tables	73
4.7	Files d'attente du nœud pour chaque canal	74
4.8	Gestion de batterie d'un nœud	75
4.9	Modélisation d'un réseau mobile Ad Hoc en utilisant le protocole EMCMI-AODV	76
4.10	Structure des tables au niveau du nœud 4	77
4.11	Charge normalisée du routage vs densité et la mobilité des nœuds	81
4.12	Nombre des paquets de contrôle générés vs densité et la mobilité des nœuds .	81
4.13	Nombre de paquets perdus vs densité et la mobilité des nœuds	82
4.14	Durée de vie du réseau vs densité et la mobilité des nœuds	83
5.1	Communication de type cross layer à travers la pile protocolaire.	87
5.2	Gestion des interfaces commutables pour un réseau de 3 noeuds et 3 canaux.	87
5.3	Files d'attente maintenues pour un canal.	88
5.4	Gestion des files d'attente d'un nœud.	88
5.5	Architecture générale d'un algorithme génétique	91
5.6	Exemple de propagation des requêtes RREQ	93
5.7	Exemple de sélection par roulette	94
5.8	Exemple de croisement	94
5.9	Exemple de mutation	95
5.10	Taux de livraison de paquets vs densité	98
5.11	Taux de livraison de paquets vs mobilité	98
5.12	Délai de bout en bout vs densité	99
5.13	Délai de bout en bout vs mobilité	99

5.14	Charge normalisée du routage vs densité	100
5.15	Charge normalisée du routage vs mobilité	101
5.16	Durée de vie du réseau vs densité	101
5.17	Durée de vie du réseau vs mobilité	102
A.1	Dualité des classes de NS2	106
A.2	Arborescence des classes C++ du moteur de simulation	107
A.3	Exemple de communication entre les deux hiérarchies NS2	107
A.4	Nœud NS2	108

Liste des tableaux

1.1	Table de routage du nœud 1	15
1.2	Tableau comparatif entre les différentes technologies de transmission du 802.11	20
2.1	Tableau des paramètres de simulation	47
3.1	Paramètres utilisés dans les simulations	62
5.1	Exemple d'une solution par l'algorithme génétique	92
A.1	Fichiers concernés par les modifications	109

Table des sigles et acronymes

ACK	<i>ACKnowledge packet</i>
AODV	<i>Ad hoc On Demand Distance Vector routing protocol</i>
CBR	<i>Constant Bit Rate</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CTS	<i>Clear To Send</i>
DCF	<i>Distributed Coordination Function</i>
DSDV	<i>Distance Source Distance Vector routing protocol</i>
DSR	<i>Dynamic Source Routing protocol</i>
DSSS	<i>direct-sequence spread spectrum</i>
EIFS	<i>Extended InterFrame Space</i>
EM-AODV	<i>Energy saving Multi-path AODV</i>
FHSS	<i>Frequency Hopping Spread Spectrum</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
MAC	<i>Medium Access Control</i>
MACA/PR	<i>Multiple Access Collision Avoidance with Piggyback Reservation</i>
MANET	<i>Mobile Ad hoc Network</i>
MCFI	<i>Multi-channel multi-interface</i>
MPR	<i>Multi-Point Relays set</i>
NS-2	<i>Network Simulator version 2</i>
OFDM	<i>Orthogonal frequency-division multiplexing</i>
OLSR	<i>Optimized Link State Routing</i>
OSI	<i>Open Systems Interconnection</i>
PAMAS	<i>Power-aware Multi Access Protocol with Signaling</i>
PCF	<i>Point Coordination Function</i>
PDR	<i>Packet Delivery Ratio</i>
QoS	<i>Quality of Service</i>
RTS	<i>Request To Send</i>
SIFS	<i>Short InterFrame Space</i>
SISC	<i>single Interface single Channel</i>
ZRP	<i>Zone Routing Protocol</i>

Introduction générale

Contexte général

Les technologies wifi mode mobile Ad Hoc [1] offrent actuellement des perspectives importantes en télécommunications. Ils présentent un moyen de communication sans fil qui permet la transmission de l'information à travers des unités portables indépendamment de leurs emplacements. Contrairement aux réseaux sans-fil à infrastructure où le maintien de la connectivité est assuré par des équipements spécifiques, aucune administration centralisée n'est disponible, ce sont les nœuds mobiles eux-mêmes qui forment l'infrastructure du réseau.

Dans ce mode du réseau l'interconnexion entre les nœuds peut changer à tout moment, il se peut qu'un hôte destination soit hors de la portée de communication d'un hôte source. Par conséquent, les nœuds du réseau doivent fréquemment redécouvrir leur voisinage et recalculer les routes. Il s'ensuit une consommation d'énergie assez importante pour des terminaux mobiles qui, en général, n'en disposent qu'en quantité limitée. Donc, pour trouver le chemin de routage, il faut choisir un protocole de routage adéquat afin d'assurer une route offrant de hautes performances. C'est pourquoi il est nécessaire de créer de nouveaux protocoles qui répondent aux nouveaux besoins des applications et qui prennent en compte les nouveaux paramètres du réseau (énergie, mobilité, délai, débit, liens asymétriques, nœuds cachés, etc.).

Motivations et problématique

Les technologies sans fil occupent aujourd'hui une place très importante en termes de recherche et d'investissements par rapport à l'environnement filaire. Suite à cette évolution, les systèmes de transmission sont susceptibles de supporter différentes gamme d'applications (voix sur IP, vidéoconférence, visiophonie et vidéo à la demande). Cependant de nouveaux problèmes peuvent apparaître, causés essentiellement par l'absence d'une administration centralisée, les contraintes de médium radio et la limitation des ressources locales.

L'absence d'infrastructure fixe rend le routage plus difficile dans les réseaux mobiles Ad Hoc, c'est pourquoi la plupart des travaux de recherche relatifs à ce domaine ont initialement eu pour objectif de développer des protocoles de routage permettant l'acheminement des paquets entre des terminaux qui ne sont pas dans la même zone radio. Nous avons choisi ici de nous intéresser au problème du routage afin d'évoluer vers un protocole reposant sur des mécanismes qui permettent de garantir la qualité de service.

L'un des problématiques aussi les plus importantes pour les réseaux mobiles Ad Hoc est l'accès au médium radio. La majorité des technologies actuellement utilise un seul canal partagé entre les nœuds. Cette méthode d'accès au canal réduit l'efficacité du réseau, au lieu d'avoir plusieurs transmissions à la fois, il y a une seule qui est permise. Notre objectif est

de proposer une approche cross-layer pour améliorer considérablement le débit du réseau en utilisant la méthode d'accès multi-interface multicanaux au niveau des protocoles MAC (MAC multi-canaux). Cependant, le contexte multicanaux ajoute une nouvelle dimension au niveau MAC qui nécessite un nouveau mécanisme appelé "le mécanisme d'allocation des canaux", puisque dans ce contexte, pour qu'un nœud transmette des données, il doit nécessairement connaître le canal sur lequel son récepteur est prêt à recevoir les paquets de données envoyées.

Malgré les progrès technologiques réalisés, l'énergie représente un facteur déterminant pour ce type de réseaux autonomes et indépendants. Par conséquent, les nœuds d'un réseau mobile ad hoc n'ont pas la possibilité d'avoir une énergie infinie puisque celle-ci, une fois consommée, doit être retrouvée ou rechargée. Pour remédier à ce problème, des nouvelles techniques permettant de gérer la consommation de l'énergie de manière efficace ont été proposés dans cette thèse.

Contributions

Nos travaux réalisés dans le cadre de cette thèse consistent à la conception et la réalisation d'une extension du protocole de routage AODV, qui améliore les performances en termes de débit, délai, taux de perte et la durée de vie du réseau. Dans un premier temps et afin de justifier le choix de la sélection d'AODV, nous avons comparé les performances de ce protocole avec DSR, DSDV et OLSR dans un canal de transmission réel tout en intégrant la probabilité d'erreur de transmission caractérisé par le modèle de Gilbert-Elliot.

Dans un deuxième temps nous avons proposé une stratégie de routage tenant compte deux paramètres essentiels : l'énergie et l'équilibrage de charge du trafic entre tous les nœuds du réseau.

Le choix du premier paramètre est très important parce que l'épuisement de l'énergie d'un nœud n'affecte pas uniquement sa capacité de communication mais peut causer le partitionnement du réseau. Notre but donc est d'allonger la durée de vie du réseau où tous les nœuds mobiles sont traités équitablement. Le deuxième paramètre consiste à équilibrer la charge entre les liens de façon qu'aucun lien ne soit congestionné par rapport aux autres. Cet équilibrage de charges se fait en minimisant la différence entre la charge maximale et la charge minimale de trafic sur les nœuds. De même, nous avons proposé une solution combinant d'une part un protocole de routage avec qualité de services basé sur des algorithmes dynamiques ou génétiques et d'autre part un mécanisme permet l'optimisation de puissance de transmission entre les nœuds afin d'utiliser de façon optimale l'ensemble des canaux orthogonaux disponibles(non-overlapping channels).

Structure de la thèse

Cette thèse comporte cinq chapitres. Dans le premier chapitre, nous présentons une étude bibliographique sur les réseaux mobiles Ad Hoc tout en se focalisant sur les aspects liés à la qualité de service et particulièrement ceux liés aux couches inférieures du modèle OSI (Open Systems Interconnection).

Dans le chapitre 2 une évaluation des performances par simulation des protocoles de routages dans un environnement de transmission réel modélisé par le modèle Gilbert-Elliot a été réalisée afin de sélectionner un protocole de routage en tant que protocole de base dans la conception de nos extensions.

Le problème d'énergie sera abordé par la suite dans le troisième chapitre par la proposition d'un mécanisme basé sur la quantité d'énergie résiduelle disponible de chaque nœud. L'étude de performances sera présentée par la suite afin de montrer le gain apporté par l'utilisation de notre approche.

Dans le quatrième chapitre, nous proposons un protocole de routage permet d'allonger la durée de vie du réseau et de réduire l'inefficacité d'un seul canal partagé. Donc la solution multi-interfaces multicanaux est envisagée. Pour ce faire on adopte une méthode d'allocation de canal permet aux nœuds de choisir le canal le plus convenable dans ses voisinages et vise à améliorer la qualité de service du réseau.

La suite possible des travaux est envisagée dans le chapitre 5, avec des extensions de la métrique proposée et l'algorithme de calcul de routes utilisé. La métrique proposée sera combinée avec les contraintes clés caractérisant les réseaux mobiles ad hoc pour définir une métrique globale de qualité de service adaptée à un environnement réel. Le nouveau protocole ainsi mis en place repose sur une corrélation entre la taille de la file d'attente, le délai, le nombre de saut et l'énergie résiduelle. La discussion des performances du nouveau protocole est consacrée aux simulations et discussions des résultats.

Nous terminons ce rapport par une conclusion qui présente également quelques perspectives possibles pour ces travaux.

Réseaux Mobiles Ad Hoc

Sommaire

1.1	Introduction	4
1.2	Caractéristiques des réseaux mobiles Ad Hoc	5
1.3	Domaines d'application	6
1.4	Stratégies de routages des données dans les réseaux mobiles Ad Hoc	6
1.5	Classification des protocoles de routages	7
1.5.1	Protocoles proactifs	7
1.5.2	Protocoles réactifs	7
1.5.3	protocoles Hybrides	8
1.6	Présentation de quelques protocoles de routage Ad Hoc	8
1.6.1	Ad-hoc On Demand Distance Vector (AODV)	8
1.6.2	Dynamic Source Routing (DSR)	10
1.6.3	Otimized Link State Routing (OLSR)	13
1.6.4	Destination Sequenced Distance Vector (DSDV)	14
1.6.5	Zone Routing Protocol (ZRP)	15
1.7	Etat de l'art sur la QoS dans les réseaux mobiles Ad Hoc	16
1.7.1	Approches au niveau de la couche physique et MAC	16
1.7.2	Approches au niveau de la couche réseau	30
1.8	Conclusion	37

1.1 Introduction

Un réseau mobile Ad Hoc (MANET) est un système distribué d'entités mobiles interconnectées par une technologie sans fil formant un réseau temporaire. Il se distingue par un déploiement facile, rapide et robuste car il ne requière pas d'infrastructure préexistante et d'autorité centralisée puisque chaque nœud peut combiner les rôles de client et de routeur.

Pour communiquer entre eux les nœuds du réseau mobile utilisent une interface de communication sans fil (médium radio ou infrarouge) qui permet de propager les signaux sur une distance limité (Figure 1.1). Le relais est ainsi rendu obligatoire, et les nœuds doivent coopérer pour acheminer les données d'une source vers une destination. La recherche d'une route entre une source et une destination est assurée par un protocole de routage. Ainsi, ce

protocole doit être capable de déterminer les meilleures routes pour transmettre des paquets, réagir aux changements survenus sur le réseau, distribuer et récupérer des informations sur la présence et la joignabilité d'une partie ou de l'entièreté du réseau.

Dans ce chapitre d'état de l'art, nous abordons dans un premier temps les réseaux mobiles Ad Hoc, où il sera question de décrire le fonctionnement, les domaines d'utilisation et surtout les différents protocoles de routage utilisés dans ce mode de réseau. En effet, une classification de ces protocoles y sera effectuée, suivie d'une description détaillée de plusieurs protocoles. La description d'un protocole de routage donné concernera les différentes opérations qui décrivent son fonctionnement, à savoir, la découverte des routes, la manipulation des paquets de contrôle et la maintenance des routes. Dans un second temps nous présentons quelques travaux de QoS dans les réseaux mobiles Ad Hoc permettant de garantir la qualité de service. Les solutions décrites sont situées au niveau Physique, MAC ainsi qu'au niveau réseau.

1.2 Caractéristiques des réseaux mobiles Ad Hoc

A partir de la définition générale citée dans la section précédente, il est intéressant de mettre en avant les caractéristiques principales qui différencient un réseau mobile Ad Hoc d'un réseau classique. D'après [1] les réseaux mobiles Ad Hoc se caractérisent principalement par :

- Topologies dynamiques : Les nœuds sont libres de se déplacer de façon aléatoire ; ainsi, la topologie du réseau - qui est typiquement multi-saut - peut changer de façon aléatoire et rapide à des moments imprévisibles et les liens de la topologie peuvent être unis ou bidirectionnels.
- Equivalence des nœuds du réseau : Dans un réseau classique, il existe une distinction entre les nœuds terminaux (stations, hôtes) qui supportent les applications et les nœuds internes (routeurs par exemple) du réseau qui sont chargés de l'acheminement des données. Cette différence n'existe pas dans les réseaux mobiles Ad Hoc car tous les nœuds peuvent être amenés à assurer des fonctions de routage.
- Bande passante limitée : La bande passante disponible dépend à la fois du nombre

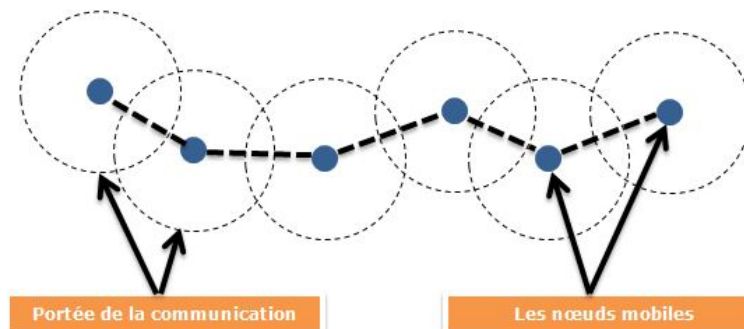


FIGURE 1.1: Modèle des réseaux mobiles Ad Hoc

de nœuds présents dans le voisinage et du trafic de données à transporter qui réduit sévèrement le volume des informations échangées.

- Sécurité physique limitée : les réseaux mobiles Ad Hoc sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques, cela se justifie entre autres par les vulnérabilités des liens radio aux attaques.
- contraintes d'énergie : En effet, la principale contrainte dans les communications sans fil est la durée de vie limitée des terminaux mobiles dont le support énergétique représente souvent une batterie dont la capacité est limitée. C'est pourquoi le paramètre d'énergie doit être pris en considération dans tout contrôle fait par le système.

1.3 Domaines d'application

Historiquement, le premier domaine d'application des réseaux mobiles Ad Hoc fut le domaine militaire pour la surveillance, la reconnaissance, la détection des mouvements de l'ennemi. Cependant, avec l'avancement des recherches dans le domaine des réseaux et l'émergence des technologies sans fils d'autres applications sont apparues. On distingue :

- Les services d'urgence : opération de recherche et de secours des personnes, inondation, tremblement de terre etc. . .
- Le travail collaboratif et les communications dans des entreprises ou bâtiments : dans le cadre d'une conférence par exemple.
- Home network : partage d'applications et communications des équipements domestiques mobiles.
- Applications commerciales : pour un paiement électronique distant ou pour l'accès mobile à l'Internet.
- Réseaux de capteurs : pour des applications environnementales (climat, activité de la terre, suivi des mouvements des animaux, . . . etc.).
- Réseaux en mouvement : informatique embarquée et véhicules communicants (VANETS).

D'une façon générale, les réseaux mobiles Ad Hoc sont utilisés dans toute application où le déploiement d'une infrastructure réseau filaire est trop contraignant soit parce que c'est difficile de la mettre en place ou la durée d'installation du réseau ne justifie pas de câblage à demeure.

1.4 Stratégies de routages des données dans les réseaux mobiles Ad Hoc

Pour les réseaux mobiles Ad Hoc, la question de l'acheminement de paquets entre une paire de nœuds devient une tâche difficile, car les nœuds peuvent se déplacer de façon aléatoire au

sein du réseau. Un chemin qui a été considéré comme optimal à un moment donné peut ne pas fonctionner dans quelques instants plus tard. Donc le but principal de routage est le rétablissement de routes qui soient correctes et efficaces entre une paire quelconque d'unités, ce qui assure l'échange des messages d'une manière continue. Le choix d'une stratégie de routage pour un réseau mobile Ad Hoc dépend de plusieurs contraintes comme l'environnement de propagation, la capacité des nœuds et les exigences de l'application de l'utilisateur.

1.5 Classification des protocoles de routages

Les protocoles de routage destinés à des réseaux filaires ne peuvent pas être utilisés pour les réseaux mobiles Ad Hoc en raison de la mobilité des nœuds. Suivant la méthode de création et la maintenance des routes lors de l'acheminement des paquets, on peut classer les protocoles de routage Ad Hoc en trois catégories :

1.5.1 Protocoles proactifs

Les protocoles de routage proactifs essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud du réseau et les routes sont sauvegardées même si elles ne sont pas utilisées. Lorsque la topologie du réseau évolue, les nœuds diffusent des messages de mise à jour à travers tout le réseau. Cette politique de routage est proche de celle des réseaux filaires. Il existe deux approches pour ce type de protocoles :

- L'approche vecteur de distance où chaque nœud diffuse les distances qui le séparent de tous les autres nœuds du réseau.
- L'approche à état des liens où il s'agit d'un partage des descriptions des liens avec les nœuds voisins.

Cette famille convient donc bien aux applications interactives. Le plus abouti de ces protocoles est OLSR [2].

1.5.2 Protocoles réactifs

À l'inverse des protocoles proactifs, les protocoles réactifs ne construisent pas les tables de routage au préalable. Si un nœud a des données à transmettre une procédure de découverte globale de routes est lancée, et cela dans le but d'obtenir une information, lorsque le chemin est établi, l'acheminement des données peut commencer. L'intérêt majeur de ce genre de protocoles réside dans le fait que le trafic dans le réseau est réduit par rapport à l'approche proactive puisque le réseau n'investit ces ressources qu'en cas de besoin. Cependant, dans ce type de protocoles la procédure de recherche de routes induit un délai supplémentaire et imprévisible ; un délai avant que le premier paquet soit transmis. Un autre problème qui se

pose dans le cas d'utilisation de ce type de protocoles est le fait qu'à chaque fois qu'une procédure de découverte de route est lancée, le réseau est inondé par un trafic important qui s'avère aussi coûteux. Le travail peut être réduit, dans le cas où un nœud de transit posséderait déjà un chemin vers la destination. Actuellement, le plus connu de ces protocoles est AODV [3].

1.5.3 protocoles Hybrides

Les protocoles hybrides utilisent les caractéristiques proactives et réactives des réseaux Ad Hoc. Ils utilisent un protocole proactif pour avoir des informations sur les voisins les plus proches (au maximum les voisins à deux sauts). Au-delà de cette zone prédéfinie, le protocole hybride fait appel aux techniques des protocoles réactifs pour chercher des routes. Ce type de protocoles s'adapte bien aux grands réseaux, cependant, il cumule aussi les inconvénients des protocoles réactifs et proactifs (messages de contrôle périodique, le coût d'ouverture d'une nouvelle route). Le ZRP (Zone Routing Protocol) [4,5] est un exemple de ce type de protocoles.

1.6 Présentation de quelques protocoles de routage Ad Hoc

1.6.1 Ad-hoc On Demand Distance Vector (AODV)

AODV est un protocole de routage réactif de type vecteur de distance destiné aux réseaux Ad Hoc à nœuds mobiles. C'est un protocole multi-sauts, adaptatif et dynamique par rapport aux conditions de la topologie, et permet d'éviter les situations de boucle fermée dans le routage.

1.6.1.1 Découverte de chemins

Le processus de découverte de chemin dans le cas d'AODV consiste à permettre à un nœud d'envoyer un paquet de contrôle RREQ (figure 1.2), qui est relayé par les nœuds intermédiaires jusqu'à ce que la destination (ou un nœud qui en connaît la route) réponde par un message RREP (figure 1.5).

Lorsqu'un nœud source a besoin d'une route vers une destination (e.g. le nœud 1 dans la figure 1.3 désire envoyer des données au nœud 7) et qu'aucune route n'est disponible, la source 1 diffuse en broadcast un message de demande de route RREQ. Ce message contient un identifiant (RREQ_ID) associé à l'adresse de la source (@SRC) qui servira à identifier de façon unique une demande de route. Le nœud 1 enregistre cet identifiant de paquet RREQ ([RREQ_ID, @SRC]) dans son buffer et l'associe à un timer qui décomptera sa durée de vie au-delà de laquelle cette entrée sera effacée.

Quand un nœud intermédiaire (cas des nœuds 2,3,4,5 et 6 de la figure 1.3) qui n'a pas

de route valide vers la destination reçoit le paquet RREQ, il ajoute ou met à jour le voisin duquel le paquet a été reçu. Il vérifie ensuite qu'il ne l'a pas déjà traité en consultant sa table historique des messages traités. Si la requête RREQ est déjà traitée, il l'abandonne et ne la rediffuse pas. Sinon, il met à jour sa table de routage à l'aide des informations contenues dans la requête afin de pouvoir reconstruire ultérieurement le chemin inverse vers la source. Il incrémente ensuite le nombre de sauts dans la demande de route et la rediffuse.

À la réception d'un paquet RREQ, la destination 7 ajoute ou met à jour dans sa table de routage un chemin vers le noeud voisin duquel il a reçu le paquet ainsi qu'une route vers la source 1. La destination 7 génère ensuite une réponse de route RREP qu'elle envoie en unicast vers le prochain saut en direction de la source (voir figure 1.4). Notons qu'un nœud intermédiaire peut aussi générer un RREP si la requête l'autorise à le faire (bit `destination_only` de la RREQ mis à 0) et qu'il dispose déjà dans sa table de routage un chemin valide vers la destination.

Les nœuds intermédiaires qui reçoivent la RREP (cas du noeud 5 dans la figure 1.4) vont mettre à jour le chemin qui mène à la destination dans leurs tables de routage et retransmettre en unicast le message (après avoir incrémente le nombre de sauts) vers le nœud suivant en direction de la source sachant que cette information a été obtenue lors du passage de la RREQ.

Lorsque la réponse de route atteint la source (nœud 1 dans l'exemple), un chemin bidirectionnel est établi entre la source et la destination et la transmission de paquets de données peut débuter.

1.6.1.2 Entretien des routes

AODV emploie un mécanisme pour signaler les problèmes liés aux liens. Lorsqu'un nœud repère la rupture d'un lien sur le trajet d'une route reliant deux nœuds, il génère vers la source une trame d'erreur RERR (figure 1.6). Cette dernière permet de relancer une nouvelle procédure de recherche de route.

Exemple : la figure 1.7 illustre ce mécanisme d'entretien des routes. Sur cet exemple, le nœud 2, qui ne peut plus joindre le nœud de destination 7 par 5, va envoyer au nœud source



FIGURE 1.2: Message de demande de route (RREQ)

1.6.2.1 Découverte de routes

Lorsqu'un nœud désire émettre des données mais ne dispose d'aucune route vers la destination, il déclenche une découverte de routes (figure 1.8). Celle-ci s'effectue en inondant le réseau comme dans le cas du protocole AODV que nous avons présenté dans la section précédente. La requête contient :

- La source S et la destination D de la Route Discovery.
- Un numéro unique de la requête.
- Un enregistrement qui liste les adresses de chaque nœud intermédiaire à travers lesquels la copie de cette requête a été transmise.

Chaque nœud qui reçoit la requête RREQ non dupliqué et qui ne possède pas de route à la destination demandée va concaténer son adresse à la liste contenue dans le RREQ et le diffuser à ses voisins.

Quand le destinataire reçoit cette requête, il retourne à la source un paquet réponse (RREP) qui liste la séquence de nœuds à travers lesquels la destination peut être atteinte (figure 1.9), cette réponse de route suivra le chemin contenu dans la liste jusqu'à atteindre la source. Ainsi, le chemin est établi entre la source et la destination et la transmission de données peut commencer.

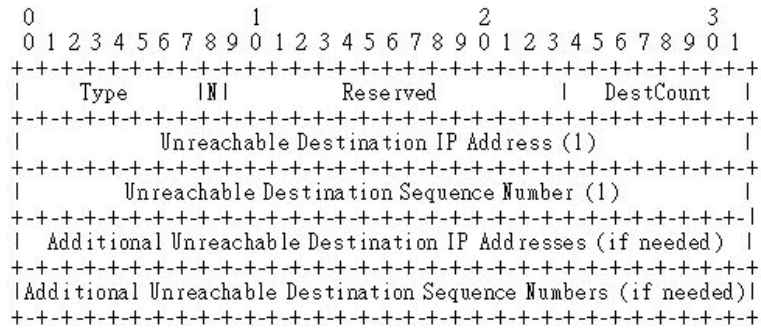


FIGURE 1.6: Message d'erreur (RERR)

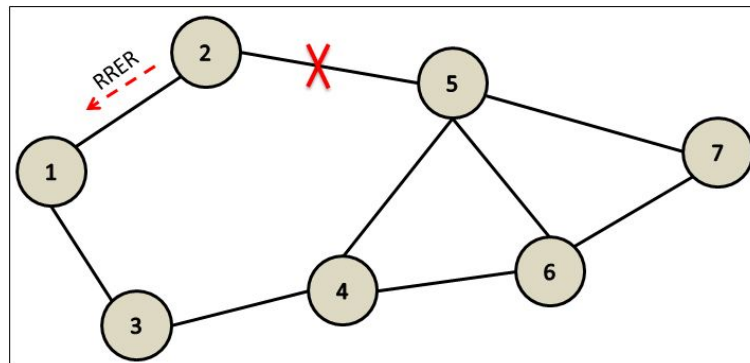


FIGURE 1.7: Mécanisme d'entretien des routes

Dans certains cas, un nœud intermédiaire peut avoir une route qui mène à la destination dans sa cache. Dans ce cas, le nœud intermédiaire peut générer une réponse de route en concaténant le chemin qu'il a reçu dans le paquet RREQ avec celui qui se trouve dans sa cache en s'assurant qu'il n'y a pas de nœuds qui figure dans les deux parties pour éviter la création des boucles de routage.

1.6.2.2 Maintenance de routes

Lorsqu'un nœud initie ou relaie un paquet de données, il doit s'assurer que le nœud suivant dans le chemin a reçu le paquet. Le contrôle de la réception du paquet par le nœud suivant se fait grâce au mécanisme d'acquittement du 802.11.

Quand un nœud détecte un problème fatal de transmission, à l'aide de sa couche de liaison, un message d'erreur de route (RRER) est envoyé à la source, indiquant quel nœud est à l'origine du problème. Lors de la réception du paquet erreur de route par l'hôte source, le nœud concerné par l'erreur est supprimé du chemin sauvegardé, et tous les chemins qui contiennent ce nœud sont tronqués à ce point-là. Par la suite, si la source dispose d'une autre route vers la même destination elle poursuit l'émission avec sa nouvelle route, sinon une

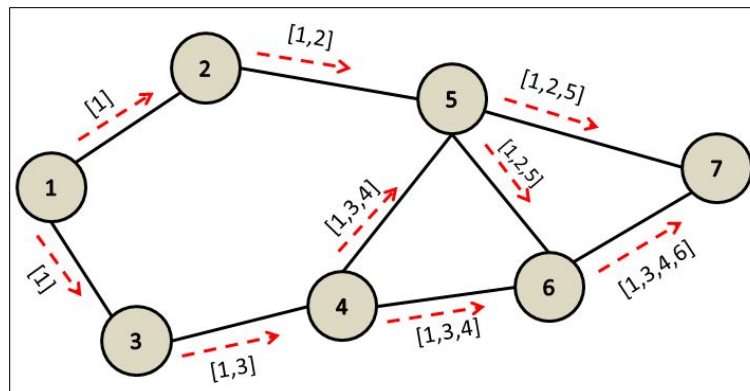


FIGURE 1.8: Découverte de route dans DSR

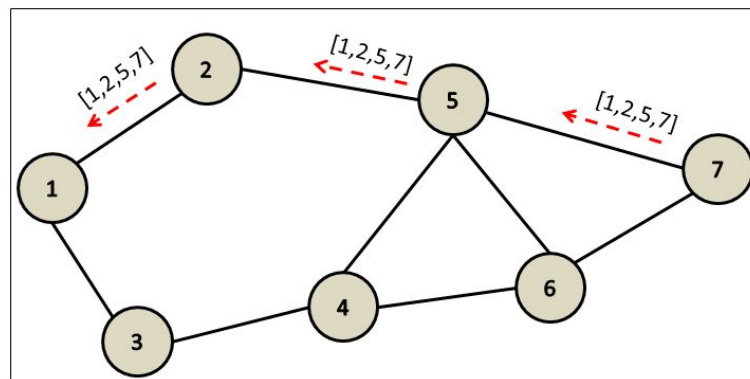


FIGURE 1.9: Exemple de RREP (Route REPLY) du protocole DSR

nouvelle opération de découverte de routes vers la destination est initiée par l'émetteur.

1.6.3 Optimized Link State Routing (OLSR)

OLSR (Optimized Link State Routing) est un protocole proactif à état de lien. Ce protocole propose une optimisation permettant de minimiser l'inondation du réseau grâce à la technique des relais multipoints (MultiPoint Relaying, MPR) et de réduire la taille des paquets échangés.

On peut diviser le fonctionnement du protocole OLSR en trois parties :

1.6.3.1 Découverte des nœuds de voisinage

La découverte des nœuds de voisinage se fait par le message HELLO. Ce message a deux rôles :

- Déterminer le type de liens existant entre un nœud et ses voisins.
- Déterminer les voisins qui serviront de Relais Multipoints (MPR).

Le relais multipoint porte des avantages du fait qu'il permet une diffusion optimisée des messages de contrôle puisque ce mécanisme évite l'envoi périodique de ces messages à tous les nœuds du réseau. Un nœud sélectionne ses MPRs parmi ses voisins à un saut avec un lien symétrique. Cet ensemble est choisi de manière à couvrir (par rapport à la portée radio) tous les nœuds qui sont à deux sauts et seuls les relais multi-points assurent la transmission des informations d'états de liens.

La Figure 1.10 montre la différence entre deux cas : une diffusion pure et une diffusion en utilisant les MPRs. Par exemple, afin d'atteindre tous les nœuds à deux sauts, la diffusion pure a besoin de 16 retransmissions du paquet envoyé par la source. En utilisant les MPRs ou les relais multipoints (nœuds en gras de la Figure 1.10), il suffit de retransmettre le paquet de la source 8 fois pour atteindre les nœuds à deux sauts.

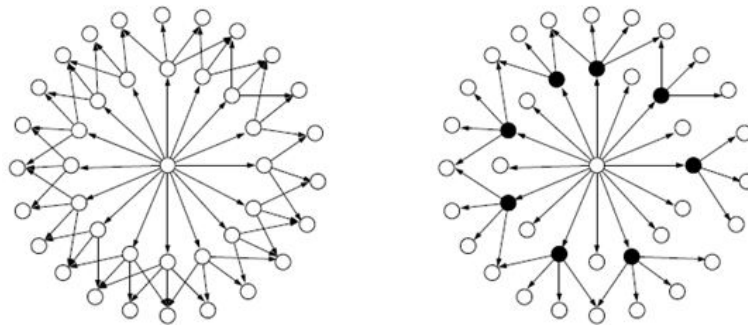


FIGURE 1.10: Diffusion pure et diffusion en utilisant les MPRs dans OLSR

1.6.3.2 Construction de la table topologique

Les messages topologiques TC (topology control) sont émis par un nœud lorsqu'il est MPR d'un de ses voisins. Les informations de ces messages sont stockées dans une table appelée table topologique. Ces informations offrent une carte du réseau contenant tous les nœuds et un ensemble partiel de liens suffisant pour construire une vue global sur l'ensemble du réseau.

1.6.3.3 Calcul des routes

À partir des informations accumulées sur la topologie du réseau, chaque nœud maintient une table de routage qui permet de router les données vers toutes les destinations présentes dans le réseau. Le calcul de la table de routage est basé sur les informations contenues dans deux bases (base d'informations de voisinage et la base de topologie). De ce fait, à chaque fois que l'une de ces bases d'information change, la table de routage doit être recalculée.

1.6.4 Destination Sequenced Distance Vector (DSDV)

Le protocole DSDV [8] est dérivé de l'algorithme classique de vecteur de distance Bellman Ford. Dans ce protocole chaque nœud maintient une table de routage qui contient :

- Toutes les destinations possibles.
- Le nombre de saut nécessaire pour atteindre la destination.
- Le numéro de séquences (SN : sequence number) qui correspond à un nœud de destination.

Pour maintenir l'uniformité des tables de routage, DSDV utilise deux types de mise à jour :

- Une mise à jour complète : chaque nœud transmet la totalité de sa table de routage aux nœuds voisins. Ainsi tout nœud, qui a reçu la mise à jour, compare les données de routage reçus avec les siennes, et la route la plus récente (celle qui a un numéro de séquence élevé) sera utilisée. Si deux routes ont le même numéro de séquence, alors la route qui possède la meilleure métrique est celle qui sera utilisée.
- Une mise à jour incrémentale : pour réduire le nombre de paquets transmis seules les entrées qui ont subi un changement par rapport à la dernière mise à jour sont envoyées.

La métrique utilisée par DSDV dans le calcul des plus courts chemins est le nombre de nœuds intermédiaires existants sur le chemin (nombre de saut). Un lien rompu est matérialisé par une valeur infinie de sa métrique afin d'éviter les grandes boucles de routage et d'offrir des informations explicites en cas d'inaccessibilité d'un lien.

Exemple : Si l'on considère que le DSDV est le protocole de routage utilisé dans la figure 1.11.

La table de routage correspondante au nœud 1 sera :

Destination	Nombre de sauts	Prochain saut	Numéro de séquence
1	0	1	NS1
2	1	2	NS2
3	1	3	NS3
4	2	3	NS4
5	2	2	NS5
6	3	2	NS6
7	3	2	NS7

TABLE 1.1: Table de routage du nœud 1

1.6.5 Zone Routing Protocol (ZRP)

Le protocole ZRP est un modèle hybride, il utilise à la fois un routage proactif et réactif. Dans ce protocole un nœud agit d'une façon proactive dans une zone limitée qui ne dépasse pas un certain nombre prédéfini de saut (rayon de la zone) et devient réactif au-delà de de cette zone. Ainsi, chaque nœud maintient une table de routage dont les mises à jour sont émises périodiquement en diffusion dans la zone proactif.

ZRP met en œuvre deux types de protocoles : IERP (IntErzone Routing Protocol) [9] et IARP (IntrAzone Routing Protocol) [10] pour atteindre tout autre nœud du réseau qui ne figure pas dans sa table de routage. Plus précisément : IARP donne toutes les routes jusqu'à un nombre de saut n limité selon une technique proactive et IERP donne les routes pour des destinations qui dépassent cette limite d'une façon réactive.

La figure 1.12 illustre un exemple de fonctionnement de protocole ZRP, avec n égale à 2. Dans cet exemple le nœud T cherche une route vers D, D n'étant pas dans la zone de T.

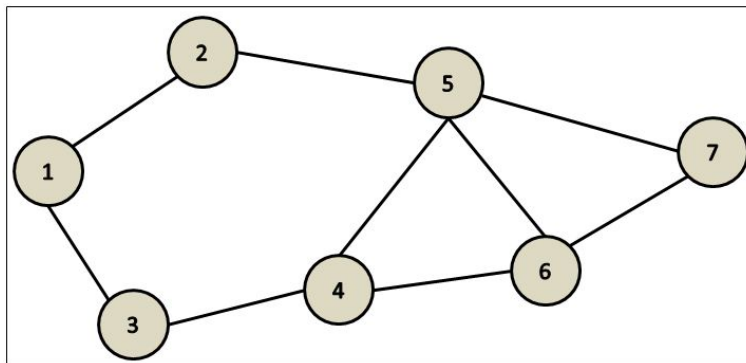


FIGURE 1.11: Découverte de route dans DSDV

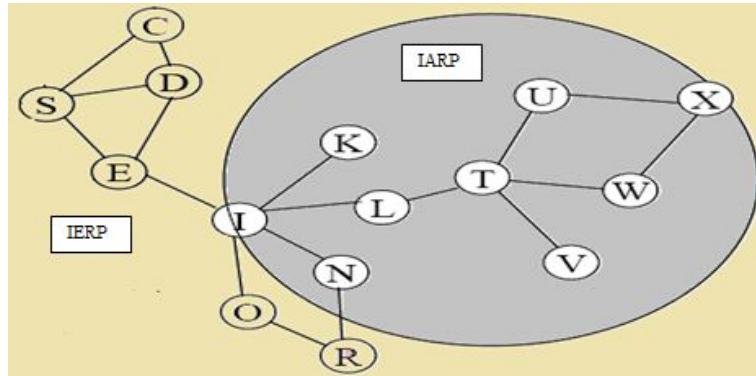


FIGURE 1.12: Principe de fonctionnement du protocole ZRP

Donc, une requête de demande de route est envoyée à tous les nœuds frontières, soit X, I et N. Ces nœuds cherchent alors une route vers D de façon réactive. Une fois la route trouvée elle est ensuite rapportée à T.

1.7 Etat de l'art sur la QoS dans les réseaux mobiles Ad Hoc

1.7.1 Approches au niveau de la couche physique et MAC

Les réseaux mobiles Ad Hoc sont construits sur un médium changeant et peu fiable d'où la nécessité d'un protocole d'accès au médium, le rôle de ce dernier est multiple :

- Fournir une transmission fiable entre deux station du réseau.
- Eviter les collisions.
- Assurer le partage de la bande passante.
- Fournir une correction ou une détection d'erreurs du certains problèmes spécifiques aux transmissions hertziennes (stations cachées ou exposées).

La norme d'accès la plus utilisée est le standard IEEE 802.11. La famille de cette norme définit une architecture cellulaire où les terminaux munis d'une carte d'interface réseau 802.11 s'associent à un point d'accès (mode infrastructure) ou entre eux (mode Ad Hoc) pour former respectivement un BSS (Basic Service Set) ou un IBSS (Independent Basic Service Set) (figure 1.13). Dans notre thèse on s'intéresse aux réseaux mobiles Ad Hoc (figure 1.13 (1)) qui ne nécessitent aucune infrastructure préalablement déployée pour permettre la communication entre ses membres. Chaque station donc opère de manière autonome afin d'assurer sa connectivité et la connectivité des autres membres.

Le standard IEEE 802.11 décrit plusieurs couches (physiques, LLC et MAC) (figure 1.14).

1.7.1.1 Approches au niveau de la couche physique

La couche physique est chargée de véhiculer les bits de l'émetteur au récepteur via une porteuse d'onde hertzienne. Initialement, le standard IEEE 802.11 permet l'utilisation de trois modes de transmission différents (FHSS, DSSS et IR), auxquelles 802.11a a ajouté le mode OFDM. La couche physique définit aussi la modulation des ondes radioélectriques et la puissance de transmission de données.

1.7.1.1.1 Modes de transmission

Étalement de spectre par saut de fréquence(FHSS)

FHSS [13] [14] est une méthode de transmission de signaux qui utilise une technique d'étalement de bande fondée sur le saut de fréquence, dans laquelle la bande ISM (Industrial, Scientific and Medical) des 2.4 GHz est divisée en 79 canaux ayant chacun 1 MHz de largeur de bande. Autrefois, elle est utilisée dans un but militaire car les canaux utilisés pour

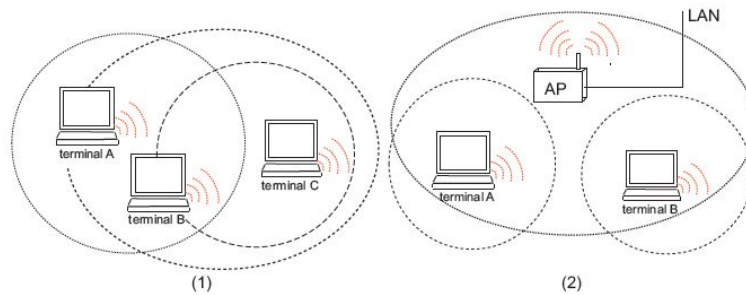


FIGURE 1.13: Réseau wLAN : (1) en mode ad-hoc (2) en mode infrastructure [11]

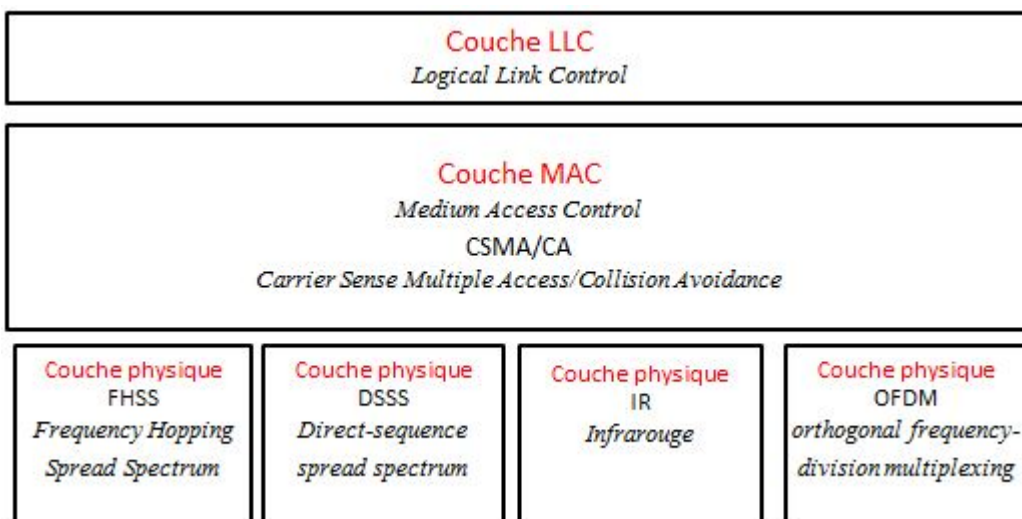


FIGURE 1.14: Norme du protocole 802.11[12]

cette technique restaient secrets. Aujourd'hui la séquence de fréquences utilisée est connue de tous ce qui implique que cette technique n'assure plus cette fonction de sécurisation des transmissions. Pour transmettre des données dans le cas d'utilisation de FHSS, l'émetteur et le récepteur s'accordent sur une séquence de sauts précise qui sera effectuée sur les 79 sous-canaux. La limitation à un débit de 2 Mb/s représente l'inconvénient majeur de cette technique, cette limitation est due à l'attribution de 1 MHz de bande passante par canal combiné à la modulation de fréquence.

Étalement de spectre par Séquence directe (DSSS)

Comme le FHSS, le DSSS divise la bande en sous bande, cependant la division se fait ici en 14 canaux de 20 MHz chacun. Cette technologie consiste à transmettre la trame de symbole $d(t)$ par une séquence d'étalement pseudo-aléatoire $c(t)$ (Code d'étalement ou Pseudo-Noise Code). Le PN-Code va ainsi diviser le code à transmettre en sous intervalles dénommés « Chip » de période bien plus petite que celui du bit à transmettre (la période du bit à transmettre étant le temps symbole). Les figures 1.15 et 1.16 illustrent ce processus dans les domaines temporel et fréquentiel. Au côté de récepteur, la même séquence de PN est utilisée à récupérer le signal large bande et de retourné au signal à bande étroite original [15].

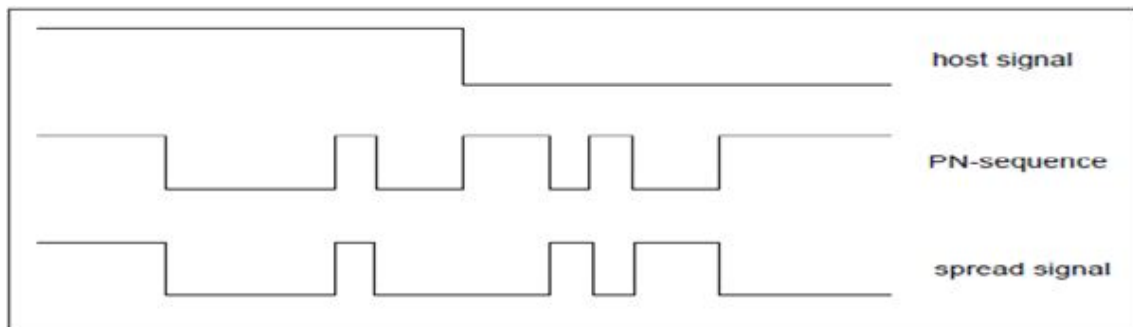


FIGURE 1.15: Processus d'étalement de spectre dans le domaine temporel

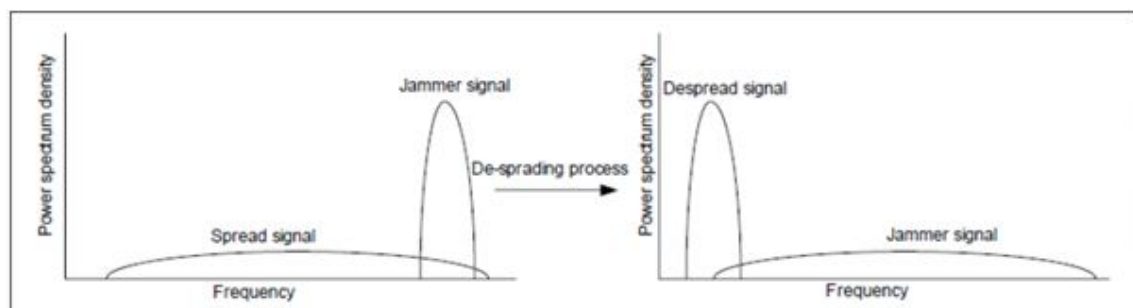


FIGURE 1.16: Processus d'étalement de spectre dans le domaine fréquentiel

Multiplexage par Répartition Orthogonale de la Fréquence (OFDM)

L'OFDM est une technique née à la fin des années 50 avec la société Collins Radio Co. Kineplex system [16]. Cette technologie représente une technique de modulation numérique des signaux, utilisée entre autres pour les systèmes de transmissions mobiles à haut débit. Elle consiste à répartir le signal sur un grand nombre de sous-porteuses dont le spectre est un sinus cardinal qui résulte une orthogonalité entre elles pour éviter l'interférence entre canaux [17]. Peu après, B. Saltzberg a expérimenté la performance d'un tel système [18].

Le Multiplexage par répartition orthogonale de la fréquence s'adapte parfaitement aux communications mobiles, et semble incontournable pour les standards de troisième et quatrième générations, où il est question de l'associer à l'accès CDMA (Code Division Multiple Access) ou aux systèmes multi-antennes [19]. Pour répartir les données à transmettre sur les N porteuses, on groupe les symboles c_k par paquets de N (figure 1.17). Les c_k sont des nombres complexes définis à partir des éléments binaires par une constellation souvent de modulation MAQ à 4, 16, 64, 2^k états. La séquence de N symboles c_0, c_1, \dots, c_{N-1} constitue un symbole OFDM. Le k -ième train de symboles parmi les N trains module un signal de fréquence f_k . Le signal modulé du train k s'écrit sous la forme complexe suivante :

$$c_k e^{2j\pi f_k t} \quad (1.1)$$

Le signal total $s(t)$ correspondant à l'ensemble des N symboles réassemblés en un symbole OFDM est :

$$S(t) = \sum_{k=0}^{N-1} c_k e^{2j\pi f_k t} \quad (1.2)$$

Infrarouge(IR)

La technologie IR s'appuie sur la lumière infrarouge dont la longueur d'onde est comprise entre 850 et 959 nanomètres. Elle permet de créer des liaisons sans fil qui ne dépasse pas quelques mètres parce qu'elle souffre des perturbations dues aux interférences lumineuses. Les débits proposés pour la connexion infrarouge sont :

- Le Basic Access Rate : 1 Mbit/s.
- Le Enhanced Access Rate : 2 Mbit/s.

Le tableau 1.2 en dresse le récapitulatif.

Technologie	Principaux avantages	Principales limitations	Applications typiques
FHSS	<ul style="list-style-type: none"> -Technologie simple et économique. -Permet de "contourner" les interférences. -Portée relativement élevée. -Technologie avantageuse en termes de sécurité et de fiabilité. -Consommation d'énergie relativement faible. 	<ul style="list-style-type: none"> -Efficacité spectrale peu élevée. -Débits relativement faibles. -Nécessite une synchronisation fine entre l'émetteur et le récepteur. -Sensible au nombre d'émetteurs émettant dans la même bande. 	<ul style="list-style-type: none"> -Convient à la transmission de signaux courts, y compris en environnement perturbé. -Solution retenue notamment par Bluetooth (1600 sauts de fréquence par seconde entre 79 fréquences dans la bande ISM 2,4 GHz).
DSSS	<ul style="list-style-type: none"> -Systèmes de redondance par étalement. -Peu sensible aux interférences et aux erreurs de transmission. -Bonne efficacité spectrale. -Possibilité d'obtenir des débits élevés. -Possibilité d'améliorer les performances par allongement du vecteur d'étalement. -Durée d'établissement relativement courte. 	<ul style="list-style-type: none"> -Technologie relativement sophistiquée. -Nécessite des composants rapides. -Consommation d'énergie relativement élevée. 	<ul style="list-style-type: none"> -Convient à la transmission de signaux relativement longs (en dessous d'un seuil de perturbations, qui est fonction du vecteur d'étalement). -Solution retenue notamment pour ZigBee et Wi-Fi (802.11b).
OFDM	<ul style="list-style-type: none"> -Grande efficacité spectrale. -Possibilité d'obtenir des débits très élevés (si le bilan de liaison le permet). -Offre une grande robustesse au regard des interférences (notamment celles qui sont dues aux multitrajets). 	<ul style="list-style-type: none"> -Consommation d'énergie relativement élevée. -Nécessite une synchronisation très fine entre l'émetteur et le récepteur. -Efficacité limitée aux interférences sélectives. -Solution relativement difficile à mettre en oeuvre avec des mobiles (effet Doppler). 	<ul style="list-style-type: none"> -Solution retenue pour Wi-Fi (802.11a dans la bande des 5 GHz et 802.11g dans la bande des 2,4GHz). -Utilisée dans l'ADSL, les courants porteurs et le WiMax.
IR	<ul style="list-style-type: none"> -Très économique. -confidentialité des transmissions. 	<ul style="list-style-type: none"> -La transmission se fait avec une longueur d'onde très faible. -Une traversée des obstacles (murs, plafonds, cloisons...) n'est pas possible. 	<ul style="list-style-type: none"> Technologie largement utilisée pour la domotique (télécommandes).

TABLE 1.2: Tableau comparatif entre les différentes technologies de transmission du 802.11

1.7.1.1.2 Puissance de transmission

La puissance de transmission a un impact majeur sur la durée de vie des nœuds et la capacité du réseau en terme de trafic transporté. En effet, les auteurs de [20, 21] montrent que la puissance nécessaire à la transmission augmente en fonction de la portée des nœuds. Augmenter la portée implique aussi augmenter la probabilité d'interférence, le taux de collisions et de perte de données. Le contrôle de puissance consiste donc à adapter les portées et les puissances de transmission des nœuds mobiles afin de minimiser la consommation d'énergie tout en assurant la connectivité du réseau.

Une autre technique utilisée dans cet axe de recherche est le contrôle de topologie, où la topologie d'un réseau mobile Ad Hoc multi sauts est l'ensemble des liens de communication entre paires des nœuds. Plusieurs propositions ont été faites pour assurer le contrôle de topologie des réseaux mobiles Ad Hoc en jouant sur le rayon de transmission. Le mécanisme proposé dans [22] a pour objectif d'augmenter la capacité de transmission du réseau avec le plus petit niveau d'énergie permettant d'assurer la connectivité entre les nœuds. Mais, assurer une portée minimale toute en gardant la connectivité du réseau pose le problème du contrôle de partitionnement et de recherche de la meilleure couverture du réseau. Afin de remédier à ce problème la référence [23] propose de calculer le diagramme de Voronoï sur l'ensemble des nœuds une fois la topologie et la localisation des nœuds sont connue, puis d'en déduire le graphe de Delaunay qui permet de relier les nœuds ayant des cellules voisines, aussi on peut extraire l'information du voisinage de la triangulation Delaunay puisque les cellules qui sont proches sont connectées. Le diagramme de Delaunay assure une connectivité totale des nœuds du réseau selon des liens courts assurant une portée minimale (figure 1.18).

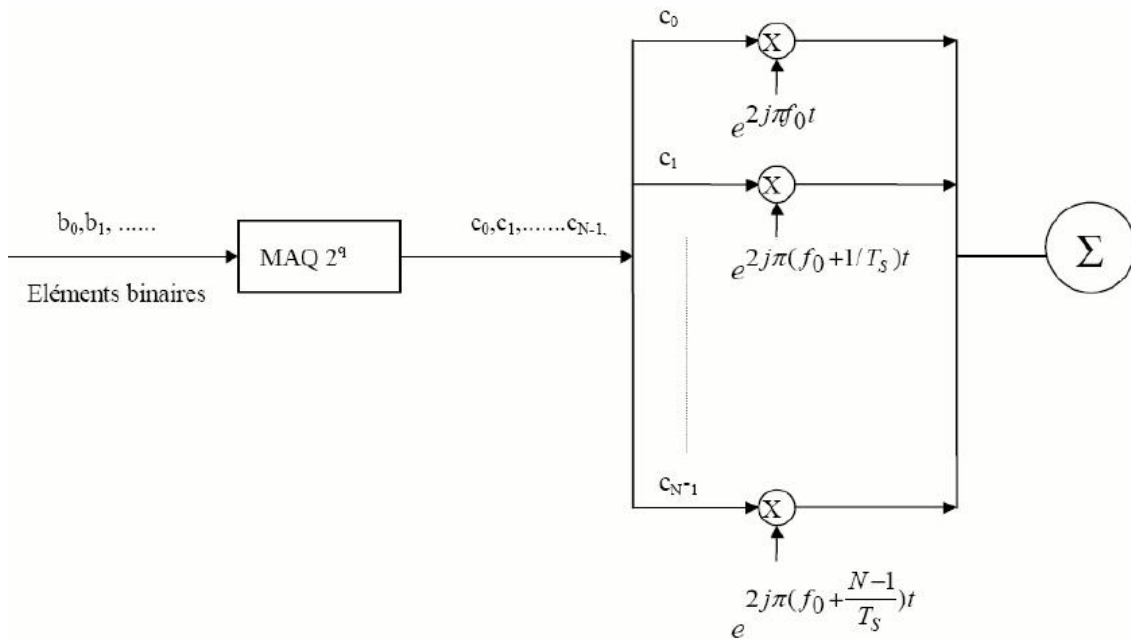


FIGURE 1.17: Principe du modulateur OFDM

1.7.1.2 Approches au niveau de la couche MAC

1.7.1.2.1 Méthode d'accès

La méthode d'accès utilisée par les équipements dans un réseau Ethernet est le CSMA/CD (Carrier Sense Multiple Access with Collision Detect), pour lequel chaque équipement doit écouter le support pour s'assurer qu'il est libre avant de pouvoir transmettre les données. Si ce n'est pas le cas, l'équipement patiente pendant un temps aléatoire avant de recommencer à émettre. Mais ce mécanisme ne suffit pas puisque deux hôtes peuvent être en attente et décident d'émettre leurs messages simultanément et causent donc une collision, notamment quand le réseau est chargé. Si une collision a lieu, les nœuds qui veulent émettre doivent attendre une durée aléatoire, puis la première ayant passé ce délai peut alors réémettre. Le processus se répète jusqu'à ce qu'un équipement puisse émettre sa trame sans qu'il y ait collision. Dans un environnement sans fil ce procédé n'est pas possible dans la mesure où les stations n'ont aucun moyen de savoir si une collision a eu lieu. Ainsi la norme sans fil 802.11 propose un protocole similaire appelé CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Lorsque le canal est libre, la station émet ses données et attend la réception d'une trame d'acquittement (ACK) de la station destination. Une fois l'ACK reçu, la station émettrice comprendra que la transmission a été faite sans collisions. Dans le cas contraire et à l'expiration d'un temporisateur, l'émetteur retransmet la trame victime de

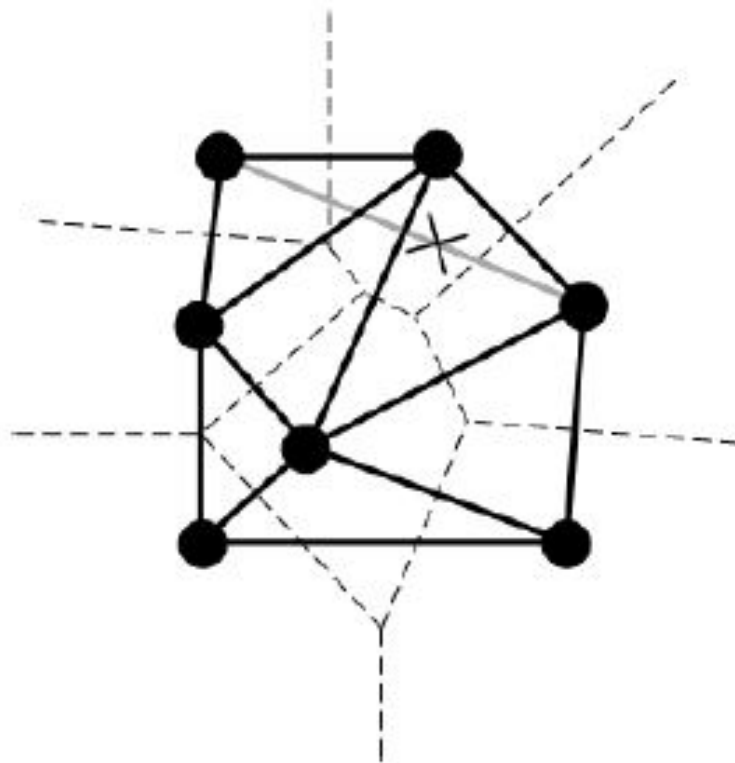


FIGURE 1.18: Graphe de Delaunay : Diagramme de Voronoi (lignes discontinues) et graphe de Delaunay associé

collisions.

La norme 802.11 propose deux méthodes d'accès :

- DCF (Distributed Coordination Function), basée sur le principe d'égalité des chances d'accès au support de transmission pour tous les utilisateurs (méthode probabiliste).
- PCF (Point Coordination Function), basée sur une méthode de pooling gérée par le point d'accès (méthode déterministe).

Nous allons détailler la méthode d'accès DCF basée sur la politique CSMA/CA qui concerne le cas le plus courant. Cette méthode d'accès fonctionne en mode Best-Effort, c'est à dire sans priorité et sans garantie. Un ensemble d'intervalles de temps est spécifié par le protocole DCF. Ces intervalles sont des multiples d'un intervalle de base IFS (InterFrame Spacing) :

Short InterFrame Space (SIFS)

C'est le plus petit écart entre deux trames. Il est utilisé pour permettre aux trames qui sont liées à un même dialogue de se succéder sans être interrompues (par exemple lorsque une station reçoit correctement une trame, elle envoie après une période SIFS à la station émettrice une trame d'acquiescement.).

DCF InterFrame Space (DIFS)

Utilisé lorsqu'une station veut initialiser une communication. Une station souhaitant émettre dans ce contexte, en dehors d'un dialogue établi, doit attendre que le support soit libre pendant ce temps DIFS avant de pouvoir émettre.

Extended InterFrame Space (EIFS)

C'est l'intervalle le plus long, il est utilisé par une station recevant un paquet qu'elle ne comprend pas. Ceci permet d'éviter que cette station (celle qui ne comprend pas l'information) ne provoque pas de collision avec un futur paquet puisque elle n'est pas concernée par l'échange.

La figure 1.19 illustre un exemple d'accès au canal avec la méthode DCF dans IEEE 802.11.

De manière optionnelle, le mode DCF supporte également l'échange de paquets d'appel (RTS/CTS :Request To Send / Clear To Send) pour le trafic en mode Ad Hoc, ce qui permet notamment de pallier au problème des stations cachées dans lequel les stations qui ne s'entendent pas tentent d'émettre simultanément vers une destination. Prenons le cas de trois

stations N1, N2 et N3 comme illustré dans la figure 1.20. Lorsque la station N1 transmet à la station N2, la station N3 ne détecte pas l'activité de N1 car N1 n'est pas dans la zone de couverture de N3, N3 peut alors transmettre en même temps une trame vers N2, en provoquant une collision.

Le fonctionnement du mécanisme RTS/CTS est le suivant : Lorsqu'une station veut envoyer des données elle envoie à la station réceptrice une trame courte RTS (Request To Send) contient la source, la destinataire et la durée estimée de la transmission, puis la station destination répond, après un SIFS, par un CTS (Clear To Send) annonçant à toutes les stations voisines que le médium va être réservé pendant la durée indiquée. Lorsque la station émettrice reçoit correctement le CTS la transmission commence après un autre SIFS. La figure 1.21 montre une transaction entre deux stations utilisent le mécanisme RTS/CTS :

Les paquets de type Broadcast sont envoyés sans utilisation du mécanisme RTS/CTS.

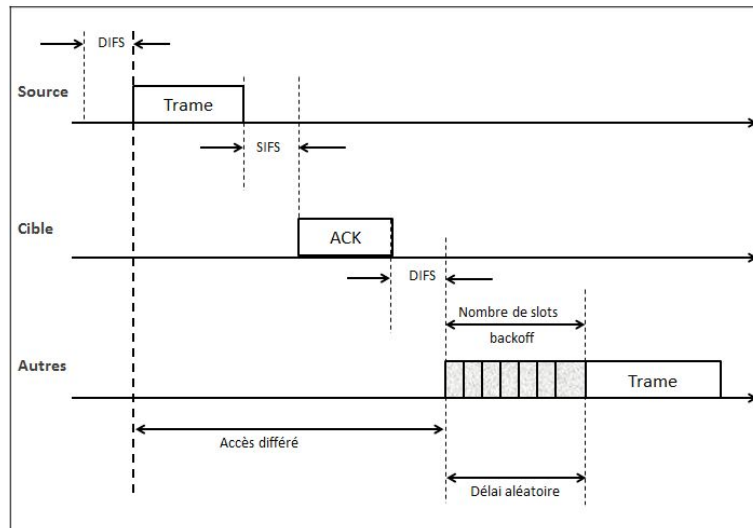


FIGURE 1.19: Exemple d'accès au canal avec la méthode DCF dans IEEE 802.11

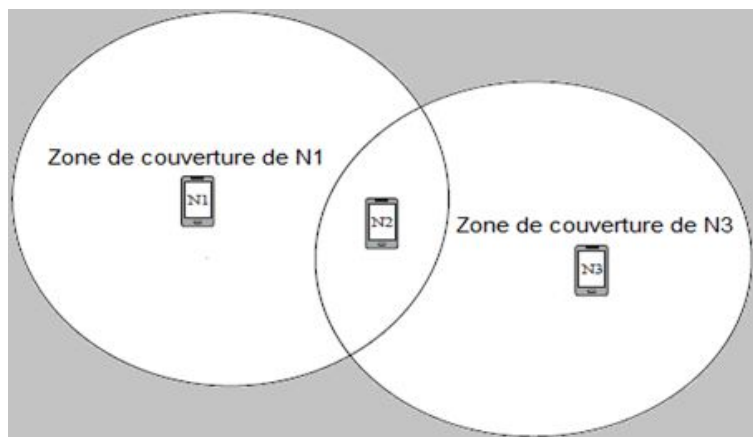


FIGURE 1.20: Problème de la station cachée

Cependant, les paquets de type Unicast suivent la séquence RTS/CTS/DONNEES/ACK entre l'émetteur et le cible. Bien entendu le RTS/CTS n'élimine pas complètement le risque d'avoir des collisions. Cependant, s'il y a une collision, elle se produit entre des trames de petite taille, ce qui améliore la qualité de service des transmissions.

1.7.1.2.2 Qualité de service au niveau MAC

Il y a eu beaucoup de recherches sur la qualité de service pour les réseaux mobiles Ad Hoc, la plupart d'entre elles sont concentrées sur une solution dans la couche MAC. Ceci est justifié par le fait que l'approvisionnement de QoS n'est pas possible à moins que supporté par un des protocoles de cette couche. Le principal objectif de ces protocoles, dans un contexte de qualité de service, est de réduire le plus possible la consommation d'énergie des nœuds, les délais d'accès pour les flux de données et d'augmenter le débit des transmissions.

Les approches existantes dans la littérature au niveau MAC peuvent être classifiées en deux parties suivant les systèmes de qualité utilisés :

- mono-canal.
- multicanaux.

A) Système mono-canal

Dans un réseau mobile Ad Hoc, un protocole d'accès au médium MAC efficace avec lequel les différents nœuds peuvent partager un canal commun est essentiel. Les protocoles idéals doivent donc éviter toutes les collisions, utiliser tous les slot time du canal de transmission et augmenter la durée de vie du réseau. Parmi ces protocoles on trouve MACA/PR (Multiple

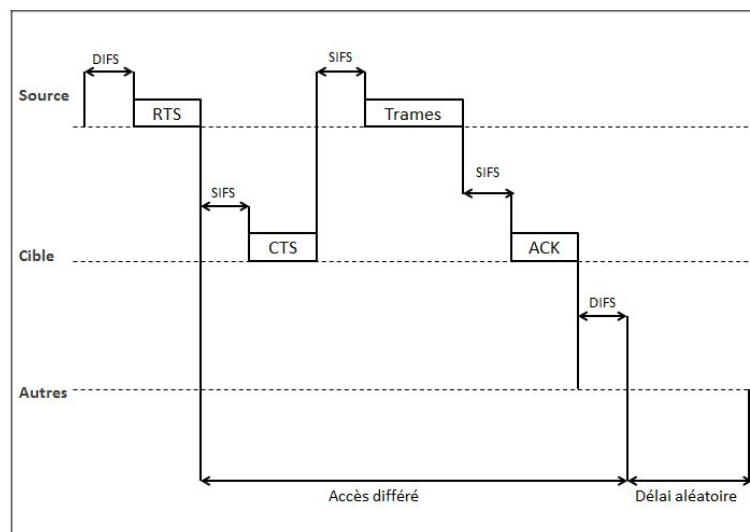


FIGURE 1.21: Transmission des données dans CSMA/CA (mode RTS/CTS)

Access Collision Avoidance with Piggyback Reservation), PAMAS (Power-aware Multi Access Protocol with Signaling) et le mécanisme Blackburst.

MACA/PR (Multiple Access Collision Avoidance with Piggyback Reservation)

Les auteurs de protocole MACA/PR (Multiple Access Collision Avoidance with Piggyback Reservation) [24] proposent une bande passante garantie dans le réseau permet une transmission rapide et fiable des données non temps réel. Pour la transmission des données non temps réel avec MACA/PR, le nœud qui veut envoyer des données doit d'abord attendre une «fenêtre» libre dans le tableau de réservation (RT) dans laquelle les paquets à envoyer et à recevoir sont enregistrés. Le nœud attend alors pendant un temps aléatoire supplémentaire et vérifie que le canal est libre avant de commencer une demande d'envoi par RTS, CTS et ACK. Si le support est occupé, il répète la procédure. Dans le cas de transmission de paquets en temps réel, le fonctionnement de la MACA/PR est différent. Dans ce mode, pour envoyer le premier paquet de données, l'expéditeur lance RTS-CTS, puis avec ACK si le CTS est reçu.

Afin de traiter les réservations de bande passante, les informations de planification en temps réel est inclut dans chaque paquet. Tous les voisins du nœud récepteur peuvent donc différer leurs transmissions pour garantir le trafic en temps réel en se basant sur l'information de la date d'arrivée du prochain paquet.

PAMAS (Power-aware Multi Access Protocol with Signaling)

Ce protocole [25] a été proposé pour conserver l'énergie du réseau en mettant les nœuds ne transmettant ou ne recevant pas des données en veille aussi longtemps que possible puisque ce mode est celui qui consomme la plus faible quantité d'énergie. PAMAS est une combinaison du protocole original MACA [26] et l'utilisation d'un canal séparé pour un signal d'occupation [27] afin de permettre à toutes les stations d'éteindre leur interface radio dans l'un des cas suivants :

- Le nœud n'a pas des paquets à émettre et au moins un voisin est en train d'émettre vers une autre station, puisque dans ce cas il ne pourra pas recevoir des données correctement.
- Le nœud a des paquets à émettre et une autre station voisin est en train de recevoir, parce que s'il émet, il la perturbe.
- Le nœud a des paquets à émettre et tous les stations dans son rayon de couverture sont en train d'émettre, parce que dans ce cas, aucun de ses voisins ne pourra lui acheminer ses données.

Le mécanisme d'utilisation des interfaces séparées du protocole PAMAS est illustré dans la figure 1.22.

Blackburst

Dans [28] les stations de priorité faible accèdent au canal d'une façon normale (utilisation de la norme IEEE 802.11). Mais lorsqu'une station souhaite émettre des données appartenant à un flux prioritaire, il brouillera le canal dès que celui-ci devient libre, et ce pendant un temps proportionnel au temps depuis lequel elle attend. Ce brouillage bloque le backoff et le mécanisme de détection de porteuse des autres stations. Le nœud le plus prioritaire donc pourra y accéder librement dès la fin de l'émission de bruit. Les auteurs montrent qu'il est possible d'éviter les collisions en choisissant correctement les temporisateurs du protocole.

B) Système multicanaux

Plusieurs protocoles MAC multicanaux pour les réseaux ad-hoc ont été proposés pour maximiser la capacité du réseau, minimiser les interférences et maintenir la connectivité du réseau, puisqu'ils permettent aux différents nœuds de transmettre parallèlement sur des canaux distincts sans collision. Dans la figure 1.23 les deux sous-figures (a) et (b) représentent respectivement le cas d'une transmission mono-canal et multicanaux avec trois canaux non recouverts disponibles. Dans le cas mono-canal, les trois transmissions se font l'une après l'autre, tandis que dans le cas multicanaux, les trois transmissions peuvent se produire simultanément dans le premier slot, ce qui aboutit à un gain de trois fois le débit. La grande difficulté pour l'accès multicanaux porte sur le choix et le partage des canaux disponibles par les nœuds dans un contexte repartit, car même avec une connaissance totale de la topologie l'affectation reste un problème NPcomplet lorsque le nombre de canaux est supérieur à deux. C'est pourquoi la problématique de l'allocation des canaux au niveau MAC a fait l'objet de nombreuses recherches qui ont proposé différents mécanismes d'allocation pour la communication Multi-Interfaces MultiCanaux (MIMC).

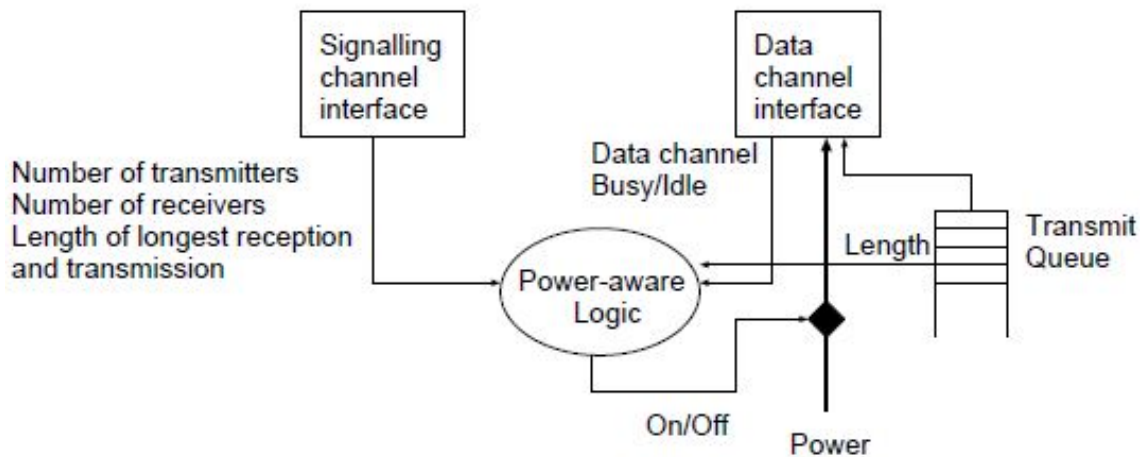


FIGURE 1.22: Interfaces séparées pour la signalisation et les données

Affectation statique de canaux (AFC)

Dans l'approche statique [29,30], toutes les interfaces sont fixées aux canaux pour une période relativement longue ou d'une manière permanente. L'avantage de cette affectation réside dans sa simplicité et la bonne performance qu'elle donne dans le cas d'un trafic uniforme. Pour ce faire, une entité coordinatrice MAC est introduite pour coordonner les différentes interfaces, se comportant ainsi comme une seule couche MAC virtuelle vis-à-vis de la couche supérieure (figure 1.24). Le protocole MUP (Multi-radio Unification Protocol) [29] est un exemple de cette approche dont l'allocation du canal est basée sur la charge des canaux. Dans ce protocole, l'entité coordinatrice se charge de sélectionner l'interface et le canal pour une meilleure communication entre les nœuds. Les auteurs de cette approche ont proposé deux modules pour le bon fonctionnement de ce protocole. Le premier module stock des informations dans une table contenant la qualité du canal pour chaque interface, adresse MAC, canal préféré, date de la dernière trame envoyée ou reçue pour chaque nœud voisin. Le deuxième module se charge de sélectionner l'interface avec un canal ayant une valeur de qualité suffisante en se basant sur la métrique SRTT (Smoothed Round Trip Time). Pour calculer la valeur de SRTT, chaque nœud diffuse périodiquement un message CS (Channel Selection) sur toutes ces interfaces, les nœuds voisins répondent alors par un message CS-ACK. Après chaque réception, le délai d'aller-retour RTT (Round Trip Time) est calculé pour estimer la SRTT. La sélection du canal se fait donc sur la base de l'interface minimisant la SRTT.

Affectation dynamique de canaux (ADC)

Dans le contexte d'une affectation dynamique, un algorithme d'affectation de canaux basé sur la négociation entre les nœuds sera utilisé. L'idée principale de cette approche consiste à partager le temps en une phase de contrôle (établissement de rendez-vous) et une phase de transfert de données (figure 1.25). Une autre stratégie d'affectation dynamique de canaux consiste à allouer n'importe quelle interface radio à n'importe quel canal et les interfaces peuvent fréquemment changer d'un canal à l'autre. Donc lorsque les nœuds veulent communiquer l'un avec l'autre, ils négocient les canaux pour la prochaine phase de transmissions [31]. Cela permet d'allouer des canaux selon la demande. Mais, ce mécanisme produit de mauvaise

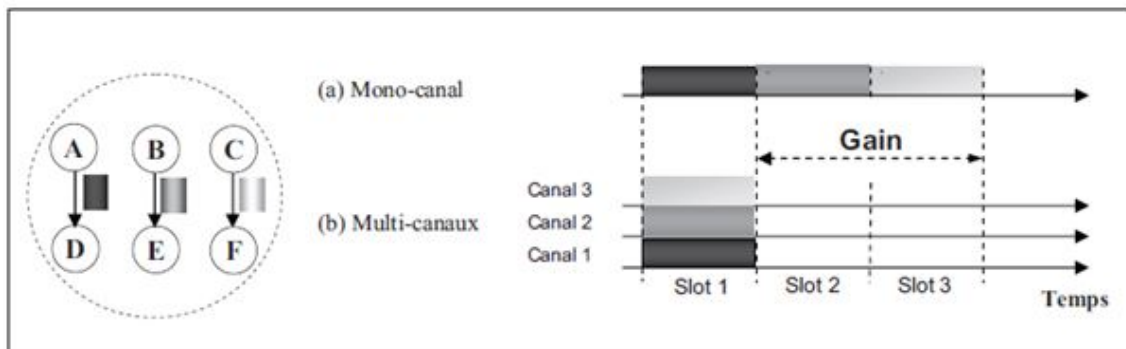


FIGURE 1.23: Bénéfice de la communication multi-canaux

performance lorsque la charge de trafic est élevée.

Affectation hybride de canaux

L'affectation hybride de canaux est une combinaison des approches statique et dynamique en appliquant l'approche d'affectation fixe à certaines interfaces et l'affectation dynamique à d'autres [32,33]. Les auteurs de "Link Layer Protocol-LLP" [33] classent les interfaces disponibles en interfaces fixes attribuées à tous les nœuds de manière à ce que tous les canaux

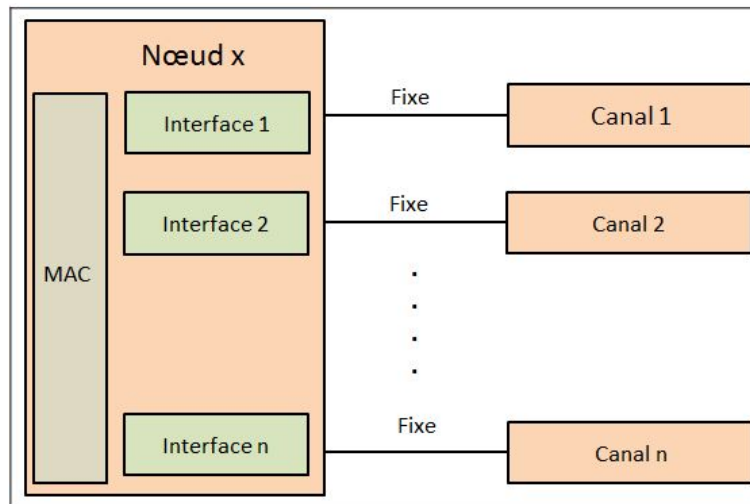


FIGURE 1.24: Approche sans commutation (nombre d'interfaces = nombre de canaux)

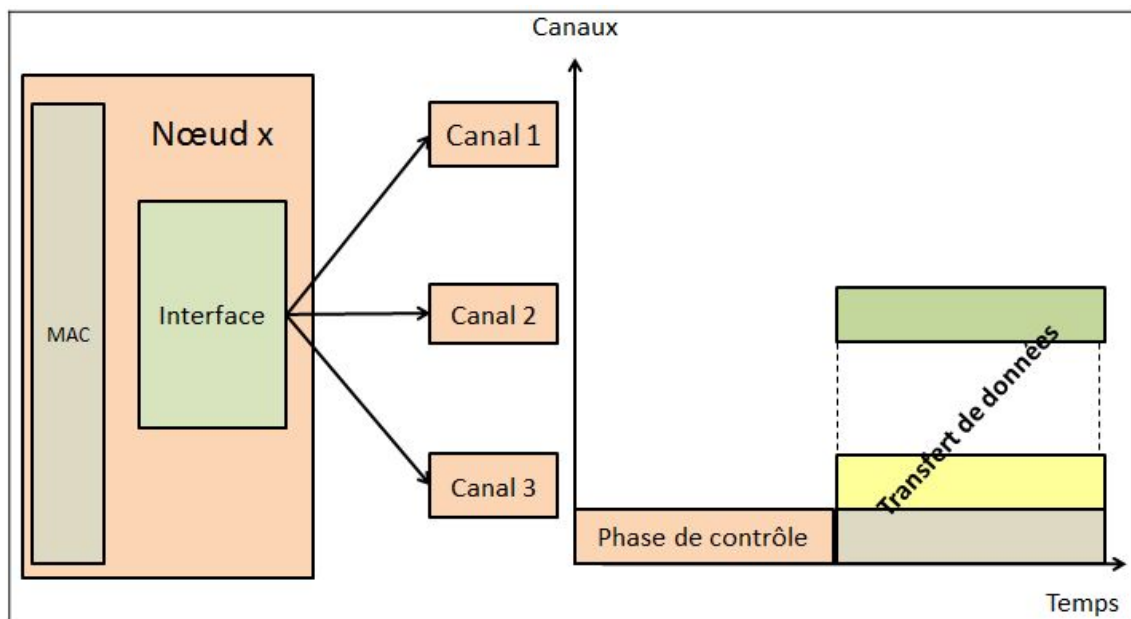


FIGURE 1.25: Approche dynamique (une seule interface)

soient utilisés (les canaux fixes ne sont pas les mêmes pour tous les nœuds) et interfaces commutables afin de profiter de tous les avantages de l'approche dynamique. Une autre idée proposée dans l'algorithme "Interference-Aware Channel Assignment- IACA" [32] basée sur le graphe de conflits multiples-radios permet de maximiser la capacité de réseau tout en minimisant les interférences. L'un des problèmes de ces deux approches est qu'elles ne prennent pas en considération la charge de trafic lors de l'attribution de canaux.

Dans DCA (Dynamic Channel Assignment) [34,35], chaque nœud du réseau maintient deux listes ; une liste de canaux libre (Free Channel List FCL) où les canaux de données libre sont enregistrés et une autre (Channel Usage List (CUL)) dans laquelle le nœud stocke des informations sur l'état d'usage des canaux par ses voisins. Lorsqu'un nœud a des données à émettre, il vérifie si un canal et l'interface de données sont libres. Si ces deux conditions sont satisfaites, l'émetteur envoie un message de contrôle RTS au nœud destinataire incluant sa liste FCL sur le canal de contrôle. Après avoir reçu le RTS, le nœud destinataire compare la FCL reçue avec sa propre FCL et sélectionne un canal libre commun. Ensuite, le nœud destinataire indique au nœud source et à ses voisins, le canal de données sélectionné en envoyant un CTS (Clear to Send). A la réception de CTS, l'émetteur envoie un message RES pour confirmer la réservation du canal, et informe ses nœuds voisins également afin de mettre à jour leurs listes CUL.

Une autre proposition de l'affectation hybride consiste à isoler les trames de contrôle et celles de données pour éviter les interférences entre les deux trafics. Chaque nœud est muni d'une interface de contrôle et une interface de données [34,35]. L'interface de contrôle est fixée de façon permanente sur un canal commun utilisée pour échanger les trames de contrôle et l'interface responsable de transmissions de données peut basculer entre les canaux restants (figure 1.26).

1.7.2 Approches au niveau de la couche réseau

Le routage consiste à déterminer un acheminement optimal des paquets en terme de distance ou de délai entre deux nœuds d'un réseau. Dans le cadre d'une qualité de service, l'objectif du protocole de routage est de trouver la meilleure route selon des critères précis de la qualité de service souhaitée (délai, bande passante, énergie. . .). Afin de s'adapter à ces critères, de nombreux travaux ont été réalisés dans ce domaine avec qualité de service. Cette section introduit les principales approches pour le routage proposées à ce jour.

1.7.2.1 Ticket based probing

Puisque la recherche des routes par diffusion comme le cas d'AODV est très coûteuse en terme de bande passante, les auteurs du protocole Ticket based probing [36] ont réussi de réduire ce coût en associant à chaque demande de routes un certain nombre de tickets pour limiter la diffusion des requêtes. Chaque nœud du réseau maintient un état local (bande passante, délai, . . .) pour tous les liens vers ses voisins immédiats grâce à la transmission

périodique de paquets de signalisation. Il peut donc ainsi sélectionner efficacement les voisins à qui transmettre les demandes de route. Pour augmenter la probabilité de trouver une route, le Ticket based probing utilise deux types de tickets : les tickets jaunes permettent de rechercher des chemins respectant la contrainte imposée et les tickets verts permettent d'obtenir des solutions de faible coût. La figure 1.27 présente un exemple de limitation des diffusions dans lequel deux sonde p1 et p2 sont envoyées à partir du nœud S. la première comporte un ticket et la deuxième deux tickets, au nœud B la sonde P2 est dédoublée en deux sondes chacune comportant un seul ticket car il ne peut pas y avoir plus de trois sondes à la fois. Dans cet exemple les chemins trouvés sont : S->A->D ; S->B->D ; S->B->C->D.

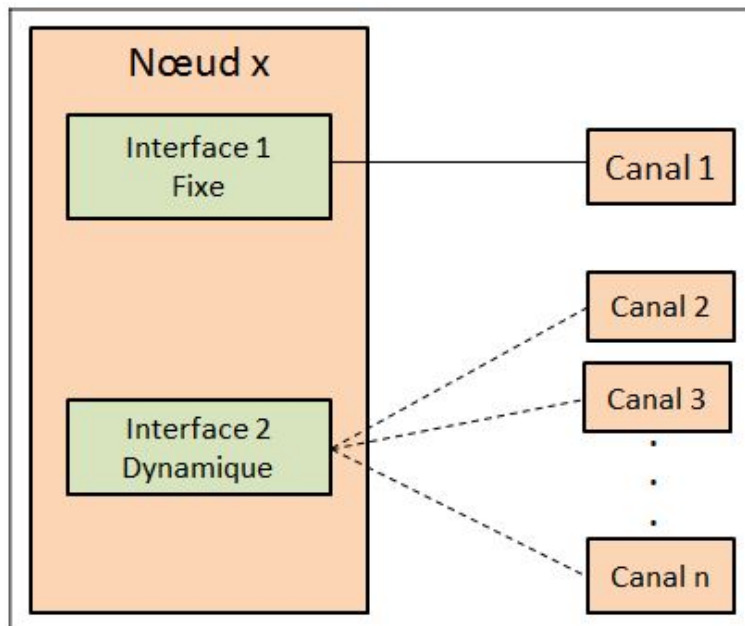


FIGURE 1.26: Affectation hybride (deux interfaces)

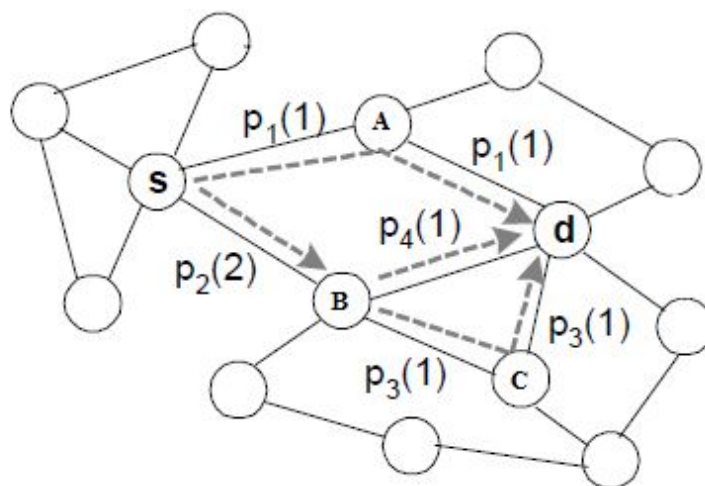


FIGURE 1.27: Limitation des diffusions (Ticket based probing)

1.7.2.2 First Fast Second Reliable

En se basant sur la caractéristique dynamique des réseaux mobiles Ad Hoc les auteurs dans [37] ont proposé une solution nommée FFSR (First Fast Second Reliable). L'objectif principal de ce protocole est d'assurer le transfert des paquets de données à travers le plus court chemin pour compléter tout le transfert de données. Dans le cas où un lien en cours est invalide, la réparation locale au voisinage de la rupture est déclenchée en premier lieu, mais si celle-ci est défectueuse, le nœud source utilise le chemin de sauvegarde, le temps qu'une nouvelle découverte de route soit faite et le transfert soit complet. Les nouveautés introduites dans cette approche sont :

- Le champ all-neighbor est ajouté au paquet RREQ (Route request) pour compter les nœuds voisins contribuant à construire le chemin.
- L'algorithme ne permet pas aux nœuds intermédiaires de retourner un paquet de contrôle réponse RREP (Route Response). Donc les nœuds doivent continuer de transférer le paquet RREQ le long du chemin jusqu'au nœud de destination même s'ils ont l'information de ce dernier dans leurs tables de routage.
- Le nœud destination peut recevoir plus d'une requête RREQ et donc il doit calculer pour chaque RREQ le champ (all-neighbor). On peut avoir les cas suivants :
 - La destination reçoit le premier paquet requête RREQ, lit l'information contenu dans ce paquet reçu concernant le nombre de sauts N et le nombre de voisins, et calcule la moyenne. Elle ne retourne pas un RREP qu'après un certain temps.
 - La destination reçoit un autre RREQ, l'étape précédente est répétée.
 - La source reçoit les paquets RREPs du nœud destination et le chemin qui compte le moins de sauts est établi entre les deux.
 - Si un nœud intermédiaire se déplace ou un lien devient invalide, un mécanisme de réparation est déclenché.
 - Si un chemin est déchu avant la transmission de paquets des données, une réparation locale est lancée.

1.7.2.3 EA_AODV

M. Tamilarasi et al [38] ont proposé le protocole de routage EA_AODV basé sur Ad hoc On-demand vecteur (AODV). Ce protocole intègre l'approche de l'équilibrage de charge et introduit le contrôle de puissance de transmission afin de maximiser la durée de vie des réseaux mobiles Ad Hoc. Le mécanisme de découverte de route dans EA_AODV est illustré dans la figure 1.28 dans laquelle le nœud 1 représente la source et 5 la destination. Au moment T on suppose que :

- Tous les nœuds du réseau contiennent des caches vides.
- Les niveaux d'énergie disponibles de chaque nœud et leur niveau de puissance d'émission requis sont tels que représentés sur la même figure.

- Les nœuds 2 et 4 se trouvent dans la plage de transmission du nœud 1.

La source déclenche donc une découverte d'itinéraire en diffusant le paquet RREQ. Puisque les nœuds intermédiaires 2 et 4 ne sont pas la cible, ces deux nœuds ajoutent leurs propres identifiants de nœud dans le paquet de requête et le rediffusent. Lorsque la destination reçoit le paquet RREQ, il envoie une réponse à la source en inversant le chemin par lequel il reçoit le paquet RREQ.

Supposons que le nœud 5 répond à la source en utilisant la route 5-3-2-1. Lorsque le nœud 3 reçoit le paquet RREP, il estime son énergie restante 'x' en utilisant son énergie actuel et la puissance d'émission requise. Le nœud 3 enregistre cette valeur dans RREP et le transmet au prochain saut (nœud 2). Le nœud 2 estime l'énergie restante 'y' de la même manière et lit également la valeur 'x' enregistrée dans le paquet RREP et la compare avec 'y'. si 'y' est inférieur à 'x' le nœud remplacera 'x' par 'y' sinon, 'x' sera conservé dans RREP. Supposons que 'y' soit inférieur à 'x'. Ainsi, le nœud source 1 enregistre ce chemin dans le cache avec la valeur 'y'. Ainsi, dans EA_AODV, les nœuds mobiles qui ont une valeur d'énergie de leurs batteries très faibles sont évités dans la phase de découverte d'itinéraire. La recherche de la route par EA_AODV ce fait de la manière illustré dans la figure 1.29.

1.7.2.4 Link Prediction

Dans [39] les auteurs ont proposé une méthode de prédiction de la disponibilité de lien basée sur la puissance du signal à utiliser dans le routage AODV. Les nœuds estiment le temps de rupture du lien et avertissent les autres nœuds au sujet de ces ruptures dans l'itinéraire. Sur la base de ces informations, la réparation de lien local ou la découverte de nouveaux itinéraires sera initiée beaucoup plus tôt que la rupture de l'itinéraire. Cela réduit les pertes de paquets de données ainsi que les délais de bout en bout. L'approche proposée est comparée avec l'AODV standard. Les résultats montrent une amélioration significative par rapport au protocole AODV.

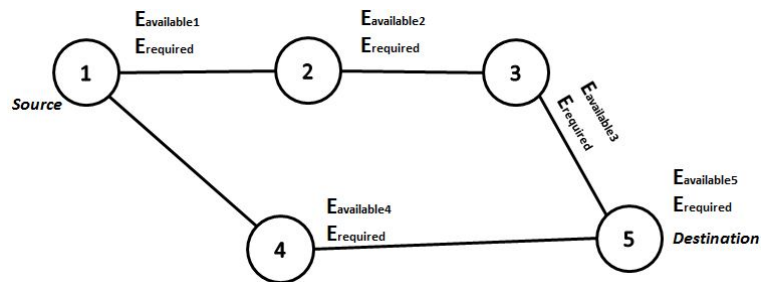


FIGURE 1.28: Routage dans EA_AODV

1.7.2.5 Energy saving Multi-path AODV (EM-AODV)

Les auteurs du protocole EM-AODV (Energy saving Multi-path AODV) [40] considèrent le nombre de sauts et l'énergie totale comme une métrique de routage. Un champ appelé T_s a été ajouté aux paquets Hello. Il prend comme valeur l'énergie résiduelle des nœuds. Lorsque les nœuds diffusent le message Hello, les nœuds voisins à un saut peuvent obtenir les énergies résiduelles mentionnées dans T_s . En raison donc de l'envoi périodique du message Hello, les nœuds de réseau peuvent mettre à jour dynamiquement les énergies résiduelles des nœuds voisins afin d'adapter les changements de la topologie.

Les auteurs ont modifié aussi les requêtes RREQ et RREP en ajoutant les champs E_{min} et E_{sum} aux paquets RREQ et le champ P_{ri} aux paquets RREP. Le champ E_{min} stock la valeur

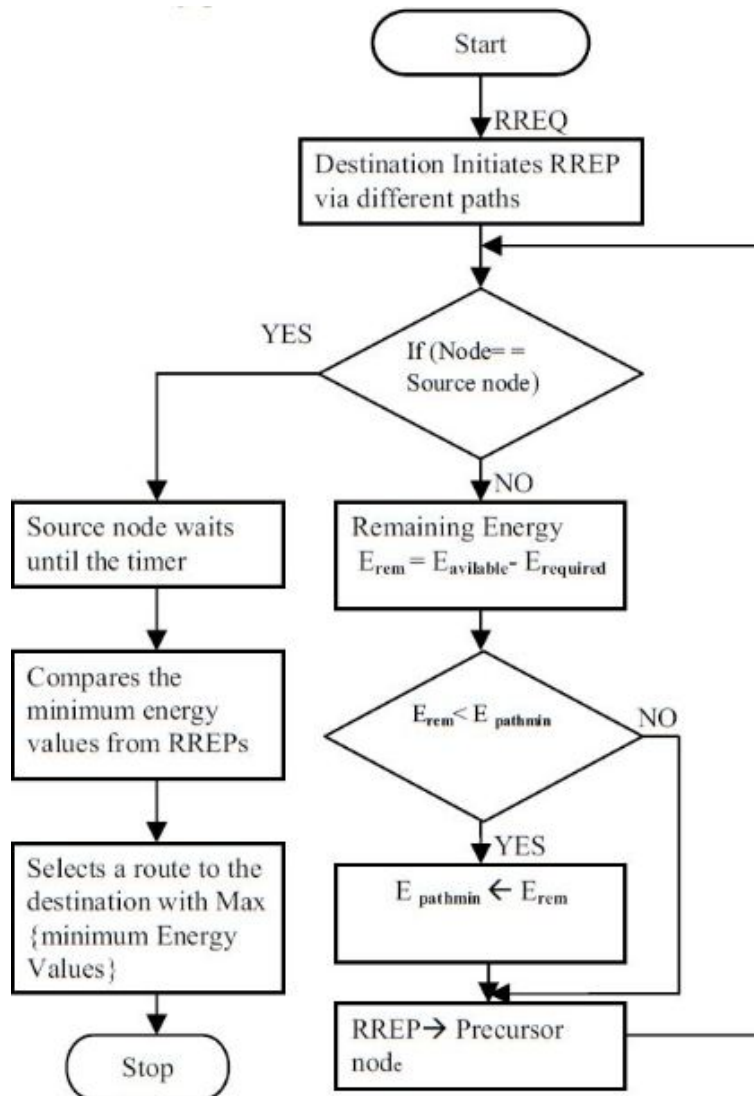


FIGURE 1.29: Algorithme de la recherche de route EA_AODV

de l'énergie résiduelle minimale des nœuds traversés par le paquet RREQ et le champ E_{sum} la valeur d'énergie résiduelle totale des nœuds du chemin. Lorsqu'un nœud intermédiaire reçoit une requête RREQ, il ne retourne pas une réponse RREP mais continue à diffuser le RREQ à ses nœuds voisins jusqu'à atteindre le nœud cible même s'il a un chemin vers ce dernier dans sa table de routage. Le nœud cible définit un délai de mise en file d'attente des paquets RREQs pendant lequel il reçoit tous les RREQ avec les ID de diffusion et calcule la priorité P_{ri} des chemins selon l'équation suivante :

$$P_{ri} = \sqrt{\left(\left(0.6E_{min} + \frac{0.4E_{sum}}{S_{sum}} \right) + \left(\frac{1}{L} \right)^2 \right)} \quad (1.3)$$

Où :

- P_{ri} : Priorité du chemin.
- S_{sum} : Nombre de saut du chemin.
- $E_{initial}$: Energie initiale du nœud.
- L : Charge courant du nœud.

Lorsque le délai est terminé, le nœud cible envoie le RREP avec la priorité au nœud source. Plus la valeur de P_{ri} est élevée, plus le chemin est fiable. Le nœud source choisit le chemin qui a la plus haute priorité comme chemin principal et les autres comme chemins de sauvegarde.

1.7.2.6 ELMA

Les auteurs du protocole de routage ELMA [41] ont proposé trois versions différentes pour maximiser la durée de vie du réseau. Le principe de ce protocole est d'équilibrer la charge du trafic sur l'ensemble des nœuds pour éviter l'épuisement rapide de leurs énergie. Chaque nœud du réseau envoie périodiquement le niveau de l'énergie résiduelle, grâce aux paquets d'énergie. Les nœuds recevant ces paquets peuvent calculer ou estimer l'énergie résiduelle du réseau. Dans cette approche, les nœuds possédant une énergie supérieure à l'énergie moyenne du réseau sont considérés comme des nœuds actifs sinon ils sont considérés comme des nœuds ralentis. La route est choisie selon le rapport d'efficacité définie par l'équation suivante :

$$\rho = \frac{E^{total}}{Hop^\alpha} \quad (1.4)$$

Où :

- E^{total} : Energie totale de la route.
- Hop : Nombre de nœud se trouvant sur cette route.
- α : Variable définie comme $1 \leq \alpha \leq 2$.

Après avoir définie le meilleur chemin. Une réponse de requête (route reply) est envoyée au nœud source qui peut à son tour calculer le nombre de paquets qui peuvent être envoyé par ce chemin, en utilisant la fonction de capacité définie par :

$$\phi_p = \frac{E_{min}}{E_{TH}} \quad (1.5)$$

Où :

E_{min} : Energie de consommation la moins élevé utilisé par le nœud sur ce chemin.

E_{TH} : Seuil de viabilité.

1.7.2.7 LEACH

Le protocole LEACH [42] consiste à partitionner le réseau en zones et clusters de façon distribuée, des nœuds 'maîtres' (Clusterheads (CHs)) sont constitués puis utilisés comme relais pour atteindre le nœud cible en optimisant la consommation d'énergie. Le nœud maître est élu périodiquement parmi les nœuds formant le cluster, en fonction de son énergie résiduelle. Le protocole LEACH se déroule en rounds. Chaque round se compose de deux phases :

Phase de construction

Dans cette phase les chefs (CHs) sont choisis et la politique d'accès au média au sein de chaque groupe est établi. Durant cette phase, chaque nœud n choisit un nombre aléatoire compris entre 0 et 1. Si ce nombre est inférieur à un seuil $T(n)$, le nœud devient automatiquement clusterhead. $T(n)$ est défini par l'équation suivante :

$$T(n) = \begin{cases} \frac{p}{1-p*r \bmod \frac{1}{p}} & \text{si } n \in G \\ 0 & \text{sinon} \end{cases} \quad (1.6)$$

Où :

p : Pourcentage des nœuds désirant devenir clusterhead.

r : Numéro du round courant.

G : Nœuds n'ayant pas été élus clusterheads durant les $1/P$ dernières périodes.

Par la suite, chaque nœud qui s'est élu clusterhead émet un message de notification à tous ces voisins. Les nœuds qui reçoivent les messages de notification, décident leur appartenance à un CH. La décision est basée sur la force du signal reçu (le clusterhead le plus proche est choisi). En cas d'égalité, un chef aléatoire est choisi. Chaque membre informe son chef de sa décision.

Phase de communication

Pendant cette phase, les nœuds mobiles peuvent commencer à envoyer leurs données au nœud maître pendant leurs propres slots. Cela leur permet d'éteindre leur interface de communication en dehors de leurs slots réservés, afin de mieux gérer leur énergie. Le CH agrège les données reçues avant de les transmettre au collecteur (puits). La communication, entre un maître et le collecteur, se fait d'une manière directe. Après la phase de communication, qui dure plus longue afin de minimiser la surcharge, la phase de construction recommence.

1.7.2.8 Hybrid Energy-Efficient Distributed Clustering

Basé sur le même principe de fonctionnement du protocole LEACH, le protocole Hybrid Energy-Efficient Distributed Clustering appelé HEED [43] sélectionne les chefs de grappes (clusters) périodiquement en se basant sur deux paramètres. Le premier paramètre est l'énergie résiduelle du nœud. Ainsi, un nœud avec une grande énergie résiduelle a plus de chance de s'élire clusterhead. Le second paramètre est le coût de communication intra-grappe qui dépend des propriétés du cluster tel que sa taille. Dans ce protocole les nœuds sont mis en veille et ne se réveillent que lorsqu'ils ont le droit de transmettre.

1.8 Conclusion

La conception et la mise en œuvre de protocole au niveau MAC et réseau qui répond aux défis de réseau mobile Ad Hoc représentent une tâche difficile. Dans ce chapitre nous avons parcouru un certain nombre de protocoles d'accès et de routage proposés dans la littérature dans le contexte des réseaux dynamiques de type Ad Hoc. Nous avons remarqué que chaque protocole ne traite qu'un aspect particulier de la transmission dans les réseaux mobiles Ad Hoc. De plus, les topologies des réseaux, ainsi que les types de trafic utilisés sont propres à chaque protocole, c'est pourquoi il est très difficile de comparer ces différentes approches. Nous avons aussi remarqué que la plupart des méthodes d'accès multicanaux qui ont été proposées traitent uniquement le cas des réseaux mono-saut.

Toutes les idées citées dans ce chapitre peuvent nous servir dans la conception de nos solutions pour assurer la qualité de service dans les réseaux mobiles Ad Hoc, dont l'objectif n'est pas seulement de trouver le meilleur chemin selon un critère précis, mais de trouver un chemin admissible satisfaisant plusieurs contraintes (délai, débit, énergie...).

Dans le chapitre suivant une évaluation des performances des protocoles de routages dans un environnement de transmission réel modélisé par le modèle Gilbert-Elliot a été réalisée afin de sélectionner un protocole de routage en tant que protocole de base de nos contributions.

Evaluation de performances

Sommaire

2.1	Introduction	38
2.2	Présentation de network simulator NS2	39
2.3	Modèles de propagation radio	39
2.3.1	Modèle de propagation en espace libre (Free space model)	40
2.3.2	Modèle de propagation Two-ray ground model	40
2.3.3	Modèle de propagation Shadowing	41
2.4	Modélisation d'un canal radio 802.11	41
2.5	Modèle de mobilité	42
2.5.1	Modèles sans mémoire	43
2.5.2	Modèles avec mémoire	44
2.6	Évaluation des performances par simulation	46
2.6.1	Environnement de simulation	46
2.6.2	Critères d'évaluation	47
2.7	Résultats de simulations	48
2.7.1	Taux de paquets délivrés	48
2.7.2	Délai de bout en bout	49
2.7.3	Charge de routage normalisée	50
2.7.4	Débit moyen	52
2.8	Conclusion	53

2.1 Introduction

Actuellement on rencontre un nombre assez important de protocoles de routage dont certains ont fait l'objet de normalisation. L'objectif de ce chapitre est de faire une étude comparative entre quelques algorithmes de routage afin de :

- Confronter par la simulation, à l'aide de NS2, les performances des protocoles DSR, AODV, OLSR et DSDV.
- Examiner l'impact des phénomènes des niveaux physique sur le comportement des protocoles de routage, en intégrant la probabilité d'erreur de transmission caractérisé par le modèle de Markov à deux états dans les différents modes de propagation radio.

Vu qu'ils ne reposent pas sur des modèles analytiques, l'évaluation exacte de certains aspects de ces protocoles est très difficile. C'est la raison qui nous amène à faire des simulations pour étudier leurs performances. Notre simulation est réalisée sous NS2 (Network Simulator 2) [44] qui nous a permis de dégager un classement des différents algorithmes de routage étudiés selon le débit, coût de routage, taux de paquets délivrés et délai de bout en bout.

2.2 Présentation de network simulator NS2

NS2 est un simulateur à événements discrets permet d'exécuter tout type de scenario sur des topologies définies par l'utilisateur. Il est essentiellement élaboré avec les idées de la conception par objets, de la réutilisation du code et de modularité. NS2 est aujourd'hui un standard de référence en ce domaine, plusieurs laboratoires de recherche recommandent son utilisation pour tester les nouveaux protocoles.

Le simulateur NS2 est écrit en C++ et utilise le langage OTCL (Object Tools Command Language) dérivé de TCL. A travers OTCL, l'utilisateur décrit les conditions de la simulation telle que la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés et les communications qui ont lieu. La simulation doit d'abord être saisie sous forme de fichier que NS va utiliser pour produire un fichier contenant les résultats.

2.3 Modèles de propagation radio

Une onde électromagnétique est le résultat d'un champ électrique et d'un champ magnétique dont les amplitudes varient de façon sinusoïdale au cours du temps. La dispersion de la puissance et les obstacles rencontrés sur le chemin de chaque composante du signal reçu (bâtiments, montagnes...) conduisent à l'affaiblissement de l'onde électromagnétique lorsqu'elle parcourt une distance.

Les obstacles rencontrés provoquent la division du signal initial en composantes qui vont emprunter des trajets différents en exploitant les propriétés de propagation.

Théoriquement, il est possible de calculer cet affaiblissement de manière « exacte », dans le cas où la géométrie du lieu de parcours et la nature précise des milieux traversés ou rencontrés sont parfaitement connus, ce qui n'est généralement pas faisable. Dès que la mobilité est introduite pour les nœuds communicants, ce calcul exact devient impossible. Il faut donc avoir recours à des modèles (Free space, Two-ray ground et Shadowing) qui sont expliqués dans les paragraphes suivants.

2.3.1 Modèle de propagation en espace libre (Free space model)

Le modèle de propagation en espace libre présente le cas idéal de condition de propagation où il n'existe qu'un chemin de visibilité directe dégagé entre l'émetteur et le récepteur. Ce modèle considère les zones de communication comme un cercle autour de l'émetteur. Si un récepteur est dans ce cercle il reçoit tous les paquets, s'il est en dehors il n'en reçoit aucun.

La puissance reçue par l'antenne du récepteur situé à une distance d de l'émetteur est calculée selon l'équation suivante :

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad (2.1)$$

Où :

- P_t : Puissance d'émission.
- G_t : Gains des antennes de l'émetteur.
- G_r : Gains des antennes du récepteur.
- λ : Longueur d'onde.
- L : Perte du système.

L'équation 3.1 montre que la puissance reçue s'atténue comme le carré de la distance qui sépare l'émetteur du récepteur. On remarque également que l'équation est un facteur de la longueur d'onde, ce qui implique que deux ondes de fréquences différentes ne s'atténuent pas de la même façon.

2.3.2 Modèle de propagation Two-ray ground model

Contrairement au modèle de propagation Free space qui ne considère qu'un chemin direct (ligne droite) pour propager les ondes radios, le modèle Two-ray ground tente également de considérer la réflexion des rayons sur le sol (d'où le nom de Two-ray). Le schéma de la figure 2.1 illustre ce principe. L'onde reçue au niveau de l'antenne réceptrice se compose donc d'une onde de trajet direct et une autre réfléchi au sol. La puissance reçue peut être exprimée par

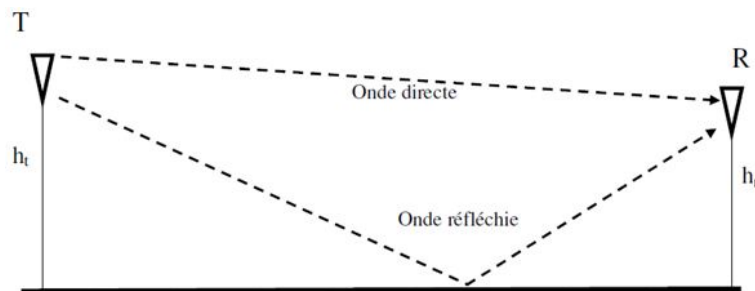


FIGURE 2.1: Principe de Two-ray ground

l'équation 2.2 :

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4} \quad (2.2)$$

Où h_t et h_r sont respectivement les hauteurs des antennes de transmission et de réception.

2.3.3 Modèle de propagation Shadowing

Le modèle shadowing est le modèle le plus proche de la réalité, puisque plusieurs phénomènes de propagation y sont pris en considération, à savoir, la réflexion, la diffusion et l'absorption. En plus, le rayon de communication n'est plus considéré comme un disque parfait. Le modèle Shadowing se compose en deux parties :

- Le modèle d'atténuation qui prédit la puissance moyenne reçue à une distance d , dénotée $P_r(d)$. Ce modèle utilise une distance de référence d_0 pour le calcul. La puissance moyenne reçue $\bar{P}_r(d)$ est calculée relativement à $P_r(d_0)$ de la manière suivante :

$$\frac{P_r(d_0)}{\bar{P}_r(d)} = \left(\frac{d}{d_0} \right)^\beta \quad (2.3)$$

β est l'exposant du path loss, il s'agit d'une valeur empirique ($\beta = 2$ pour le modèle Free Space). L'atténuation en fonction de la distance est souvent mesurée en dB. Nous avons ainsi :

$$\left[\frac{\bar{P}_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log \left(\frac{d}{d_0} \right) \quad (2.4)$$

- Les variations de la puissance reçue à une distance donnée. C'est une variable suivant une loi log-normale, c'est-à-dire dont la distribution mesurée en dB est gaussienne. Le modèle Shadowing dans son ensemble peut donc être représenté par :

$$\left[\frac{\bar{P}_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log \left(\frac{d}{d_0} \right) + \chi_{dB} \quad (2.5)$$

où χ_{dB} est la variable aléatoire gaussienne dont la moyenne est zéro et l'écart type σ_{dB} . σ_{dB} est appelé shadowing déviation et il est également obtenue par des mesures en environnement réel.

2.4 Modélisation d'un canal radio 802.11

Un module radio ne peut détecter les informations reçues que d'une seule transmission à la fois, dont le signal correspondant doit être reçu avec une puissance supérieure au bruit ambiant. Ainsi, plusieurs facteurs affectent la qualité du lien radio et par suite celle d'une communication à travers ce lien. Par conséquent, une représentation analytique approximative serait plus appropriée. En ce sens, et pour montrer l'efficacité de chaque protocole de routage

on les simule sur deux types de canaux : un canal idéal et un autre basé sur les chaînes de Markov (modèle Gilbert-Elliot). Pour cela, nous avons implémenté le modèle de Gilbert-Elliot dans NS2 où les transmissions (les paquets transmis) sont exposées à un taux d'erreur de façon uniformément distribuée et indépendante. Ainsi, pour caractériser l'état du canal, deux états sont considérés : G -Bon (transmission correcte) et B -Mauvais (transmission erronée). Chaque état est caractérisé par une probabilité d'erreurs au niveau bit (BER pour Bit Error Rate), une probabilité de transition P_{xy} vers le deuxième état ainsi qu'une probabilité de rester dans le même état (figure 2.2). Lorsque le canal se trouve dans l'état Bon, toutes les trames sont correctes dans le cas contraire toutes les trames sont invalides ou perdues. La matrice de transition ainsi associée au modèle est définie par :

$$\begin{bmatrix} 1-r & r \\ p & 1-p \end{bmatrix} \quad (2.6)$$

Où :

$$p = P(e_t = B \mid e_{t-1} = G) \text{ et } r = P(e_t = G \mid e_{t-1} = B) \quad (2.7)$$

Les probabilités stationnaires π_G et π_B associées aux états G et B sont définies respectivement par :

$$\pi_G = \frac{r}{p+r} \quad (2.8)$$

$$\pi_B = \frac{p}{p+r} \quad (2.9)$$

La probabilité de pertes π_p est indiquée par l'équation 2.10 :

$$\pi_p = p * \pi_G + (1-r) * \pi_B \quad (2.10)$$

2.5 Modèle de mobilité

Dans un réseau mobile Ad Hoc les nœuds mobiles peuvent être en mouvement, comme ils peuvent être immobiles. Afin d'évaluer les performances des protocoles de routages, ces derniers doivent être testés sous plusieurs modèles de mobilité.

Plusieurs modèles synthétiques sont définis dans la littérature afin de simuler le comporte-

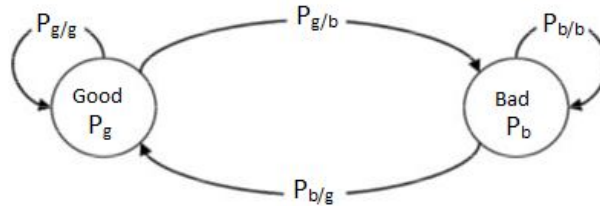


FIGURE 2.2: Diagramme d'état de transition du modèle de Gilbert-Elliot

ment des nœuds mobiles. Dans les paragraphes suivants, nous allons donc étudier les différents modèles de mobilité utilisés dans les environnements de simulation.

2.5.1 Modèles sans mémoire

Dans ces modèles, un nœud choisit une position et une vitesse d'une façon absolument aléatoire et sans aucune mémoire du passé. Dans ces approches, on peut avoir fréquemment des comportements extrêmes des nœuds comme une accélération soudaine et un arrêt soudain.

Les principaux modèles existants dans cette catégorie sont :

2.5.1.1 Random Walk

Le Random Walk (RW) a été introduit pour la première fois par Einstein en 1926 [45]. Dans ce modèle un nœud mobile se déplace de son emplacement actuel vers un nouvel emplacement en choisissant aléatoirement une direction et une vitesse de déplacement. Ce vecteur est modifié périodiquement et la nouvelle valeur de ses paramètres est choisie uniformément sur $[V_{min}, V_{max}]$ pour la vitesse et $]0, 2\pi]$ pour la direction. V_{min} et V_{max} sont deux constantes représentant respectivement la vitesse minimale et maximale à laquelle un mobile peut se déplacer. Le modèle de mobilité Random Walk donc est un schéma de mobilité sans mémoire où la vitesse et la direction d'un nœud mobile sont indépendantes de son passé. Un exemple de trajectoire générée à partir de ce modèle est présenté dans la figure 2.3.

2.5.1.2 Random Waypoint

Ce modèle était d'abord utilisé par Johnson et Maltz dans l'évaluation du protocole de routage DSR [47]. Bientôt, il est devenu un modèle de mobilité «repère» pour évaluer les protocoles de routage, en raison de sa simplicité et sa grande disponibilité. Un exemple de trajectoire générée à partir de ce modèle est présenté dans la figure 2.4. Au départ le nœud

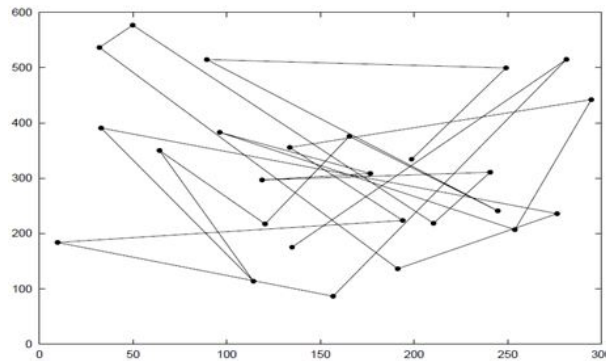


FIGURE 2.3: Trajectoire d'un mobile générée à partir du modèle Random Walk [46]

est statique (immobile) puis il choisit de façon aléatoire un point de destination dans l'espace, ensuite le nœud se déplace vers ce point à une vitesse constante choisie arbitrairement entre une vitesse minimale et maximale déjà prédéfinies. A l'arrivée, le nœud prend un temps de repos avant de choisir à nouveau une nouvelle destination et une nouvelle vitesse et répète le même processus.

2.5.1.3 Modèle Random Direction

Développer par Royer et al [48] le modèle random direction est une amélioration de modèle Random Waypoint. Il a été créé pour éviter l'effet de concentration des nœuds au centre produit par le Random Waypoint, où la probabilité qu'un nœud mobile choisit une nouvelle destination localisée au centre de la surface de la simulation est haute. Le fonctionnement de ce modèle est le suivant :

- Placement de l'ensemble des nœuds dans la zone de simulation en choisissant leur position uniformément répartie suivant les deux directions de l'espace, chaque mobile choisit uniformément une direction dans $[0; 2\pi]$, et une vitesse dans $[V_{min}; V_{max}]$,
- Déplacement des mobiles selon la direction et la vitesse choisies jusqu'à atteindre le bord de la zone de simulation, une fois qu'un mobile ayant atteint cette bordure, il choisit une nouvelle direction avec une nouvelle vitesse après avoir effectué une pause dont la durée de temps est choisie uniformément dans $[T_{min}; T_{max}]$

2.5.2 Modèles avec mémoire

Dans ces modèles, appelés aussi corrélés, la vitesse et la direction des nœuds mobiles à chaque instant, dépendent de l'instant précédent. Parmi les modèles corrélés existants on trouve :

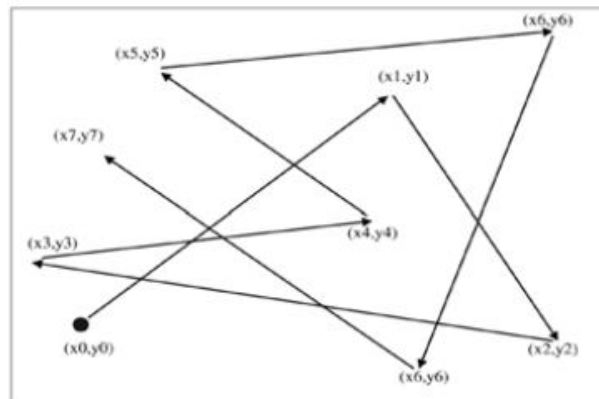


FIGURE 2.4: Trajectoire d'un mobile générée à partir du modèle Random Waypoint

2.5.2.1 Gauss Markov

Le modèle de mobilité Gauss Markov a été proposé dans [49]. C'est un modèle de mobilité dont la position et la vitesse à tout instant, dépendent de la position et la vitesse au moment précédent. La vitesse, la direction et la position d'un nœud varient selon les formules suivantes :

$$\begin{cases} \nu_t^x = \alpha \nu_{t-1}^x + (1 - \alpha) \nu^x + \sigma^x \sqrt{1 - \alpha^2} W_{t-1}^x \\ \nu_t^y = \alpha \nu_{t-1}^y + (1 - \alpha) \nu^y + \sigma^y \sqrt{1 - \alpha^2} W_{t-1}^y \end{cases} \quad (2.11)$$

Avec :

$$\bar{\nu}_t = [\nu_t^x, \nu_t^y]^T \quad (2.12)$$

$$\bar{\nu}_{t-1} = [\nu_{t-1}^x, \nu_{t-1}^y]^T \quad (2.13)$$

sont les vitesses dans t et t-1,

$$\bar{W}_{t-1} = [W_{t-1}^x, W_{t-1}^y]^T \quad (2.14)$$

est le processus Gaussian aléatoire, et

$$\bar{\alpha} = [\alpha^x, \alpha^y]^T \quad (2.15)$$

$$\bar{v} = [v^x, v^y]^T \quad (2.16)$$

$$\bar{\sigma} = [\sigma^x, \sigma^y]^T \quad (2.17)$$

sont les vecteurs de niveau de mémoire. Un effet absolument aléatoire est obtenu en plaçant $\alpha = 0$ (modèle est semblable à Random Walk) et un effet linéaire est obtenu en plaçant $\alpha = 1$ (la vitesse du nœud à l'instant t égale à la vitesse à t-1). Pour d'autres valeurs de α , la vitesse et la direction à chaque instant, dépend de vitesse V_{t-1} et de variable aléatoire de Gausse W_{t-1} de l'instant t-1.

2.5.2.2 Boundless

Dans les modèles de mobilité traditionnels, les nœuds mobiles se réfléchissent sur les bords tendant ainsi à se concentrer au centre. L'intérêt du modèle Boundless [46] réside dans le fait qu'il s'affranchit des bordures de l'aire de simulation. Un nœud qui atteint le bord, continue pour sortir et rentrer de l'autre coté de la surface de simulation. A cause de cet effet de bord, les nœuds dans ce modèle, se déplacent comme si les bords de la simulation n'existaient pas. L'action d'un nœud qui sort et rentre de l'autre côté, est semblable à celle d'un nœud qui sort définitivement de la simulation avec un nouveau qui arrive en même temps pour le remplacer. Les nœuds sont distribués aléatoirement et se déplacent selon l'équation suivante :

$$\begin{cases} x_i = x_{i-1} + v * \cos(d) \\ y_i = y_{i-1} + v * \sin(d) \end{cases} \quad (2.18)$$

Où :

v : la vitesse.
 d : la direction.

2.6 Évaluation des performances par simulation

Dans les sections suivantes, nous commencerons par la description de l'ensemble des paramètres de simulation ainsi que les différentes métriques de performances auxquelles nous nous sommes intéressés. Ensuite, nous présenterons l'ensemble des résultats auxquels nous avons abouti tout en fournissant les interprétations nécessaires.

2.6.1 Environnement de simulation

Pour l'évaluation des performances des protocoles de routage étudiés (DSDV, DSR, OLSR et AODV), on a utilisé le simulateur NS2. Le réseau considéré est composé de 30 nœuds. La topologie dans laquelle les nœuds se déplacent est de 1000m x 1000m et le temps de simulations est mis à 150 secondes.

Dans toutes les simulations on a utilisé les paramètres standards pour le médium et les différents modèles de propagation radio (la bande passante du médium est de 1MB/s et les modèles de propagation radio sont Free space, Two-Ray Ground et Shadowing).

Le protocole d'accès au médium utilisé est IEEE 802.11 et chaque nœud possède une zone de transmission de 250 m. Le type de la queue en chaque nœud est «drop tail». La politique dans ce type de file consiste à mettre en file d'attente les paquets qui arrivent, et à rejeter tous les paquets quand la taille de la file d'attente dépasse la taille du tampon. La probabilité de rejet de Drop Tail $d(k)$ est donnée par l'équation suivante :

$$d(k) = \begin{cases} 0 & \text{si } k \leq K \\ 1 & \text{if } k > K \end{cases} \quad (2.19)$$

Avec :

k : la taille de la file d'attente.
 K : la taille du tampon.

Dans nos simulations on a fixé le nombre de paquets maximum dans le tampon d'émission à 50 paquets.

Le tableau 3.1 résume les paramètres utilisés lors de nos simulations.

Paramètres utilisés pour le modèle de trafic	
Temps de simulation	600s
Type de trafic	CBR
Nombre de connexions	20
Taille de paquets	512 Octet
Paramètres utilisés pour le modèle de mobilité	
Aire de réseau	1000 m x 1000 m
Nombre de noeuds	30
Modèle de mobilité	RWP
Paramètres utilisés pour les couches liaison et physique	
Protocole MAC	IEEE 802.11
Modèle de propagation	-Free space -Two-Ray Ground -Shadowing
Débit	1 Mbps
Type de la File d'attente	DropTail/PriQueue
Taille max des files	50
Modèle d'erreur	-Uniform -Two state (Markov)

TABLE 2.1: Tableau des paramètres de simulation

2.6.2 Critères d'évaluation

Dans l'objectif de comparer les performances des protocoles de routage des réseaux mobiles Ad Hoc les métriques suivantes ont été étudiés :

- Le taux de paquets délivrés ou PDR (Packet Delivery Ratio) : c'est le rapport entre le nombre de paquets reçus et le nombre de paquets émis. Pour calculer ce taux, deux compteurs sont nécessaires. Le premier devrait être incrémenté pour chaque paquet envoyé. Le deuxième devrait être augmenté pour chaque paquet reçu. Finalement le rapport de la livraison de paquet est défini par la formule suivante :

$$PDR = \frac{\sum \text{Paquets de données reçus par la destination}}{\sum \text{Paquets de données émis par la source}} * 100 \quad (2.20)$$

- Le délai de bout en bout : c'est la durée prise par le paquet à partir de nœud source au nœud de destination. Nous calculons ce délai de chaque paquet dans le réseau suivant la formule suivante :

$$\text{Délai bout en bout} = \text{instant de réception de paquet} - \text{instant d'émission de paquet} \quad (2.21)$$

- Charge de routage normalisée (Normalized Routing Load (NRL)) : c'est le ratio entre le nombre de paquets du contrôle générés par le protocole de routage et le nombre de

paquets de données bien reçus.

$$NRL = \frac{\sum \text{Paquets du contrôle générés par le protocole de routage}}{\sum \text{Paquets de données bien reçus}} \quad (2.22)$$

- Le débit : c'est le ratio du nombre total des paquets reçus sur le temps total de la durée de la simulation. L'expression mathématique est la suivante :

$$Throughput = \frac{\sum \text{nombre total des bits reçus}}{\text{la durée de la simulation}} \quad (2.23)$$

Ainsi, pour chaque métrique, nous effectuons les mesures dans les cas suivants :

- Cas 1. Exécution des protocoles sans aucune modification de canal de transmission de NS2.
- Cas 2. Exécution des protocoles avec modification de canal en intégrant la probabilité d'erreur de transmission, caractérisé par les deux états de Markov.

2.7 Résultats de simulations

2.7.1 Taux de paquets délivrés

Les figures 2.5 et 2.6 montrent que les valeurs des taux de délivrance des paquets de AODV et DSR pour les propagations free space et shadowing sont supérieurs à ceux des DSDV et OLSR. Cependant, pour la propagation two-ray ground, le taux DSR devient inférieur à celui du protocole OLSR.

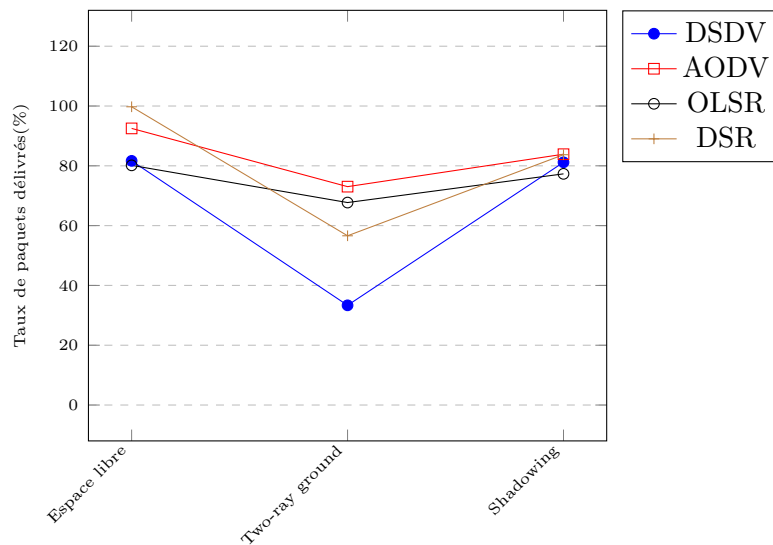


FIGURE 2.5: Taux de paquets délivrés : canal avec modèle Gilbert-Elliot

On remarque aussi que les connexions sans fil entre les nœuds souffrent beaucoup de l'effet des obstacles dans le cas où on utilise un canal réel (simulé par le modèle Gilbert-Elliot), qui cause de nombreuses coupures de liens et par conséquent résulte à un faible taux de délivrance des paquets.

2.7.2 Délai de bout en bout

La figure 2.7 et 2.8 montrent l'évolution de délai de bout en bout des protocoles AODV, DSDV,OLSR et DSR en fonction de l'état du canal et les modèles de propagation radio. Les

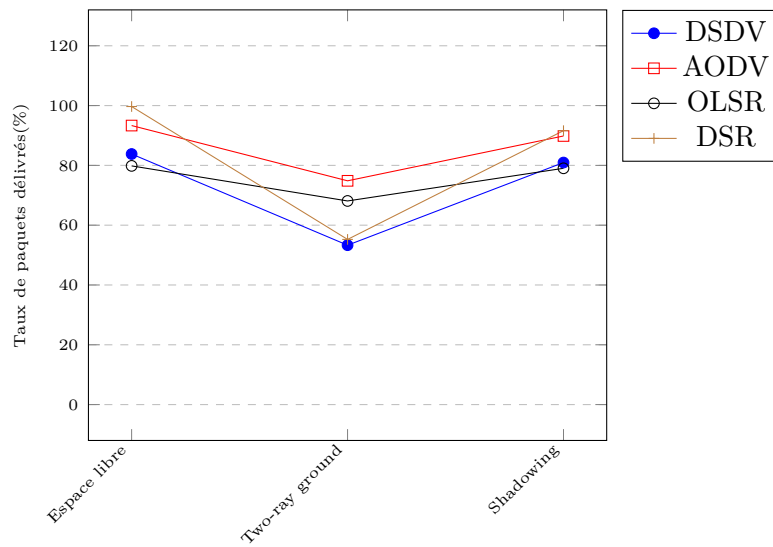


FIGURE 2.6: Taux de paquets délivrés : canal idéal

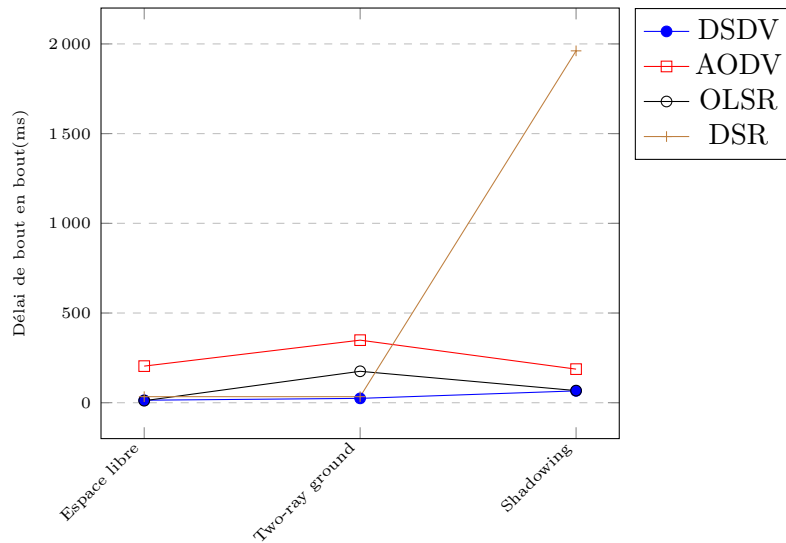


FIGURE 2.7: Délai de bout en bout : canal avec modèle Gilbert-Elliot

résultats de simulations montrent que le modèle Free Space a des délais de bout en bout moins importants par rapport aux autres modèles de propagation (two-ray-ground et shadowing). Cela s'explique par le taux de perte significatif de two-ray-ground et shadowing qui engendre un délai plus important pour la retransmission des paquets perdus, d'où une augmentation dans le délai de bout-en-bout.

Comme prévu, les protocoles OLSR et DSDV possèdent des délais de bout en bout plus faibles comparés aux DSR et AODV. La nature proactive de DSDV et OLSR permet à ces deux protocoles de découvrir rapidement la route optimale, par la suite le temps de transmission des paquets prend moins de temps par rapport aux protocoles réactifs AODV et DSR, ce qui explique des meilleurs performances de DSDV et OLSR en terme de délai de bout en bout.

On remarque aussi que l'intégration du modèle Gilbert-Elliot pendant la simulation augmente le délai de transmission de bout en bout. Cette augmentation est justifiée par le traitement additionnel ajouté dans chaque nœud intermédiaire lors de l'envoi ou la réception des paquets.

2.7.3 Charge de routage normalisée

La charge de routage normalisée est définie comme étant le rapport entre le nombre des paquets de contrôle transmis par tous les nœuds et le nombre de paquets de données livrés aux nœuds de destinations. Cette charge normalisée donne une idée sur l'utilisation des ressources du réseau en terme de nombre de paquets de routage, indépendamment des paquets de données délivrés vers leurs destinations. Cette mesure révèle le degré d'efficacité du protocole de routage. En effet, plus le nombre de paquets de contrôle est élevé, moins le protocole est efficace.

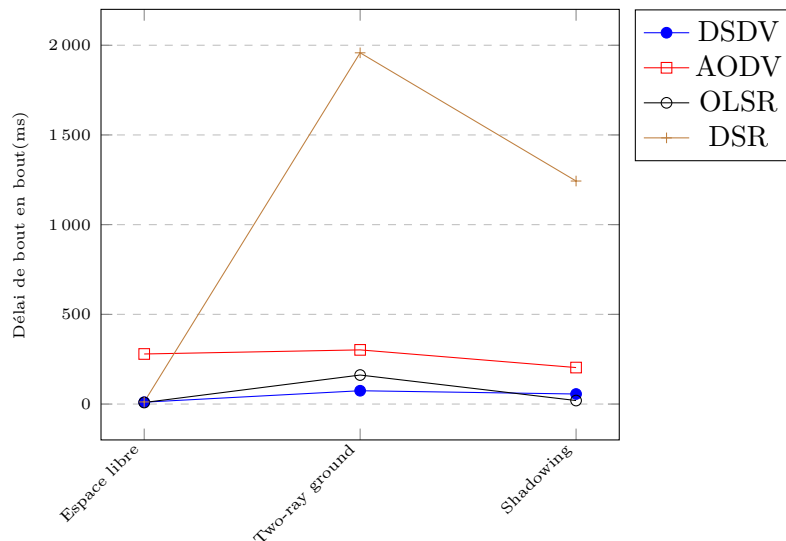


FIGURE 2.8: Délai de bout en bout : canal idéal

En général, nous constatons que DSR et AODV provoquent une surcharge plus importante que DSDV et OLSR (figures 2.9 et 2.10), ce qui est tout à fait normale, car ils ont besoin de déterminer une route valide à chaque fois quand un nœud la demande et en absence d'une qui est active ce qui nécessite une génération et un envoi de plusieurs paquets de contrôles (RREP et RREQ), ce qui augmente le coût de routage d'AODV et DSR. Par contre dans le cas de DSDV et OLSR un nœud connaît déjà les nœuds de destinations à l'aide de l'échange périodique des tables de routages, donc à la demande d'une connexion ou un envoi de message par un nœud, il suffit de choisir le nœud de saut suivant pour atteindre le nœuds de destination si ce dernier n'est pas dans sa zone de couverture.

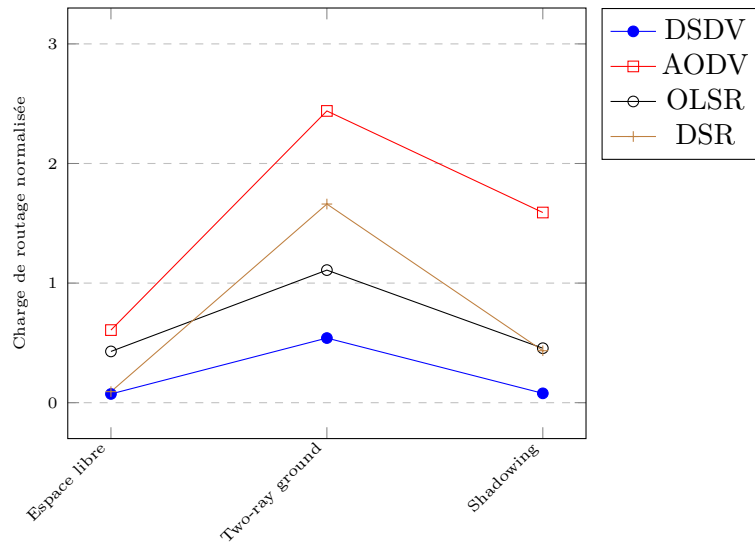


FIGURE 2.9: Charge de routage normalisée : canal avec modèle Gilbert-Elliot

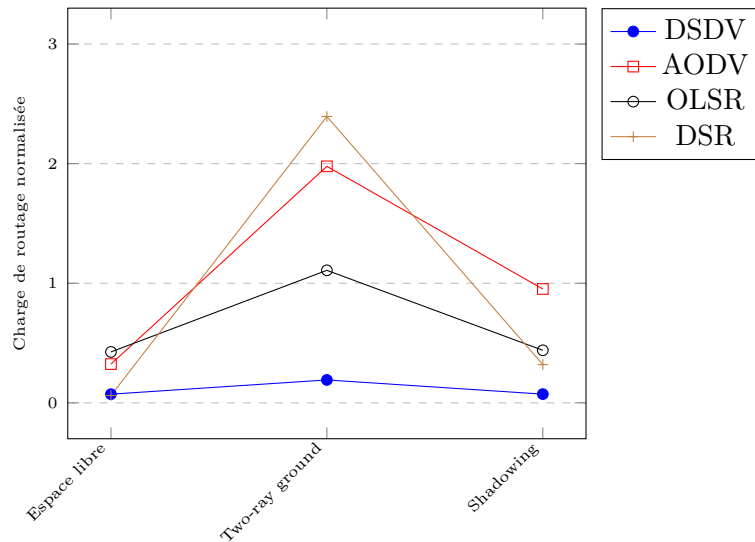


FIGURE 2.10: Charge de routage normalisée : canal idéal

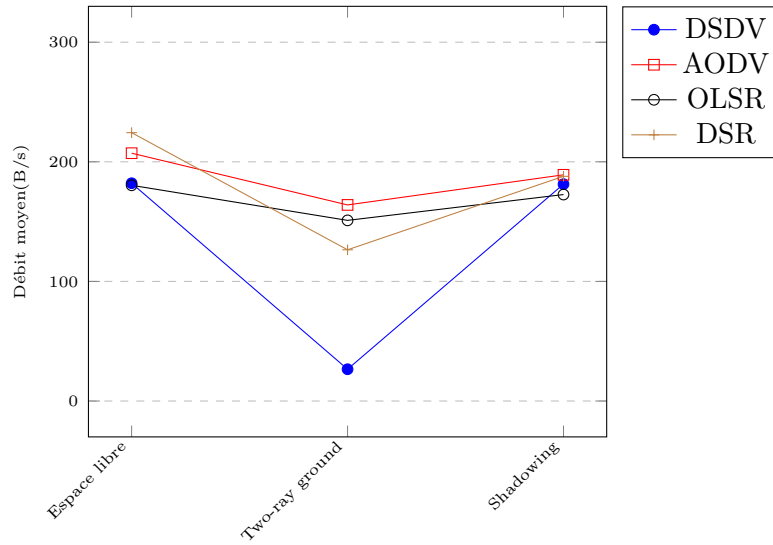


FIGURE 2.11: Débit moyen : canal avec modèle Gilbert-Elliot

Nous pouvons constater aussi à travers ces expérimentations que l'extension qu'on a introduit (implémentation du canal réel) influe directement sur les performances générales des protocoles, elle augmente la charge de routage normalisée par rapport au modèle idéal, puisque dans un environnement réel, la maintenance de l'architecture du réseau et le taux d'erreur des paquets de données élevé génèrent une surcharge de contrôle.

2.7.4 Débit moyen

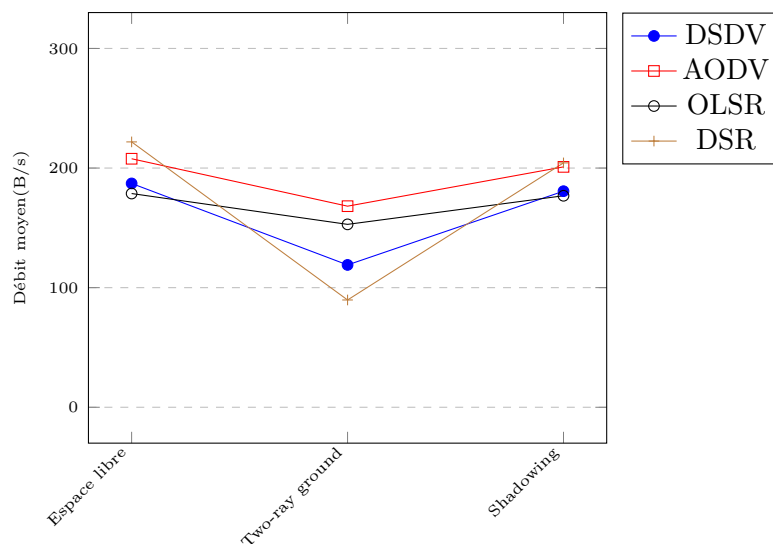


FIGURE 2.12: Débit moyen : canal idéal

En analysant les figures 2.11 et 2.12, on peut constater que la simulation dans un en-

vironnement réel de la couche physique modifie complètement les résultats obtenus dans le cas idéal. De plus, le débit moyen a un impact important sur la qualité des communications dans le réseau mobile Ad Hoc. Cela va influencer le choix du protocole de routage dans la recherche de routes. Il est intéressant de noter que les protocoles AODV, DSR, DSDV et OLSR offrent de bons résultats dans le cas de simulation dans un environnement idéal où y a une faible probabilité d'erreur des paquets de données, donc peu de retransmission des paquets dans le réseau ce qui augmente le débit de la liaison. En résumé, les résultats des simulations effectuées montrent que le protocole AODV assure de bonnes performances en terme de débit moyen par rapport aux autres protocoles, et ce pour toutes les modèles de propagations radio.

2.8 Conclusion

Les performances d'un réseau mobile Ad Hoc sont en effet directement liées aux conditions de propagation des ondes radio entre l'émetteur et le récepteur. Les solutions proposées pour le routage dans les réseaux mobiles Ad Hoc sont évaluées et validées sur des plateformes de simulation, mais les modèles adoptés pour simuler la couche physique est très loin de la représenter dans un environnement de transmission réel. Ainsi, l'évaluation des performances des protocoles de routages proposés sur ces plateformes ne tient pas en compte des conditions réalistes dans lesquelles la transmission évolue, ce qui conduit à une étude de performances limitée et imprécise, qui ne correspond pas aux performances obtenues dans un environnement réel. L'utilisation des scénarios de simulation réalistes est donc un facteur important dans le développement des protocoles de routages Ad Hoc.

Dans ce chapitre nous avons proposé une couche physique qui tient en compte les spécificités liées à l'environnement de transmission réel en intégrant le modèle de Gilbert-Elliot dans NS2. À travers notre étude comparative menée dans ce chapitre, nous avons dégagé les points suivants :

- Il n'y a pas un protocole qui est favori par rapport aux autres dans tous les scénarios et les critères d'évaluation. En effet, chaque protocole a des comportements différents par rapport aux coûts de routage, taux de paquets délivrés, le délai et le débit moyen.
- Dégradation de la performance des quatre protocoles de routages lors de l'intégration de la probabilité d'erreur (transmission caractérisée par le modèle de Markov à deux états) dans le modèle de simulation.
- DSDV et OLSR performant bien en terme de délai de bout en bout par rapport aux protocoles DSR et AODV. Cela est expliqué par leurs caractéristiques proactives.
- AODV et DSR ont des meilleures performances en terme de taux de paquets délivrés lorsqu'on les compare avec OLSR et DSDV.
- Les protocoles DSDV et OLSR performant bien en terme de coût de routage.
- Le débit moyen obtenu par AODV et DSR dépasse largement celui obtenu par les protocoles DSDV et OLSR.

Après l'étude des différents protocoles de routage existants dans les réseaux mobiles Ad Hoc, nous nous sommes intéressés à choisir un. Notre choix s'est porté sur AODV, vue son comportement stable au niveau de la qualité des routes fournit mais aussi en délai de transmission et en débit moyen. Le protocole AODV est le mieux adapté dans notre étude dont le but est la proposition de nouvelles métriques de routage en tenant compte la mobilité, le niveau d'énergie des nœuds, la taille de la file d'attente et le délai de bout en bout.

Optimisation d'énergie dans AODV en environnement réel (E-AODV)

Sommaire

3.1	Introduction	55
3.2	Protocole E-AODV	56
3.2.1	Hypothèses	56
3.2.2	Paquets et structures de données de E-AODV	56
3.3	Description du E-AODV	56
3.3.1	Découverte de chemin dans E-AODV	57
3.3.2	Sélection des routes	58
3.3.3	Maintenance des chemins	61
3.4	Tests et résultats	61
3.4.1	Paramètres de simulation	61
3.4.2	Résultats des simulations	61
3.5	Conclusion	64

3.1 Introduction

La définition la plus connue de la durée de vie d'un réseau est l'intervalle de temps qui sépare l'instant de déploiement de l'instant où le premier nœud épuise son énergie. Chaque nœud doit donc être gérer ses ressources énergétiques afin de maximiser la durée de vie totale du réseau. Ainsi, la consommation d'énergie devrait être une question cruciale lors de la conception de nouveaux protocoles de routage Ad Hoc du fait que les stations du réseau sont en activité permanente (écoute du canal, réception, transmission, routage, traitement de l'information, ...).

Nous proposons, dans ce chapitre, des extensions de l'un des plus importants protocoles de routage actuels qui est AODV (Ad hoc On demand Distance Vector). Ces extensions prennent en compte l'énergie résiduelle des nœuds du réseau lors de découverte de route. Dans notre protocole les nœuds ayant une faible énergie, sont évités de la processus de routage afin de maintenir des valeurs d'énergie pareilles pour tous les mobiles, ainsi garantir une plus longue vie du réseau.

3.2 Protocole E-AODV

Dans cette section nous allons présenter le principe de fonctionnement de notre extension, ainsi que son intégration dans le protocole AODV afin de maximiser la durée de vie du réseau.

3.2.1 Hypothèses

Dans cette contribution on suppose N nœuds distribués aléatoirement. Chaque nœud dans le réseau a un identifiant unique, assez d'énergie pour communiquer et capable de calculer son énergie résiduelle.

3.2.2 Paquets et structures de données de E-AODV

Les nœuds mobiles utilisant E-AODV échangent quatre types de paquets de contrôle, à savoir :

- Les requêtes de chemins (RREQs).
- Les réponses de chemin (RREPs).
- Les erreurs de chemin (RERRs).
- Les requêtes HELLO.

Dans notre étude les mêmes formats de paquets RERR et HELLO définis dans [3] pour le protocole AODV sont réutilisés dans E-AODV, tandis que les formats de paquets RREQ et RREP ont été modifié et augmenté par les champs :

- "RE" dont la valeur représente les énergies résiduelles des nœuds parcourus soit par le paquet RREQ, soit par RREP.
- "LB" qui maintient les adresses des nœuds intermédiaires.

3.3 Description du E-AODV

Le protocole E-AODV est constitué de trois phases :

- Découverte des chemins.
- Sélection des chemins.
- Maintenance des chemins.

3.3.1 Découverte de chemin dans E-AODV

3.3.1.1 Initialisation de la demande de route

La procédure de la découverte des chemins dans E-AODV est similaire à celle de l'AODV. Si le nœud source a besoin d'un chemin vers une destination alors qu'aucun n'est déjà stocké dans sa table de routage ou s'il possède un chemin obsolète, le nœud doit initier une découverte de route. Pour ce faire, il diffuse une demande de route RREQ.

Afin d'éviter le problème de chevauchement des chemins, les nœuds intermédiaires dans E-AODV ne suppriment pas toutes les copies des paquets RREQ comme dans AODV mais transfère ceux provenant sur des liens différents.

Dans AODV un nœud intermédiaire peut aussi décider de renvoyer un message RREP vers la source s'il dispose un chemin avec un numéro de séquence supérieur à celui connu de cette dernière. Mais dans notre extension, seul le nœud de destination doit répondre à un paquet RREQ en mettant le drapeau D (*destination_only*) à 1 dans la demande de route. De cette façon plusieurs routes conservant une certaine diversité peuvent parvenir à la destination.

3.3.1.2 Propagation de la demande de route

Dans notre protocole chaque nœud intermédiaire recevant la demande de route la retransmet après avoir effectué des mises à jour concernant le paquet RREQ. Par exemple lorsqu'un nœud intermédiaire reçoit un paquet RREQ d'un autre nœud, il ajoute son énergie résiduelle et son adresse respectivement dans les champs RE et LB. Le même processus est répété jusqu'à ce que le paquet RREQ atteigne sa destination.

Quand un nœud reçoit un message RREQ dans un réseau utilise AODV, il a l'obligation de le rediffuser vers la destination. Dans E-AODV chaque nœud détermine à partir de son énergie résiduelle E_r , s'il va accepter la rediffusion des paquets RREQ. Si cette valeur est supérieure à un seuil x fixé le paquet de requête RREQ est rediffusé, mais dans le cas contraire, le paquet est rejeté. De cette façon la destination recevra un paquet de requête seulement lorsque tous les nœuds intermédiaires ayant de bons niveaux d'énergie.

3.3.1.3 Propagation de la réponse de route

Chaque nœud intermédiaire recevant une réponse de route doit la transférer au prochain saut en direction de la source après avoir incrémenté le nombre de sauts, insérer son énergie résiduelle et son adresse respectivement dans les champs RE et LB du paquet RREP. Le prochain saut peut être trouvé en consultant sa table de routage qu'il a construit lors du passage de la demande de route (RREQ). Le passage de la réponse de route donne aussi lieu à des mises à jour de la table de routage des nœuds intermédiaires en utilisant les informations contenues dans le paquet RREP.

3.3.2 Sélection des routes

3.3.2.1 Mécanisme de l'E-AODV

À la réception d'une demande de route pour la première fois, le nœud ne la traite pas mais associe à cette demande un timer d'attente. Il enregistre ensuite toutes les demandes de route similaires (identifiées par le même couple [ID, @SRC]) dans sa table historique. Ce timer d'attente est nécessaire afin de recevoir tous les paquets RREQ provenant des autres voisins avant les traités. Pour illustrer ce processus, la figure 3.1 montre le cheminement de la demande de route initialisée par la source jusqu'à ce qu'elle atteigne la destination. Dans cette figure, il y a sept nœuds {S, 1, 2, 3, 4, 5, 6, D}, le nœud S est la source et D la destination. La valeur à l'intérieur d'un nœud représente son énergie résiduelle, nous inscrivons aussi une valeur à côté de chaque arc (i, j) qui représente le coût du lien pour la transmission des données du nœud j au nœud S. Dans notre extension, chaque liaison est proportionnelle au coût relatif au nombre de saut H, et inversement proportionnelle aux coûts relatifs à la moyenne d'énergie résiduelle entre deux nœuds voisins ($mean(EN_i, EN_j)$). On peut écrire alors :

$$C_{lien} = \frac{H}{mean(EN_i, EN_j)} \quad (3.1)$$

Où :

- H : Nombre de saut du nœud j vers le nœud source.
- EN_i : Energie résiduelle du nœud i.
- EN_j : Energie résiduelle du nœud j.

Une fois le timer expire, le nœud applique l'algorithme de Floyd-Warshall pour calculer les meilleurs chemins vers les autres nœuds mentionnés dans le champ LB.

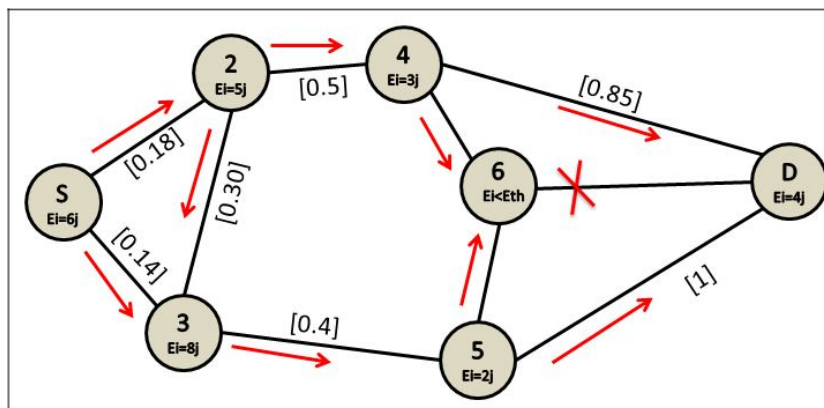


FIGURE 3.1: Découverte de routes dans E-AODV.

3.3.2.2 Calcul du coût d'une route

Le protocole E-AODV est basé sur AODV, donc son principe de fonctionnement consiste à résoudre le problème de la recherche d'une route p , à un instant de découverte de route t , de telle sorte que la fonction coût utilisé par ce protocole soit réduite au minimum. Lorsque l'on va choisir une route on prend en compte le minimum de saut possible et également l'énergie encore disponible des nœuds par lesquels vont être traversées les données.

Le calcul du coût d'une route dans E-AODV est donc conditionné par deux facteurs :

- L'énergie résiduelle des nœuds constituant le chemin.
- Le minimum de saut possible.

Chaque liaison sans fil est donc annotée avec le coût de la liaison en termes d'énergie résiduelle et le nombre de saut vers la source (équation 3.1). Ces informations nous permettent de construire un graphe valué (G) .

Notre objectif dans cette contribution, ne consiste pas de chercher un chemin d'un nœud vers un autre, mais l'ensemble des chemins minimaux entre chaque couple de nœuds. C'est pourquoi on a utilisé l'algorithme de Floyd-Warshall (voir Algorithm 1) pour calculer le plus court chemin entre toute paire de nœuds. Donc, on note $D = (d(i, j))(i, j) \in [0, n]^2$ la matrice des distances dans le graphe G . A l'initialisation, on calcule le plus court chemin entre toute paire de sommets n'ayant pas de sommets intermédiaires, donc il suffit de prendre la longueur des arcs qui existent et mettre un poids infini si l'arc n'existe pas :

$$D[i, j]^0 = \begin{cases} 0 & \text{si } i = j \\ \frac{H}{\text{mean}(EN_i, EN_j)} & \text{si } i \neq j \text{ et } (i, j) \in E \\ \infty & \text{si } i \neq j \text{ et } (i, j) \notin E \end{cases} \quad (3.2)$$

d vérifie l'inégalité triangulaire :

$$\forall (i, j, k) \in G^3 \quad d(i, k) \leq d(i, j) + d(j, k) \quad (3.3)$$

Par la suite, si on note $d^k(i, j)$ la valeur du plus court chemin de i à j dont les seuls sommets intermédiaires sont dans l'ensemble $\{1, 2, \dots, k\}$, alors on a l'égalité suivante :

$$d^k(i, j) = \min \left(D[i, j]^{k-1}, D[i, k]^{k-1} + D[k, j]^{k-1} \right) \quad (3.4)$$

En effet, si on considère deux sommets i et j :

- Soit le plus court chemin entre i et j parmi ceux visitant $\{1, \dots, k+1\}$ ne passe pas par le sommet $k+1$, et alors :

$$d^{k+1}(i, j) = d^k(i, j) \quad (3.5)$$

Chapitre 3. Optimisation d'énergie dans AODV en environnement réel (E-AODV)

- Soit ce plus court chemin passe par le sommet $k + 1$. Dans ce cas il ne peut y passer qu'une seule fois (sinon il y a une boucle). Ce plus court chemin est le résultat de la concaténation du plus court chemin entre i et $k + 1$ et du plus court chemin entre $k + 1$ et j . On a alors :

$$d^{k+1}(i, j) = d^k(i, k + 1) + d^k(k + 1, j) \quad (3.6)$$

L'algorithme précédent se contente de calculer les plus courts chemins mais ne garde pas la trace des nœuds intermédiaire. Ces derniers peuvent être stockés dans une matrice annexe P . Cette matrice contient la liste des sommets intermédiaires par lesquels on doit passer pour obtenir le chemin le plus court. $P[i, j]^k$ représente ici le prédécesseur du sommet j dans le plus court chemin de i à j n'utilisant comme sommets intermédiaires que des sommets parmi $\{1, \dots, k\}$. Au début et pour $k = 0$, un plus court chemin ne possède aucun sommet intermédiaire, donc :

$$P[i, j]^0 = \begin{cases} null & \text{si } (i, j) \notin E \text{ ou } i = j \\ i & \text{si } (i, j) \in E \text{ et } i \neq j \end{cases} \quad (3.7)$$

Dans le cas général, si le plus court chemin est de la forme $i \rightarrow k \rightarrow j$ le prédécesseur de j est le même que celui du plus court chemin de k à j et n'utilisant comme sommets intermédiaires que des sommets parmi $\{1, \dots, k-1\}$. Autrement, on prend le même prédécesseur de j que celui qui se trouvait sur le plus court chemin de i à j et n'utilisant comme sommets intermédiaires que des sommets parmi $\{1, \dots, k-1\}$. Nous avons donc, dans tous les cas :

$$p[i, j]^k = \begin{cases} p[i, j]^{k-1} & \text{si } D[i, j]^{k-1} \leq D[i, k]^{k-1} + D[k, j]^{k-1} \\ p[k, j]^{k-1} & \text{si } D[i, j]^{k-1} > D[i, k]^{k-1} + D[k, j]^{k-1} \end{cases} \quad (3.8)$$

Algorithm 1 Floyd-Warshall

```

for all node pairs  $[i, j] \in NXN$  do
 $D[i, j] \leftarrow \infty$  and  $p[i, j] \leftarrow 0$ 
for all nodes  $i \in N$  do  $D[i, i] \leftarrow 0$ 
foreach arc  $(i, j) \in E$  do  $D[i, j] \leftarrow \frac{H}{\text{mean}(EN_i, EN_j)}$  and  $p[i, j] \leftarrow i$ 
for  $k \leftarrow 1$  to  $N$  do
foreach  $[i, j] \in NXN$  do
if  $D[i, j] > D[i, k] + D[k, j]$  then
begin
 $D[i, j] \leftarrow D[i, k] + D[k, j]$ 
 $p[i, j] \leftarrow p[k, j]$ 
end;
end;

```

3.3.3 Maintenance des chemins

Comme dans AODV, le protocole E-AODV utilise les messages HELLO pour maintenir les informations de connectivité. À chaque intervalle de temps (`hello_interval`), chaque nœud diffuse un message HELLO pour préciser aux voisins qu'il est toujours à portée radio. Ce message est destiné au voisinage direct d'où l'initialisation du TTL (Time-To-Live) à 1 dans l'entête IP. À la réception d'un message HELLO, le nœud vérifie s'il a une entrée pour le voisin duquel le message est reçu dans sa table de voisinage pour la mettre à jour ; sinon il enrichit son voisinage d'un nouveau voisin et ajoute ainsi l'entrée correspondante dans sa table de routage. Si la connectivité à un voisin ne peut pas être déterminée, le nœud doit supposer que le lien est perdu et une réparation de route doit être faite. Pour ce faire le protocole E-AODV utilise le paquet RERR. Les nœuds recevant ce paquet retirent de leurs tables de routages toutes les entrées qui utilisent ce lien et le nœud source initie une nouvelle procédure de découverte de routes une fois reçoit ce paquet de contrôle.

3.4 Tests et résultats

3.4.1 Paramètres de simulation

L'objectif des simulations est d'évaluer les performances d'E-AODV et les comparer avec AODV. Plus particulièrement, nous voulons montrer que E-AODV permet de réduire considérablement, par rapport à AODV les paquets de contrôle dans le réseau, augmenter le taux de livraison de paquets et le nombre des nœuds qui ont une énergie résiduelle non nul à la fin de la simulation (augmentation de la durée de vie du réseau). Nous exécutons les simulations dans un terrain de $500*500 m^2$. Le nombre de nœuds mobiles varie entre 35 et 60 nœuds disposés aléatoirement. Le Protocole MAC utilisé est le 802.11 alors que le protocole de transport est UDP. Le temps de simulation est fixé à 10 minutes et la bande passante du canal sans fil est fixée à 1Mb/s. Le modèle de trafic utilisé est constant bit rate (CBR). Le tableau 3.1 résume les paramètres utilisés dans nos simulations.

3.4.2 Résultats des simulations

3.4.2.1 Charge de routage normalisée

Ce paramètre donne une idée sur la consommation de la bande passante par les paquets de contrôle par rapport aux paquets utiles de données.

Dans un premier temps, nous varions la densité du réseau (figure 3.2(a)). Comme c'était attendu, la charge de routage des deux protocoles augmente avec l'augmentation de la densité. On remarque que quel que soit la densité, la charge du E-AODV était inférieure que celui du AODV, parce que E-AODV réinitie moins fréquemment les découvertes de chemins grâce à

Chapitre 3. Optimisation d'énergie dans AODV en environnement réel (E-AODV)

Paramètres utilisés pour le modèle de trafic	
Temps de simulation	600s
Type de trafic	CBR
Nombre de connexions	20
Taille de paquets	512 Octet
Paramètres utilisés pour le modèle de mobilité	
Aire de réseau	500 m x 500 m
Nombre de noeuds	35-40-45-50-55-60
Modèle de mobilité	RWP
Paramètres utilisés pour les couches liaison et physique	
Protocole MAC	IEEE 802.11
Modèle de propagation	Two-Ray Ground
Débit	1 Mbps
Type de la File d'attente	DropTail/PriQueue
Taille max des files	50
Modèle d'erreur	-Uniform -Two state (Markov)

TABLE 3.1: Paramètres utilisés dans les simulations

l'utilisation de routes qui représentent des énergies résiduelles importantes (routes stables). Nous constatons aussi qu'AODV génère plus de paquets de routage sous des conditions d'une forte mobilité par rapport à E-AODV (figure 3.2(b)). Cela est dû au nombre important des paquets RREQs générés par AODV en réinitialisant à chaque fois des découvertes de routes.

Dans des scénarios d'une faible mobilité, les chemins deviennent de plus en plus robustes ce qui diminue le besoin d'initialisation des découvertes de chemins pour les deux protocoles. Cependant, AODV continue à générer un surdébit important puisque à n'importe quel moment un nœud intermédiaire peut épuiser son énergie ce qui oblige le protocole d'initialiser la découverte de chemin.

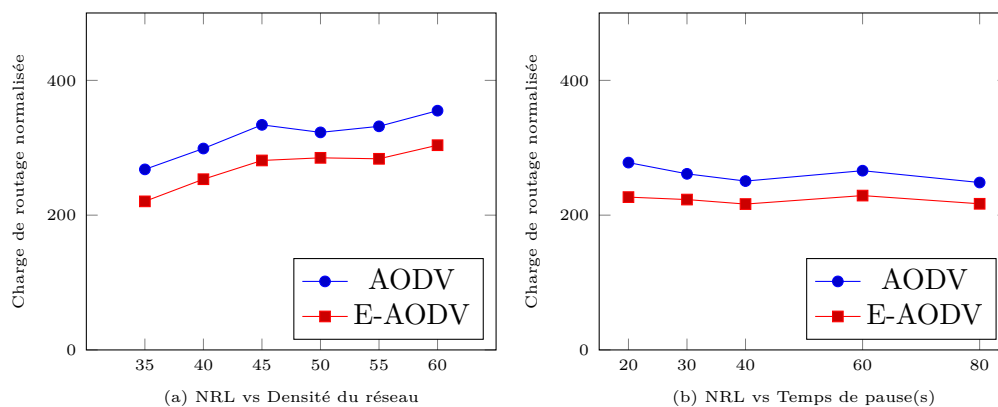


FIGURE 3.2: Charge de routage normalisée (NRL) vs densité et la mobilité des nœuds

3.4.2.2 Taux de paquets délivrés

La figure 3.3 montre que le taux de paquets délivrés augmente relativement avec l'augmentation du nombre de nœuds. En effet l'augmentation du nombre de nœuds conduit à la surcharge du réseau, ce qui favorise les diversités des chemins de routage. Les chemins optimisés en nombre de sauts et énergie résiduelle d'E-AODV augmentent le taux de livraison de paquets par rapport aux chemins d'AODV.

L'augmentation du temps de pause conduit à la stabilité du réseau, ce qui favorise l'arrivée des paquets de données à leur destination, si pour cette raison que le taux de paquets délivrés augmente. Pour des temps de pauses minimaux, on constate que E-AODV présente de meilleurs résultats qu'AODV, puis avec l'augmentation du temps de pause les deux protocoles se rejoignent et offrent les mêmes résultats. Nous concluons donc que l'implémentation de notre nouvelle approche n'est totalement réussie que pour une forte mobilité.

3.4.2.3 Énergie consommée

L'efficacité de la politique de distribution de la charge implémentée dans E-AODV est claire dans tous les cas de simulations (figure 3.4), où l'énergie consommée par notre protocole était inférieure à celui de l'AODV. Ce résultat est justifié par l'augmentation du surdébit d'AODV par rapport à notre approche.

3.4.2.4 Nombre de nœuds vivants

La figure 3.5 représente le nombre de nœuds vivants sur un total de nœuds varie de 35 à 60 et un temps de pause de 20 à 60s. Grâce à l'intégration de l'énergie résiduelle dans la qualité de lien entre deux voisins, E-AODV favorise les nœuds disposant d'une plus grande énergie résiduelle. A partir des résultats obtenus, on constate que notre approche à réaliser une augmentation de 23% de la durée de vie moyenne d'un nœud, et par conséquent augmente

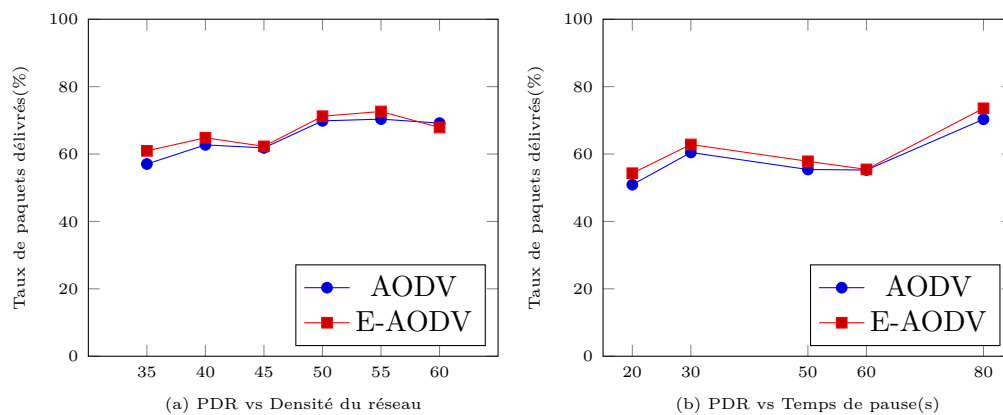


FIGURE 3.3: Taux de paquets délivrés vs densité et la mobilité des nœuds

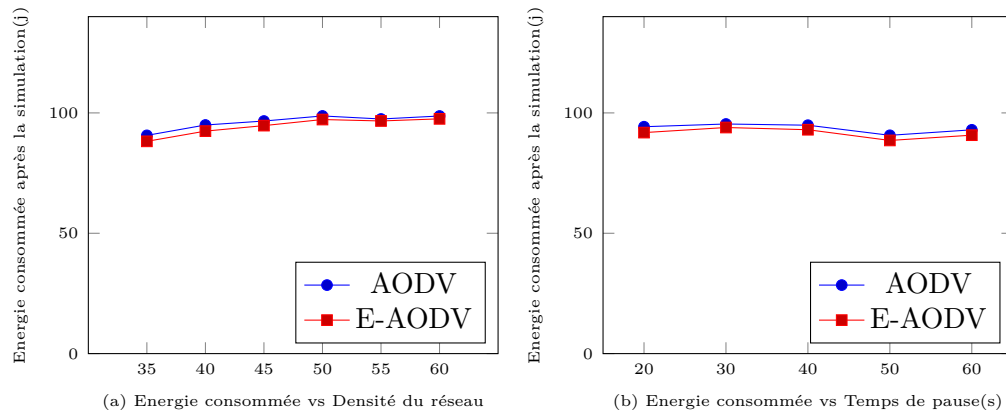


FIGURE 3.4: Energie consommée après la simulation vs densité et la mobilité des nœuds

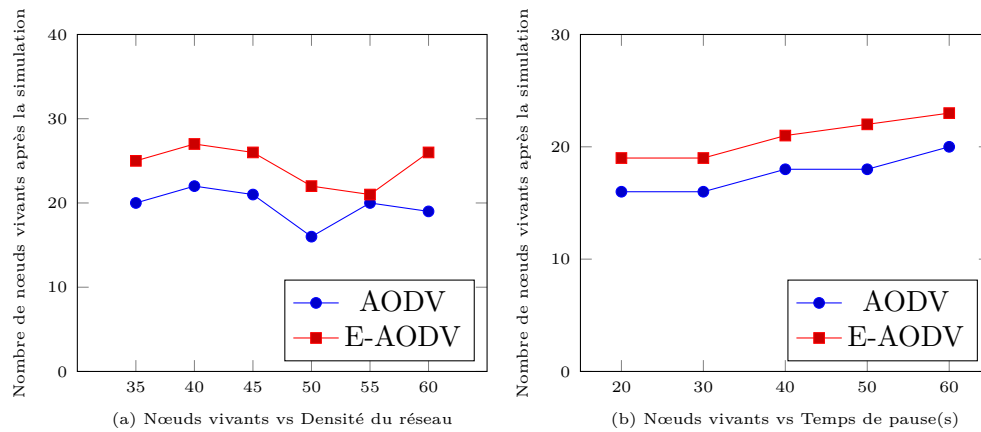


FIGURE 3.5: Nombre de nœuds vivants après la simulation vs densité et la mobilité des nœuds

la durée de vie de tout le réseau, ce qui répond bien à nos attentes de la modification du protocole standard (AODV).

3.5 Conclusion

Vu la consommation ininterrompue en énergie des nœuds du réseau mobile Ad Hoc et afin d'augmenter la durée de vie de ce dernier, plusieurs approches ont été proposées pour réduire les sources de gaspillage d'énergie. Le routage orienté conservation d'énergie essaye de réduire la consommation d'énergie en choisissant les nœuds les plus appropriés pour la communication. Dans cet axe, notre travail a porté sur la conception d'une nouvelle approche pour l'économie de l'énergie dans les réseaux mobiles Ad Hoc au niveau de la couche réseau. L'approche que nous avons proposée est basée sur le rapport entre l'énergie résiduelle des nœuds et le nombre de saut qui sépare l'émetteur et le récepteur. Ce rapport permet d'équilibrer les mobiles du réseau, ainsi un mobile disposant d'une grande quantité d'énergie a une grande chance d'être élu comme nœud intermédiaire pour assurer la communication entre deux nœuds.

Chapitre 3. Optimisation d'énergie dans AODV en environnement réel (E-AODV)

Les expérimentations effectuées permettent néanmoins de montrer clairement le gain apporté par l'utilisation de notre approche.

protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

Sommaire

4.1	Introduction	66
4.2	Stratégie et outils d'implémentation	67
4.3	Description de la solution proposée	67
4.4	Architecture Cross-layer du protocole EMCMI-AODV	68
4.4.1	Approche au niveau de la couche physique	68
4.4.2	Approche au niveau de la couche MAC	71
4.4.3	Approche au niveau de la couche réseau	73
4.4.4	Sélection des chemins	76
4.5	Tests et résultats	80
4.5.1	Paramètres de simulation	80
4.5.2	Résultats de simulation	80
4.6	Conclusion	83

4.1 Introduction

Dans les réseaux mobiles Ad Hoc, les communications se font sur un support radio électrique basé sur des ondes électromagnétiques libres dans l'espace. Comme ce support est partagé sur tous les usagers du réseau, cela peut causer des interférences entre les signaux émis simultanément par les différents nœuds qui utilisent le même canal radio. Aussi, bien que l'accès au canal de transmission soit géré par des techniques d'accès comme le CSMA ; cependant, la collision entre les paquets persiste toujours, et influe gravement sur la qualité de service (QoS) du réseau. Tout cela explique l'importance de l'utilisation de la technologie multi-interfaces multi-canaux (MIMC) puisqu'elle réduit la contention, la probabilité de collision et fournit des transmissions simultanées dans une même zone de transmission sans chevauchement dans différents canaux améliorant ainsi considérablement le débit.

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

Dans ce chapitre nous proposons un nouveau système qui, premièrement, offre une large gamme de débits suffisamment élevés et des taux d'erreur faibles. Deuxièmement, Il augmente la durée de vie du réseau.

A la fin de ce chapitre, nous évaluons par simulation les performances de notre système et nous le comparons à d'autres systèmes proposés dans la littérature.

4.2 Stratégie et outils d'implémentation

Lorsqu'un nœud intermédiaire utilise un seul canal pour relayer les paquets d'un autre nœud, il est obligé d'attendre d'avoir reçu toutes les données avant de commencer à envoyer le premier paquet vers le prochain nœud, parce qu'une seule interface réseau ne peut pas fonctionner simultanément en réception et en émission, c'est pourquoi la capacité des nœuds intermédiaires est divisée par deux pour assurer la bi-direction. Par contre avec un réseau multicanaux, un nœud peut envoyer et recevoir des données en même temps.

Dans le cadre de notre approche, nous nous intéressons à trouver une solution Cross-layer qui permet d'améliorer les performances du réseau mobile Ad hoc en partageant l'information concernant les états des canaux radio et le niveau de batterie des nœuds entre la couche physique, Mac et réseau. Les niveaux de batterie sont mesurés afin de devenir disponible ou accessible aux couches supérieures.

La solution cross-layer envisagée est une solution à deux niveaux, elle vise en premier lieu à modifier le comportement du protocole de routage AODV et en deuxième lieu à améliorer le fonctionnement de la couche Mac 802.11 et physique en utilisant la technologie MIMC.

4.3 Description de la solution proposée

Notre solution permet de développer un nouveau système de routage nommé EMCMI-AODV. En effet, le protocole AODV utilise la métrique de nombre de sauts comme métrique de base lors du choix des routes mais ne tenant pas compte d'autres paramètres de QoS. La notion du nombre de saut donc n'est pas optimale à cause des problèmes de propagation c'est pour cela d'autres critères de mesure devraient être donnés aux routes choisies. Nous avons proposé donc une solution cross-layer basée sur l'interaction des couches Physique, Mac et Réseau. EMCMI-AODV obtient en premier temps les énergies résiduelles des nœuds du réseau à travers les paquets de contrôle (RREQ ou RREP) et stocke ses valeurs par la suite dans des tables de nœuds voisins. Ces tables sont utilisées par la couche réseau afin de construire des chemins de routage économiques en énergie et ayant une forte connectivité.

La métrique d'énergie a déjà été utilisée dans plusieurs travaux de recherche sauf que son utilisation est basée sur une couche physique simple (SISC) pour transmettre et recevoir des données, dans notre approche nous modifions le protocole AODV pour s'adapter au modèle

multi-interfaces multicanaux (MIMC). Ainsi, le protocole EMCMI-AODV est composé d'un sous protocole de routage au niveau de la couche réseau, un sous protocole d'accès au média multi-interfaces multicanaux au niveau de la couche Mac et un sous protocole de contrôle d'énergie au niveau de la couche physique.

4.4 Architecture Cross-layer du protocole EMCMI-AODV

L'architecture Cross-layer proposée dans notre contribution garde une compatibilité avec la pile protocolaire d'OSI en ajoutant une entité intermédiaire via laquelle les interactions entre les couches non adjacentes passent [50][51][52]. Le point fort de cette approche réside dans le fait de garder une compatibilité arrière avec la pile protocolaire en incorporant des extensions évolutives contenant de nouvelles fonctionnalités du cross-layer. La figure 4.1 illustre notre architecture Cross-layer. Au niveau de la couche réseau, notre nouveau protocole de routage établit la table d'énergies résiduelles des nœuds du réseau et calcule le meilleur chemin en se basant sur les contraintes de cette dernière.

Dans la couche Mac, notre protocole introduit des procédures pour gérer les interfaces, la sélection du canal et récupérer l'information de toutes les interfaces définies pour un nœud donné[33]. EMCMI-AODV permet donc aux différents nœuds de transmettre parallèlement sur des canaux distincts sans collision, ce qui permet d'augmenter le débit, la durée de vie du réseau et potentiellement réduire les délais de transmission.

Au niveau de la couche physique EMCMI-AODV on a développé un mécanisme dédié au calcul de l'énergie résiduelle des nœuds et un contrôle efficace d'énergie de transmission radio ce qui permet aussi de prolonger la durée de vie du réseau.

4.4.1 Approche au niveau de la couche physique

Afin d'augmenter le nombre de transmissions radio, on a ajouté des interfaces physiques radios à chaque nœud du réseau, ce qui permettra d'utiliser plusieurs canaux en même temps, de transmettre les données plus rapidement et d'éviter la rupture de chemin et l'arrêt de transmission.

Comme par défaut dans NS2, un nœud présente une seule interface (figure 4.2), nos mo-

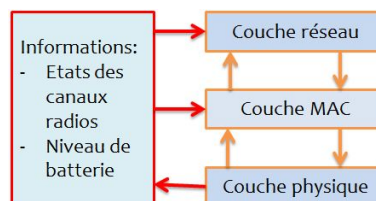


FIGURE 4.1: Partage de l'information entre les trois couches physique, Mac et Réseau.

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

difications seront portées sur le canal de propagation d'une manière à obtenir la modélisation d'un système multi-interfaces multicanaux.

Où

- Routing Agent : (couche 3 du modèle OSI) définit le protocole de routage qui sera utilisé dans le réseau.
- Link Layer(LL) : simule les protocoles de la couche liaison de données et il est directement connecté à un module ARP qui permet de connaître l'adresse physique d'une carte réseau correspondant à une adresse IP.
- MAC (Media Access Control) : indique le type de protocole d'accès multiple utilisé. Il Traite des paquets reçus ou envoyés à partir de « Link Layer ».
- Interface Queue (file d'interface) : représente la file d'attente.
- Network Interface (PHY) : comme une interface de matériel, pour accéder au canal.
- Channel : Simule les effets de canal réel sans fils dans le signal transféré.

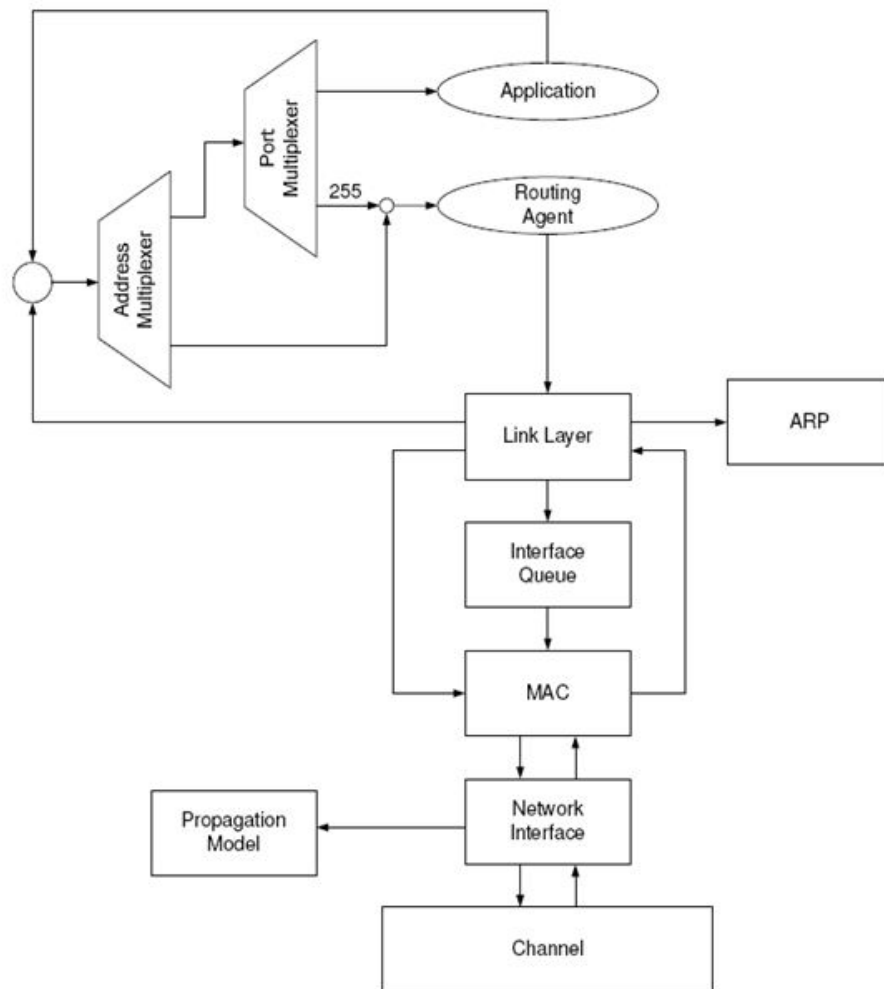


FIGURE 4.2: Architecture d'un nœud mobile dans NS2.

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

L'adaptation d'un nœud multi-interface nous a obligés à apporter des modifications au niveau de la couche physique afin de permettre l'implémentation de protocole de routage EMCMI-AODV. Ces modifications nous permettront aussi d'implémenter la solution multi-interface sur NS2 et de pouvoir la simuler. Pour ce faire, on a modifié le modèle MobileNode existant dans NS2. La figure 4.3 illustre le schéma de la nouvelle architecture d'un nœud mobile utilisé dans la technologie MIMC.

La différence entre les deux architectures est que dans la nouvelle architecture chaque nœud a des formes des couches de lien, ARP, files d'interface, MAC, interface du réseau et des entités de canal. Chaque forme représente une interface du réseau sans fils. D'autre part, la nouvelle architecture ne change pas le matériel IEEE 802.11. En conséquence, il est possible d'implémenter et de simuler la nouvelle architecture dans la réalité.

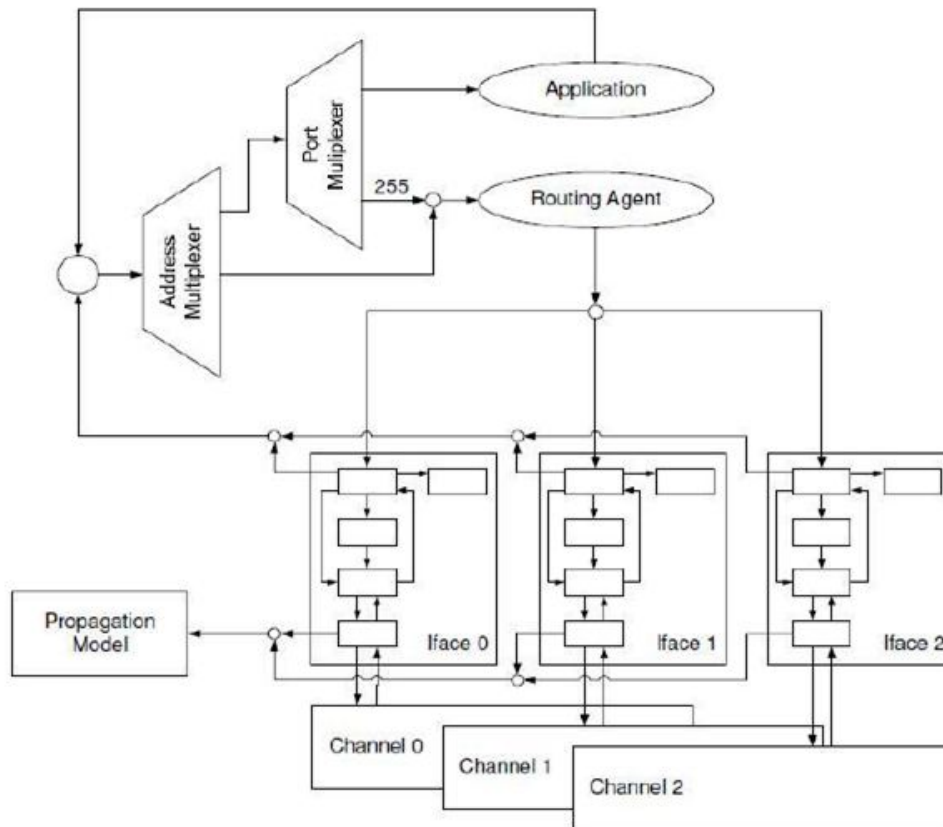


FIGURE 4.3: Nouvelle architecture d'un nœud mobile dans NS2

4.4.2 Approche au niveau de la couche MAC

4.4.2.1 Méthode d'accès multi-interface multicanaux

Nous allons présenter dans cette section la méthode d'accès multi-interfaces multicanaux utilisée dans notre approche où on considère que chaque nœud peut avoir une ou plusieurs interfaces radio physiques. Pour ce faire, les nœuds mobiles doivent être équipés par plusieurs interfaces 802.11 ou doivent posséder la capacité à changer dynamiquement leur fréquence d'opération. Dans notre solution on suppose que le nombre des canaux est supérieur au nombre des interfaces, ce qui est le cas le plus courant et le plus complexe à régler.

4.4.2.2 Principe de fonctionnement de la méthode d'accès multi-interface multicanaux

Pour faire des réseaux où sont utilisés différents canaux il faut gérer le mécanisme d'assignation de canaux et éviter d'avoir les mêmes problèmes cités précédemment (nœuds cachés, collisions, ...). Donc on peut diviser le principe de fonctionnement de cette méthode en deux étapes :

4.4.2.2.1 Gestion de l'interface fixe

Au début, chaque nœud choisit un canal fixe parmi les canaux disponibles d'une manière aléatoire. Ensuite, chaque nœud transmet périodiquement un paquet (Hello) contenant son canal fixe utilisé et une table appelée TableVoisins qui regroupe les canaux fixes utilisés par les autres nœuds voisins.

La gestion de l'interface fixe implique donc le choix d'un canal à attribuer à l'interface fixe de façon permanente et d'informer les voisins que ce canal est utilisé par cette interface. Par exemple dans la figure 4.4, le canal 2 est utilisé par l'interface fixe du nœud 4, alors ce dernier doit informer ses voisins qu'il utilise ce canal parce que toutes les transmissions vers ce nœud vont se faire sur le canal 2.

Quand un nœud reçoit un paquet Hello, il ajoute les canaux fixes utilisés par ses voisins dans sa TableVoisins (figure 4.5).

Pour équilibrer l'utilisation des canaux, l'approche consiste à obliger les nœuds voisins d'utiliser différents canaux pour leurs interfaces fixes. Si deux nœuds voisins emploient le même canal fixe, alors l'un des deux nœuds change son canal fixe vers un autre d'une probabilité p . Ensuite, ce nœud transmet des paquets Hello à ses voisins pour leur informer du nouveau canal fixe choisi (figure 4.6).

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

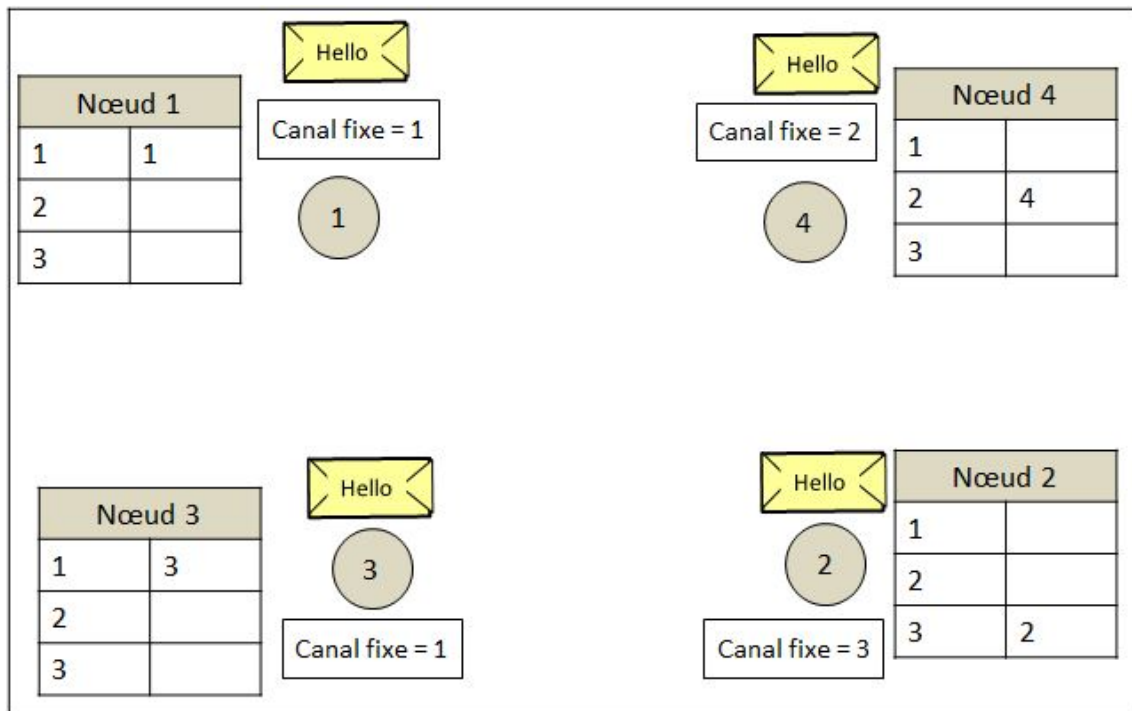


FIGURE 4.4: Paquet Hello diffusé indiquant le canal fixe utilisé par les nœuds voisins

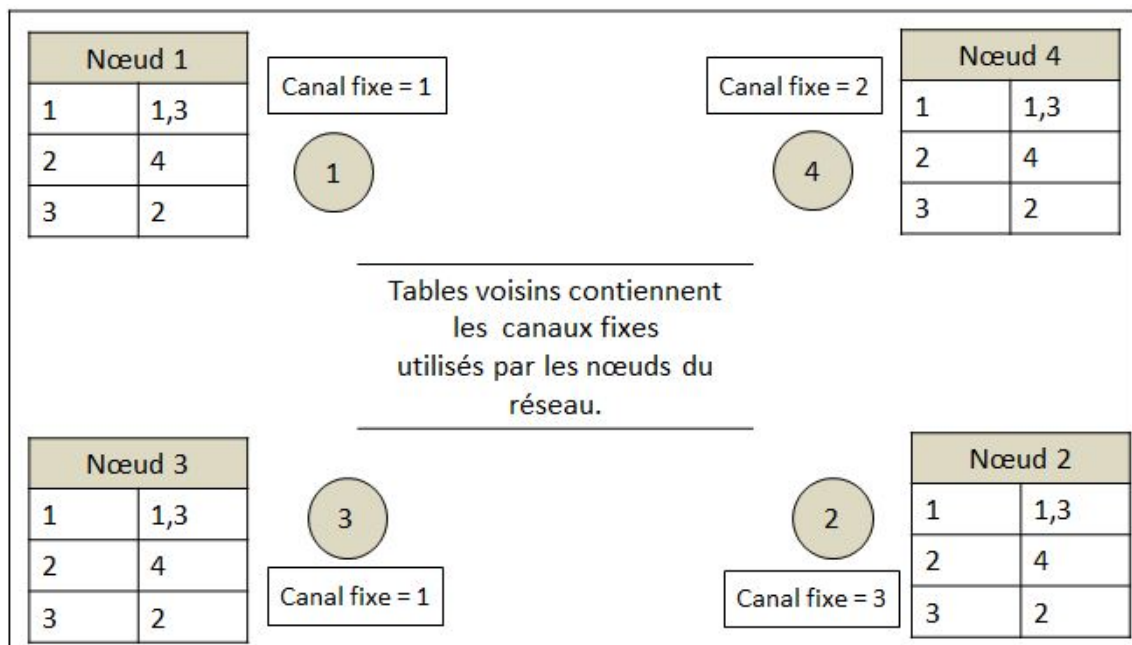


FIGURE 4.5: Etat de convergence du tablevoisins

4.4.2.2 Gestion de l'interface commutable

Après avoir sélectionné l'interface fixe, on va développer un mécanisme de gestion pour affecter les interfaces commutables aux autres canaux. Dans notre contribution les nœuds doivent commuter le canal avant chaque transmission. Puisque on a besoin d'une coordination entre l'émetteur et le récepteur afin d'être sur le même canal en même temps. Chaque canal est donc associé à une file d'attente comme le montre la figure 4.7. Si un nœud veut envoyer des données à un autre, le canal fixe de ce dernier est cherché dans TableVoisins, puis les paquets de données sont insérés dans la file d'attente correspondante.

En général si l'émetteur a le même canal fixe que le récepteur, les paquets sont placés dans la file du canal fixe. Sinon ceux-ci est placés dans la file d'un canal appartenant à l'un des interfaces commutables.

4.4.3 Approche au niveau de la couche réseau

Notre solution cross-layer proposée nous a permet de construire un nouvel algorithme de routage dont le fonctionnement sera basé sur des informations provenant de la couche physique et MAC. Le but principal de cette coopération entre la couche physique, MAC et réseau est d'établir une communication fiable dans un environnement Ad Hoc tout en répondant au besoin de qualité de service. Un chemin serait alors choisi, non plus sur la base du nombre de sauts, mais sur la base de la qualité de ses liens. Le résultat de cette approche est un nouveau protocole nommé EMCMI-AODV. Comme son nom l'indique, EMCMI-AODV se

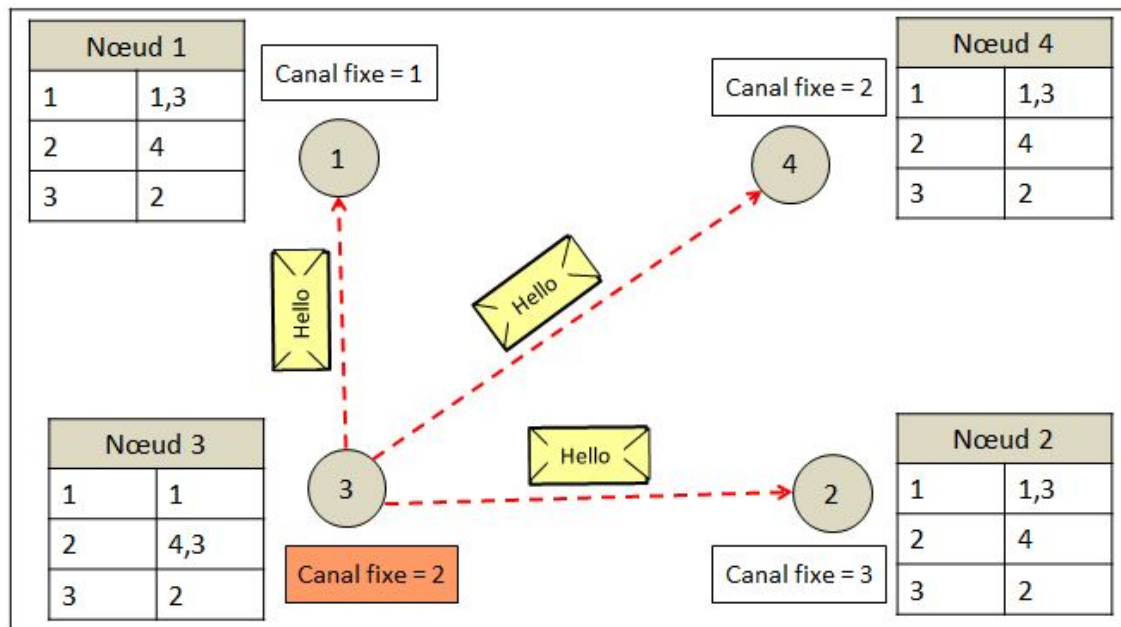


FIGURE 4.6: Nœud 3 change son canal fixe à 2 puis diffuse l'information aux voisins pour qu'ils mettent à jour leurs tables

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

base essentiellement sur le fonctionnement d'AODV ainsi que sur l'utilisation de l'énergie résiduelle des nœuds du réseau pour le calcul d'une nouvelle métrique de routage.

4.4.3.1 Mécanisme de propagation de la demande de route

4.4.3.1.1 Initialisation de la demande de route

Avant d'envoyer un message à la destination, le nœud source vérifie l'existence d'une route valide dans sa table de routage. Ainsi, s'il ne connaît pas la destination ou s'il possède un chemin obsolète vers ce dernier, le nœud doit initier une découverte de route. Pour ce faire, il diffuse une demande de route RREQ à tous ses nœuds voisins. Dans notre protocole seuls les nœuds destinations doivent répondre à un paquet RREQ car il sera difficile de contrôler les chemins qui ont des meilleures énergies si les nœuds intermédiaires répondent de leurs caches. Pour cela, la source doit imposer que seulement la destination lui réponde en mettant le drapeau D (destination_only) à 1 dans la demande de route. Ce bit impose à tous les nœuds intermédiaires de rediffuser la demande de route vers la destination et de ne pas répondre avec une RREP même s'ils ont déjà un chemin valide vers le nœud de destination. Pour éviter le problème de chevauchement des chemins dans notre extension, les nœuds intermédiaires ne suppriment pas toutes les copies des paquets RREQ comme dans AODV mais transfère ceux provenant sur des liens différents (comme le cas dans notre protocole E-AODV).

4.4.3.1.2 Inondation par TTLs croissants

Pour assurer le routage dans un réseau mobile Ad Hoc dont on ignore la topologie, la recherche par diffusion propose de retransmettre récursivement la requête de recherche de route à tous les voisins direct jusqu'à la localisation du nœud de destination. Mais la mise

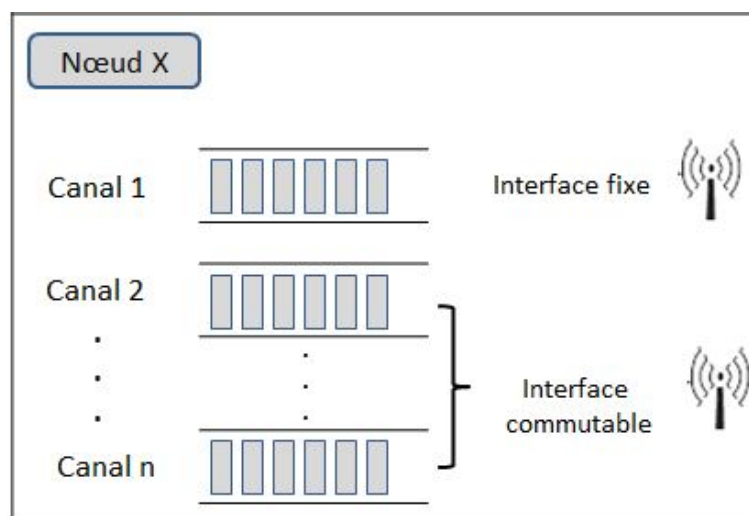


FIGURE 4.7: Files d'attente du nœud pour chaque canal

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

en place d'une route par diffusion peut être coûteuse et provoque des délais importants avant l'ouverture de la connexion entre les nœuds. Dans notre contribution nous avons tenté d'optimiser le coût de l'inondation en proposant une méthode de recherche basée sur l'inondation par TTLs croissants. La source tente ainsi une suite d'inondations en augmentant leur TTL jusqu'à recevoir une réponse de la destination.

4.4.3.1.3 Utilisation d'un seuil d'énergie

Dans le protocole AODV, le nœud a l'obligation de transférer des paquets pour d'autres nœuds quel que soit son énergie résiduelle. Le principe de notre protocole EMCMI-AODV est de déterminer si un nœud participe au processus de découverte de route ou pas selon son énergie résiduelle E_r .

À la réception d'un paquet RREQ, le nœud mesure son énergie résiduelle E_r et remonte l'information à la couche réseau. Cette dernière compare systématiquement les mesures envoyées par la couche MAC à un seuil Th . Quand E_r est plus important que la valeur du Th , le nœud avance le message RREQ, mais dans le cas contraire, il écarte le message et refuse de participer à relayer des paquets (figure 4.8), de cette façon la destination recevra le paquet RREQ seulement lorsque tous les nœuds intermédiaire ont des bonnes niveaux d'énergie.

4.4.3.2 Métriques

De façon générale la couche réseau n'est pas concernée par la manipulation d'information reliée à l'énergie résiduelle du nœud, mais quand cette information lui est disponible, elle la permet de connaître la qualité du lien et la sélection des chemins pour la formation de routes fiables. Notre protocole utilise donc une fonction poids pour déterminer la qualité des chemins. Cette fonction utilise deux sortes de métriques (l'énergie résiduelle et le nombre de sauts minimum).

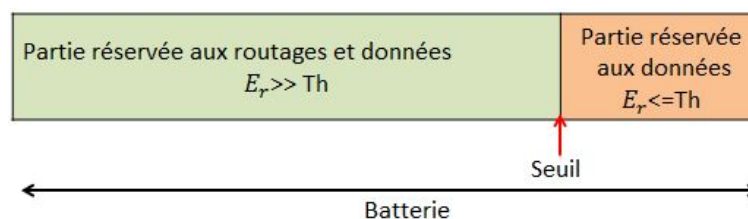


FIGURE 4.8: Gestion de batterie d'un nœud

4.4.4 Sélection des chemins

4.4.4.1 Modèle et hypothèses

On modélise le réseau mobile Ad Hoc par un graphe $G(V, E)$, V représente l'ensemble des nœuds du réseau et E dénote l'ensemble des arcs correspondants aux communications directes possibles.

Soit u et v deux nœuds de V , l'arc (u, v) existe, cela veut dire que les nœuds u et v sont en mesure de communiquer directement. Dans la figure 4.9, il y a sept nœuds $\{1, 2, 3, 4, 5, 6, 7\}$, le nœud 1 est la source et 7 la destination. La valeur à l'intérieur d'un nœud représente son énergie résiduelle. Nous inscrivons aussi une valeur à côté de chaque arc (i, j) qui représente le coût du lien pour la transmission des données du nœud j au nœud 1 avec :

$$C_{lien} = \frac{H}{mean(EN_i, EN_j)} \quad (4.1)$$

Où :

H : Nombre de saut depuis le nœud j vers le nœud source.

EN_i : Energie résiduelle du nœud i .

EN_j : Energie résiduelle du nœud j .

4.4.4.2 Calcul des routes et acheminement des données

Basé sur AODV, le protocole (EMCMI-AODV) apporte des modifications dans l'entête des paquets de routage échangés ainsi que dans les tables de routage maintenues par l'ensemble des nœuds du réseau. Cette tâche est réalisée par la diffusion de message RREQ sur une adresse de

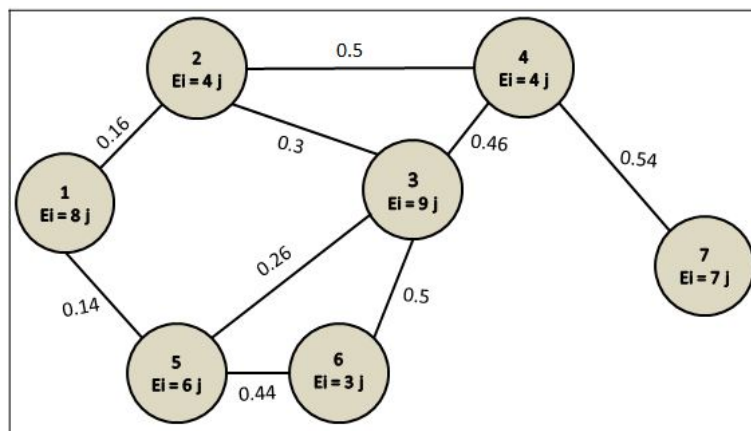


FIGURE 4.9: Modélisation d'un réseau mobile Ad Hoc en utilisant le protocole EMCMI-AODV

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

diffusion à travers le réseau. Pour propager les caractéristiques du chemin emprunté, les entêtes des paquets RREQ et RREP sont modifiés de la même façon qu'E-AODV. Les paquets RREQ et RREP possèdent donc dans leurs entêtes des paramètres déjà présents dans le protocole AODV plus la métrique liée à l'énergie et les contraintes à respecter.

Puisque le nouveau protocole EMCMI-AODV est inspiré de notre protocole E-AODV, il garde la plupart des mécanismes de sélection des routes avec des modifications pendant la sélection des canaux et interfaces pour l'acheminement des données. Chaque nœud utilise EMCMI-AODV comme protocole de routage maintient trois tables : la table de routage , table des énergies résiduelles et table de nœuds et leurs distances deux à deux comme le montre la figure 4.10 :

4.4.4.2.1 Table de routage

La table de routage de notre protocole (EMCMI-AODV) consiste à indiquer à chaque nœud le saut suivant pour acheminer les données vers la destination voulue, en tenant compte l'énergie résiduelle des nœuds intermédiaires. Elle est composée de champs suivants :

- Champ numéro de séquence : permet d'utiliser les routes les plus nouvelles, un nœud le mis à jour chaque fois qu'une nouvelle information provenant d'un message RREQ,

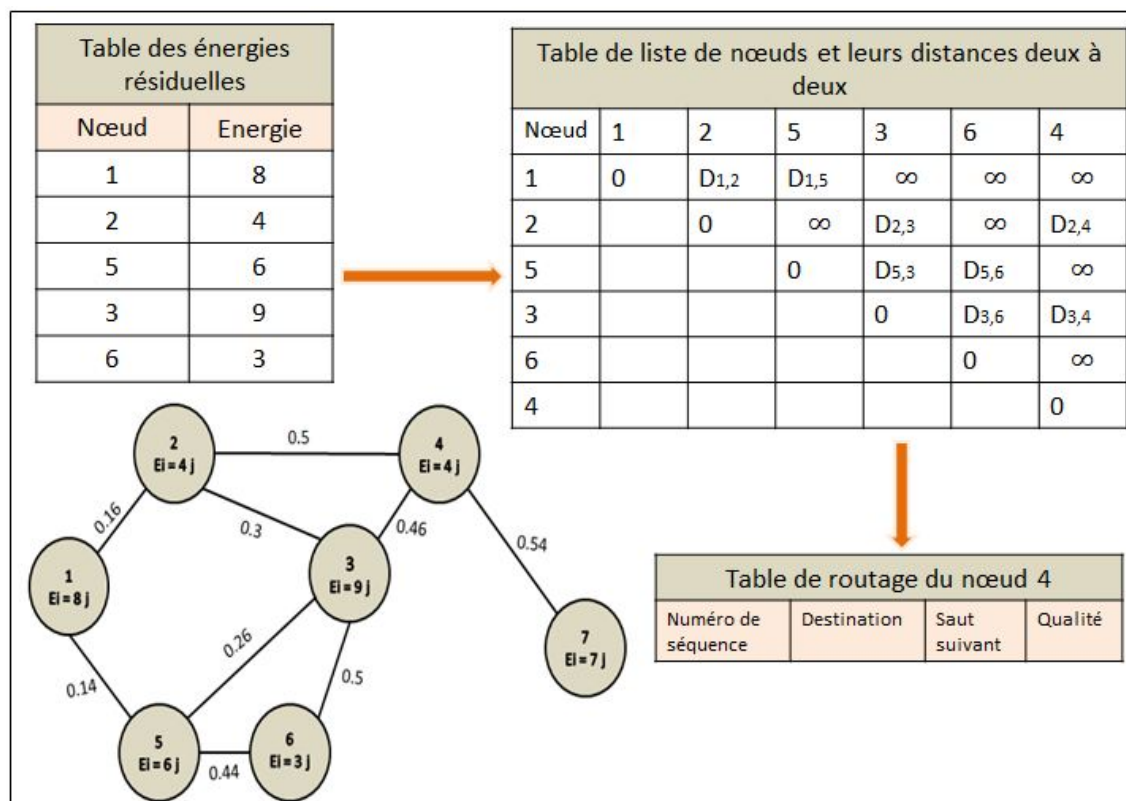


FIGURE 4.10: Structure des tables au niveau du nœud 4

RREP ou RERR, il contient donc la dernière valeur connue associée au nœud destination.

- Destination : indique toutes les destinations fournies par EMCMI-AODV à partir du nœud courant.
- Saut suivant : indique le nœud suivant à prendre pour une destination donnée.
- Qualité : indique la qualité de la route en termes d'énergie et le nombre de saut vers une destination donnée.

4.4.4.2.2 Table d'énergies résiduelles

Au fur et à mesure que la RREQ ou RREP avance vers le nœud de destination ou le nœud source (cas de RREP), les nœuds collectent les informations concernant les adresses et les énergies résiduelles des nœuds visités par ces paquets et les mettent dans la table des énergies résiduelles.

4.4.4.2.3 Table de liste de nœuds et leurs distances deux à deux

c'est dans cette table que chaque nœud enregistre les distances $D(i,j)$ pour chaque paire de nœuds du réseau. Où $D(i,j)$ représente la distance de la route reliant directement le nœud i à j . On suppose par convention que :

$$D(i, j) = \begin{cases} \infty & \text{si } i \neq j \text{ et } (i, j) \notin E \\ 0 & \text{si } i = j \end{cases} \quad (4.2)$$

4.4.4.3 Parcours de réseau pour calculer le chemin le plus court

On considère dans cette partie que chaque arête (lien entre deux nœuds voisins) du réseau est associé un coût positif. Le coût d'un chemin $p = \langle v_0, v_1, v_2, v_3, \dots, v_k \rangle$ est égal à la somme des coûts des arcs composant le chemin, c'est-à-dire :

$$\text{coût}(p) = \sum_{i=1}^k \text{coût}(v_{i-1}, v_i) \quad (4.3)$$

Le coût d'un plus court chemin entre deux sommets v_i et v_j noté $\delta(v_i, v_j)$ est défini par :

$$\delta(v_i, v_j) = \begin{cases} \infty & \text{s'il n'y a pas de chemin entre } v_i \text{ et } v_j \\ \min(\text{coût}(p)) & \text{sinon} \end{cases} \quad (4.4)$$

Où p est un chemin de v_i à v_j .

Note : notre protocole se base sur l'algorithme distribué de Bellman Ford ou Floyd-Warshall pour calculer la longueur des plus courts chemins pour chaque couple de nœuds du

réseau.

4.4.4.3.1 Algorithme de Bellman-Ford

Puisque dans notre approche on veut calculer les plus courts chemins ayant une unique source (nœud qui a reçu un message de contrôle RREQ ou RREP) vers tous les autres nœuds, on peut utiliser dans un premier temps l'algorithme de Bellman-Ford. Sa complexité est en $O(|V| \times |E|)$, où V est l'ensemble des nœuds du réseau et E l'ensemble des arcs.

Au début de notre algorithme, le poids de la source est initialisé à 0, et celui des autres sommets à $+\infty$. Puis, $|V|-1$ itérations sont réalisées, regardant à chaque fois tous les arcs et mettant à jour si nécessaire le poids des sommets depuis la source (voir Algorithm 2).

4.4.4.3.2 Algorithme Floyd-Warshall algorithm

L'idée est un peu similaire. Mais au lieu de faire varier la longueur des chemins, on va faire varier les sommets intermédiaires autorisés, on s'intéresse d'abord aux chemins sans sommet intermédiaire, puis de sommet intermédiaire éventuel 1, puis 1 et 2... (Même principe d'E-AODV). $D_k(i, j)$ sera le poids minimal d'un chemin de i à j n'utilisant que des sommets de numéro au plus k . Les distances sont donc non pas en termes de nombres d'arcs d'un chemin comme dans le cas classique, mais en termes de somme des valeurs de chaque arc qui

Algorithm 2 Bellman-Ford

Données

G un graphe valué

s un sommet de G

Variables locales :

L tableau des distances depuis s /* indexé par les sommets */

(u,v) un arc

début

initialisation

$L[s] \leftarrow 0$

pour tous les sommet $v \neq s$ faire $L[v] \leftarrow \infty$

tant que L change faire

/ relaxation de l'arc (u,v) */*

pour chaque arc (u,v) de G faire

si $L[v] > L[u] + \text{length}((u,v,G))$ alors

$L[v] \leftarrow L[u] + \text{length}((u,v,G))$

finsi

finprch

fintq

end;

Sorties : L , le tableau des distances des plus courts chemins de s

le compose(voir algorithm 1 chapitre 5).

4.5 Tests et résultats

4.5.1 Paramètres de simulation

Dans le cadre de ce travail, l'évaluation des performances des protocoles de routage EMCMI-AODV Floyed-Warshall et EMCMI-AODV Bellman-Ford sera abordée par des simulations sous NS2. Pour notre simulation, nous considérons des sources de type UDP, CBR et un modèle de mobilité 'random waypoint'. Les scenarios utilisent une surface de simulation égale à : 1000m x 1000m.

Les critères de performance évalués sont :

- Charge normalisée du routage (Normalizing routing overhead).
- Taux de perte de données.
- Durée de vie du réseau.

Chaque simulation est exécutée pendant 150 secondes et répétée 10 fois. Nous comparons nos protocole avec EAODV-pri [40] , EAODV-Min[53] et l'implémentation de multi-interfaces multicanaux dans AODV standard (on va le nommé Original AODV-MCMI), on évalue ces performances en augmentant le nombre de nœuds et le temps de pause afin d'étudier l'impact de la densité et la mobilité sur le bon fonctionnement des protocoles de routage.

4.5.2 Résultats de simulation

4.5.2.1 Charge normalisée du routage

Pour étudier l'impact de la charge du trafic et la mobilité sur les protocoles de routage, nous avons fait varier le nombre des nœuds du réseau de 5 à 25 et le temps de pause de 10 à 100ms. Nous remarquons, dans ces résultats (figure 4.11 et 4.12) l'importance nette et significative de l'overhead généré par Original AODV-MCMI, EAODV-Pri et EAODV-Min par rapport à celui de nos protocoles (EMCMI-AODV Floyed-Warshall et EMCMI-AODV Bellman-Ford) et il s'accroît avec l'augmentation du nombre des nœuds. En effet, nos approches permettent de limiter le nombre de messages RREQ diffusés dans le réseau grâce à l'utilisation de l'inondation par TTLs croissants et le seuil d'énergie.

Nous constatons aussi que tous les protocoles performant mieux quand la mobilité est faible. Quand la mobilité croît, le taux de charge normalisée du routage croît aussi pour tous les protocoles. La mobilité des nœuds engendre directement un changement dans la topologie du réseau et par la suite une recherche d'une nouvelle route est nécessaire. Le protocole performe alors suivant le mécanisme de maintenance de route utilisé.

4.5.2.2 Paquets perdus

On va s'intéresser à travers cette série de simulations à comparer les performances de nos extensions avec les performances d'Original AODV-MIMC, EAODV-Pri et EAODV, en termes de taux de perte. La figure 4.13 trace le taux de perte en fonction de la densité et le temps de pause des nœuds du réseau. Nous remarquons que pour une faible densité les cinq protocoles donnent un taux de perte élevé. En effet, comme la connectivité du réseau est faible étant donné que la densité est faible, les protocoles ne trouvent pas des routes pour atteindre certaines destinations qui conduisent à la saturation des files d'attente (buffers) au niveau des nœuds.

Pour des scénarios de mobilité plus basse (réseau stable), le nombre de paquets perdus dans le cas d'utilisation de nos approches s'augmente pour qu'il dépasse ceux des autres

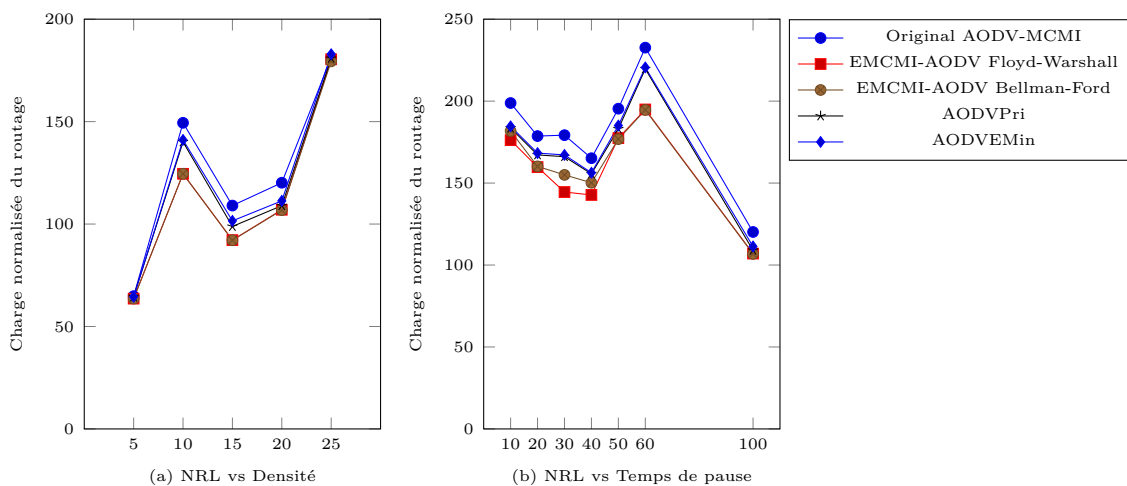


FIGURE 4.11: Charge normalisée du routage vs densité et la mobilité des nœuds

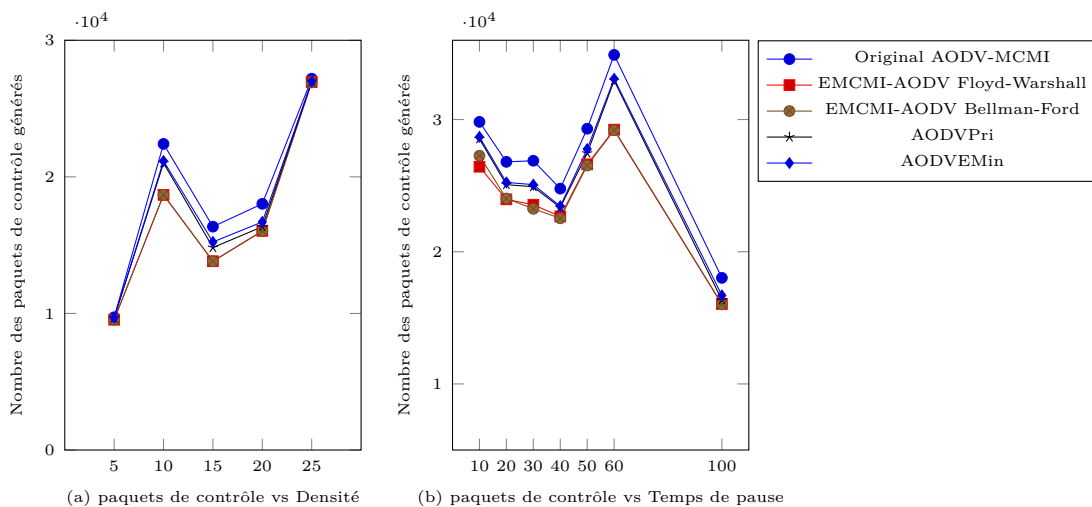


FIGURE 4.12: Nombre des paquets de contrôle générés vs densité et la mobilité des nœuds

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

protocoles. Ce comportement peut être expliqué par le fait que dans Original AODV-MIMC, EAODV-Pri et EAODV-Min les nœuds intermédiaires sont autorisés à répondre directement par des chemins stockés dans leurs caches, donc la probabilité de suppression des paquets de ces protocoles reste inférieur à celle de nos approches.

Mais pour des scénarios d'une forte mobilité, on constate que nos extensions permettent d'obtenir un nombre inférieur de paquets perdus, puisque les routes dans les caches des autres protocoles devient invalides ce qui augmente la probabilité de suppression des paquets.

Donc, en générale les pertes de paquets sont dues principalement à la saturation des files d'attentes au niveau des nœuds et aux interférences. En effet, beaucoup de nœuds du réseau tentent d'accéder au canal en même temps, ce qui génère des interférences qui vont affecter la qualité des communications. Les paquets sont supprimés au niveau de la couche MAC après sept tentatives de retransmissions. Parmi ces paquets, il peut y avoir des paquets de contrôle. La perte de ces derniers peut rendre certaines routes obsolètes, ou même créer des boucles, et dans ce cas-là, les paquets de données peuvent emprunter les mauvaises routes et finir par être perdus.

Avec nos approches, on arrive à éliminer l'impact des interférences, en augmentant la qualité du lien (via deux contraintes : énergie résiduelle et nombre de saut), tout en changeant la fréquence de communication. En effet, selon la figure 4.13, et dans un réseau à fort trafic, grâce à nos protocoles, on arrive à absorber plus de 30% des pertes, qui passent de 400 paquets dans le cas d'Original AODV-MIMC, EAODV-Pri et EAODV à 280 paquets avec EMCMI-AODV Bellman-Ford.

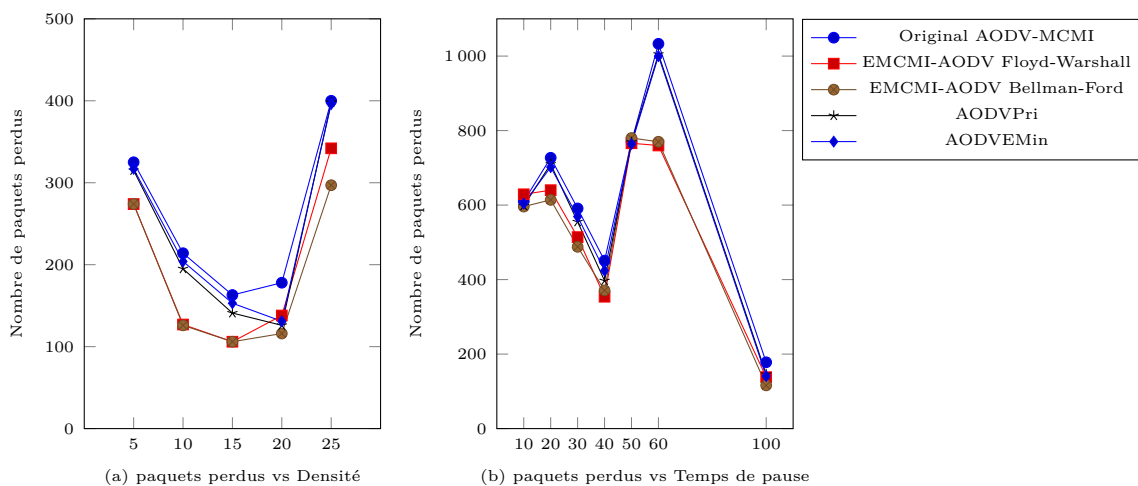


FIGURE 4.13: Nombre de paquets perdus vs densité et la mobilité des nœuds

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

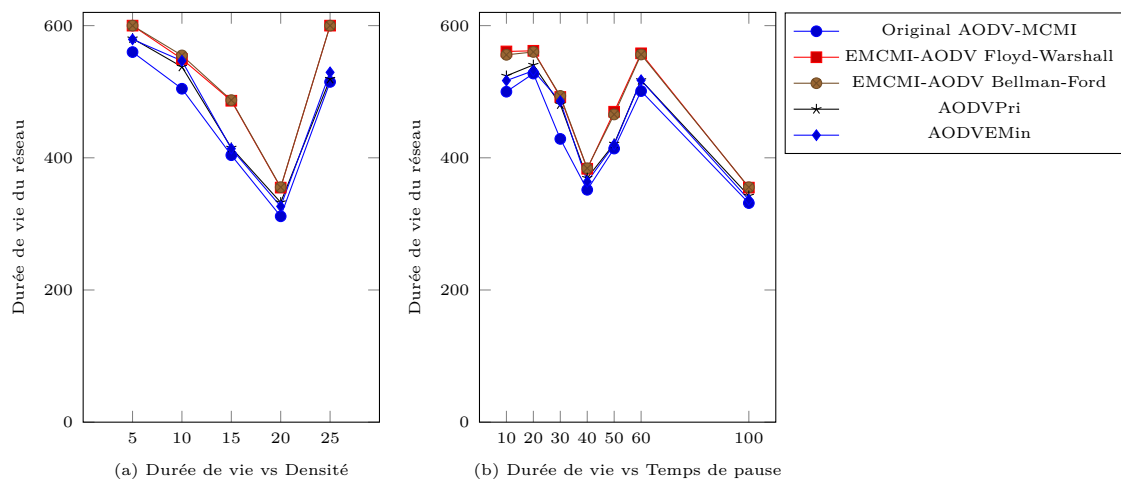


FIGURE 4.14: Durée de vie du réseau vs densité et la mobilité des nœuds

4.5.2.3 Durée de vie du réseau

La durée de vie du réseau en fonction du nombre de nœuds et le temps de pause est illustrée sur la figure 4.14. D'après les résultats de simulations, nous pouvons remarquer qu'il y a une amélioration de performance au niveau de la durée de vie du réseau lors de l'utilisation de nos approches. Cette amélioration est à l'origine principalement de la réduction des messages de contrôle et l'emploi d'un seuil d'énergie par nos solutions : les messages de contrôle ne sont plus diffusés dans tout le réseau, seuls les nœuds ayant une énergie supérieure au seuil fixée échangent ces messages. Les protocoles de routages EMCMI-AODV Floyd-Warshall et EMCMI-AODV Bellman-Ford favorisent donc les nœuds disposant d'une plus grande énergie résiduelle et procède à l'utilisation équitable de cette dernière. A partir de ce graphe, on constate que nos approches à réaliser une augmentation de 22% de la durée de vie du réseau.

4.6 Conclusion

Parmi les problématiques les plus importantes pour les réseaux mobiles Ad Hoc sont l'accès au médium radio sans fil et l'économie de l'énergie. Dans ce chapitre, nous avons proposé deux protocoles de routage EMCMI-AODV Floyd-Warshall et EMCMI-AODV Bellman-Ford. Ces protocoles partagent le même principe de routage d'AODV en plus ils utilisent une politique de l'équilibrage de la charge au niveau réseau et un mécanisme de gestion des interfaces fixes et commutables au niveau MAC pour assurer une durée de vie maximale au réseau, gérer d'une manière efficace l'ensemble des interfaces présentes dans le réseau et permet aux différents nœuds de transmettre simultanément dans une même zone de portée sur des canaux distincts.

Nous avons montré par simulations que nos protocoles tenaient une meilleure stabilité et donnaient des résultats intéressants. De plus, nos solutions sont indépendantes du nombre de nœuds dans le réseau ce qui lui donne un caractère évolutif. Nous avons pu constater aussi

Chapitre 4. protocole de routage multi-interfaces multi-canaux à basse consommation d'énergie

que nos approches fonctionnent bien dans les réseaux à forte mobilité puisqu'ils sont basés sur une métrique qui prend en considération la qualité des liens et la disponibilité des canaux, c'est à dire les besoins en qualité de services.

Optimisation multi-objectif des services dans les réseaux mobiles Ad Hoc

Sommaire

5.1	Introduction	85
5.2	Modélisation du système	86
5.2.1	Méthodes au fonctionnement couplé	86
5.2.2	Gestion de l'interface fixe et commutable	87
5.2.3	Gestion de la file d'attente	88
5.2.4	Estimation de l'énergie résiduelle	89
5.2.5	Estimation de délai disponible	89
5.3	Description du GA-AODV et DP-AODV	90
5.3.1	Découverte des chemins	90
5.3.2	Sélection des chemins	90
5.4	Modèles de simulation & métriques de performance	96
5.4.1	Modèle de trafic	96
5.4.2	Modèle de mobilité	97
5.4.3	Modèle de propagation	97
5.4.4	Métriques de performance	97
5.5	Résultats et discussion	97
5.5.1	Taux de livraison normalisé (Packet delivery ratio)	97
5.5.2	Délai de bout en bout	99
5.5.3	Charge normalisée du routage	100
5.5.4	Durée de vie du réseau	101
5.6	Conclusion	102

5.1 Introduction

Les protocoles de routage Ad Hoc effectuent un routage au-mieux où un seul critère d'optimisation est considéré dans le choix des chemins, typiquement, le nombre de sauts ou

le débit. Cependant, le transfert des données volumineuses requière la garantie d'une certaine qualité de service qui s'exprime en termes de plusieurs métriques telles que le débit, délai et le taux de perte.

Afin de traiter le problème de plus court chemin multi-objectif, nous avons proposé une extension de l'algorithme AODV mono-objectif. Cette extension repose sur l'utilisation de quatre métriques :

- Le délai de bout-en-bout : Il s'agit de temps écoulé entre l'envoi d'un paquet par un émetteur et sa réception par le destinataire.
- La taille de la file d'attente : Pour minimiser la latence des transmissions, les chemins composés des nœuds les moins chargés doivent être sélectionnés.
- Énergie résiduelle des batteries : Calculées pour éviter les nœuds de faibles énergies afin de prolonger la durée de vie du réseau.
- Nombre de saut : Compteur qui sert à stocker le nombre de sauts que le paquet a effectué.

Notre travail s'effectue en deux phases : la phase de modélisation et la phase de résolution. En phase de modélisation, les variables de décision, la fonction objective et les contraintes à respecter sont définies. La phase de résolution consiste à appliquer un algorithme génétique ou dynamique sur le problème modélisé.

5.2 Modélisation du système

5.2.1 Méthodes au fonctionnement couplé

Pour accroître les performances de la pile protocolaire dans son ensemble et optimiser la performance de transmission de bout en bout entre une source et une destination, au lieu d'effectuer des optimisations locale à chaque couche, on va créer dans notre contribution une architecture plus sensible aux spécificités des réseaux mobiles Ad Hoc où l'ensemble des informations des différentes couches protocolaire sera pris en compte en même temps [54, 55].

Notre système permet la consultation en temps réel d'informations disponibles dans les trois couches inférieures du modèle de référence (figure 5.1). Les informations disponibles dans une couche et pouvant être utilisées par d'autres sont :

- Couche Physique : relative à la technologie ou l'architecture utilisée, elle donne des informations sur l'état du canal de transmission, le niveau de batterie du nœud et peut modifier la puissance de transmission pour accroître la portée des transmissions.
- Couche MAC : le nombre de retransmission effectuée au niveau de cette couche peut donner une idée sur la qualité du lien sans fil, énergie nécessaire pour la transmission des données et l'état de queue.

- Couche réseau : faire intervenir des informations sur la qualité de transmission dans l'algorithme de routage, Consommation d'énergie, voisinage des nœuds et le nombre de sauts qui sépare deux nœuds par une route donnée.

5.2.2 Gestion de l'interface fixe et commutable

Dans cette contribution le système multi-interfaces multicanaux est utilisé. La figure 5.2 illustre un exemple de gestion de l'interface fixe et commutable. Le nœud N1 souhaite transmettre des paquets à N3 en passant par N2. Les interfaces fixes des nœuds N1, N2 et N3 sont assignées aux canaux 2,3 et 1 respectivement. Supposons que l'interface commutable du nœud N1 est assignée au canal 1, celle de N2 à 2 et celle de N3 à 3. Premièrement, l'interface commutable de N1 permute vers le canal 3 car le canal fixe du nœud N2 est 3. Ensuite, N2 commute vers 1 pour que les paquets soient reçus par N3 sur son canal fixe 1. Une fois les interfaces commutables sont bien affectées, il n'est plus nécessaire de changer de canal pendant la transmission de ce flux.

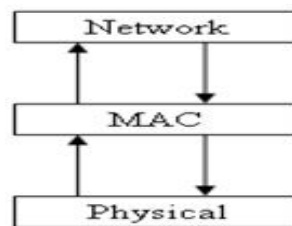


FIGURE 5.1: Communication de type cross layer à travers la pile protocolaire.

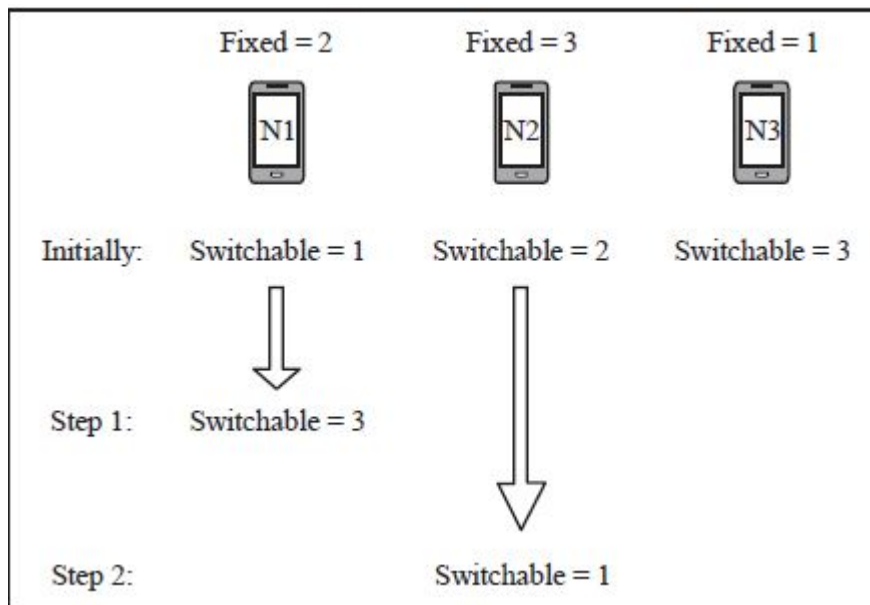


FIGURE 5.2: Gestion des interfaces commutables pour un réseau de 3 noeuds et 3 canaux.

5.2.3 Gestion de la file d'attente

Dans la modélisation proposée chaque canal est associé à une file d'attente comme le montre la figure 5.3. Pour envoyer un paquet, le canal fixe de la destination est cherché dans la table de voisinage puis le paquet est inséré dans la file d'attente correspondante. Si l'émetteur a le même canal fixe que le récepteur, le paquet est mis dans la file d'attente du canal fixe. Sinon celui-ci est mis dans la file d'attente d'un canal appartenant à l'interface commutable. La figure 5.4 représente la gestion des files d'attente d'un nœud. Cette représentation montre la classification des paquets dans les files d'attente selon leur prochain saut et son canal utilisé. L'efficacité de la file d'attente est déterminée par un certain nombre de mesures de performance qui sont calculées ou évaluées à partir de ses caractéristiques physiques.

De manière générale on peut identifier trois mesures de performance intéressantes à calculer lors de l'analyse de n'importe quelle file d'attente :

- Une mesure du temps d'attente subit par un paquet quelconque.
- Une indication sur la manière dont les paquets peuvent s'accumuler dans le système.

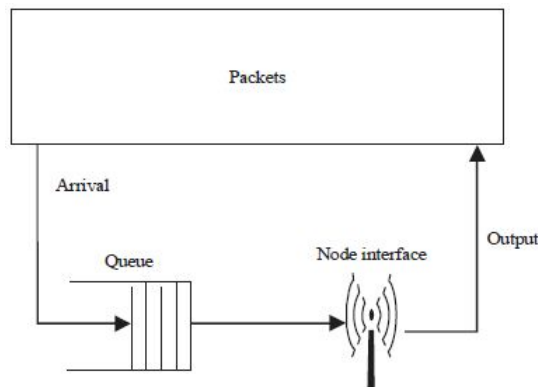


FIGURE 5.3: Files d'attente maintenues pour un canal.

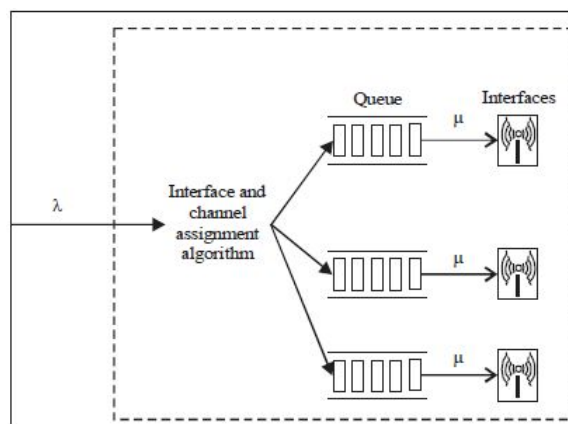


FIGURE 5.4: Gestion des files d'attente d'un nœud.

- Une mesure de la durée de repos ou d'activité des nœuds.

Dans notre contribution nous nous intéressons à l'analyse d'un système de file d'attente M/M/C dont le but est de déterminer les paramètres permettant de décrire les performances de tels systèmes. Dans le système M/M/C les paquets arrivent selon un processus de Poisson d'intensité λ , c'est-à-dire que le nombre de paquets qui arrivent dans un intervalle de temps t , $N(t)$, est une variable aléatoire de type Poisson de paramètre λt .

Chaque interface a un taux de service μ , mais le taux de service dépend du nombre de paquets dans le système, si le nombre est inférieur à S seul un sous-ensemble d'interface égal au nombre de paquets sont actifs, si le nombre de paquets est supérieur ou égal à S , les S interfaces sont actifs.

Le taux d'utilisation de notre système est donné par :

$$p = \frac{\lambda}{s\mu} \quad (5.1)$$

5.2.4 Estimation de l'énergie résiduelle

La métrique la plus utilisée dans les protocoles de routage Ad Hoc est le nombre minimum de sauts. Cependant, cette métrique a un effet négatif sur la consommation de l'énergie des nœuds du réseau. Par exemple, dans la figure 5.6, le trafic allant du nœud 1 au nœud 6 va toujours passer par le nœud 4, ce qui va provoquer la consommation totale de l'énergie de ce dernier. L'utilisation d'un algorithme de routage qui tient compte de la métrique énergétique va, au contraire, utiliser une autre route que celle passant par le nœud 4 (par exemple, la route passant par les nœuds 2,5 et 7).

Notre approche donc consiste à utiliser les nœuds les mieux chargées, de façon à améliorer la durée de vie du réseau. Pour ce faire, nous assignons à chacun des liens un poids qui est fonction de l'énergie résiduelle de chaque couple de nœuds voisins et le nombre de sauts vers la source. Le routage se fait donc préférentiellement suivant les routes de plus faible poids, en agrégeant l'ensemble des liens composant la route. L'équation 4.1 (on la note fl) permet d'attribuer le poids à chacun des liens.

Note : Quand l'énergie résiduelle d'un nœud est plus important que la valeur d'un seuil fixée, le nœud avance le message RREQ, mais dans le cas contraire, il écarte le message et refuse de participer à relayer des paquets

5.2.5 Estimation de délai disponible

Le délai de transmission dans un réseau peut être décomposé en délai dû à l'application (audio ou vidéo par exemple), délai dans la file d'attente et délai d'accès au médium. De nombreuses solutions ont été apportées pour estimer ce paramètre. Des modèles l'estiment

par le délai de bout en bout tels que [56] ou [57]. D'autres modèles [58], [59], [60], [61] et [62] l'estiment par le délai des liens. Dans notre contribution l'estimation de ce paramètre est basée sur le temps moyen qu'un paquet passe dans la file d'attente et le nombre de saut entre le nœud source et destination. :

$$f_2 = T_q E(H) \tag{5.2}$$

Où :

T_q : Temps moyen qu'un paquet passe dans la file.

$E(H)$: Nombre de saut entre le nœud source et le nœud de destination.

5.3 Description du GA-AODV et DP-AODV

Dans cette section, nous proposons de formuler le problème de routage à l'aide de l'algorithme génétique et dynamique.

5.3.1 Découverte des chemins

Quand un nœud intermédiaire reçoit un paquet RREQ, il insère les valeurs suivantes dans l'entête du RREQ :

- Son énergie résiduelle dans le champ "RE".
- Sa valeur d'estimation du délai dans le champ "Q".
- Son adresse dans le champ "LB".

5.3.2 Sélection des chemins

Après la réception du premier paquet RREQ, chaque nœud attend pour un certain temps (Wait_Time) avant de commencer la sélection des chemins par l'exécution de l'algorithme génétique ou dynamique suivant le protocole utilisé (GA-AODV ou DP-AODV).

5.3.2.1 Algorithme génétique

Contrairement aux méthodes exactes, les méthodes heuristiques dont l'algorithme génétique ne trouvent pas forcément la meilleure solution mais se contentent de trouver une solution réalisable au problème dans un temps raisonnable. Le principe de la méthode est inspiré des aspects biologiques. En effet, L'algorithme génétique est basé sur des mécanismes dérivés de l'évolution naturelle, où seuls les meilleurs individus subsistent et reproduisent de

nouveaux individus pendant une certaine succession d'itérations appelées générations, jusqu'à ce qu'un critère d'arrêt soit vérifié.

Au départ, on commence par générer une population initiale (routes) d'individus sous forme de chromosomes. Ensuite, plusieurs solutions appelées enfants plus ou moins bonnes sont ainsi créées par croisements des individus-parents, puis sont soumises à un procédé d'évaluation de pertinence afin de trouver la solution la plus adaptée au problème. Des mutations peuvent intervenir lors de la création d'une progéniture, ceci permet une diversification évitant une convergence prématurée (figure 5.5).

Le protocole GA-AODV est donc caractérisé par différents points :

- Instance du problème.
- Espace de recherche.
- Codage des points de l'espace de recherche (par un chromosome).
- Opérateurs génétiques classiques.
 - Sélection.
 - Hybridation.
 - Mutation.
 - Diversification.

Nous allons préciser pour chacun de ces points les options choisies pour l'implantation de notre extension de routage Ad Hoc.

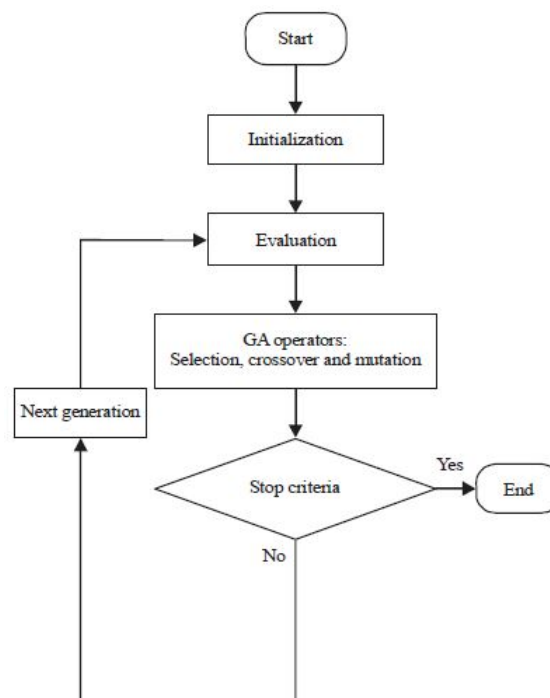


FIGURE 5.5: Architecture générale d'un algorithme génétique

5.3.2.1.1 Instance du problème

Une instance de routage est un graphe complet de n sommets dont les arêtes sont pondérées par un coût strictement positif. L'instance sera alors implantée comme une matrice $M(n \times n)$ dont les coefficients sont strictement positifs sauf sur la première diagonale où ils sont tous nuls. M est appelé matrice de coût. Ainsi la distance entre le sommet j et le sommet i est M_{ij} et $M_{ij} = M_{ji}$.

5.3.2.1.2 Espace de recherche

C'est l'ensemble S_n des permutations de $\{1, 2, \dots, n\}$. Un point de l'espace de recherche est une permutation.

5.3.2.1.3 Codage des points de l'espace de recherche

Il peut être difficile d'utiliser le codage binaire dans le problème de routage. Puisque la représentation binaire traditionnelle crée des difficultés pour les problèmes d'optimisation de grandes dimensions à haute précision. Dans notre contribution on a conservé les variables du problème dans le codage de l'élément de population, pour ne pas passer par le codage binaire intermédiaire. La structure du problème de routage est alors plus facilement conservée dans le codage.

Donc le codage des individus de notre population se base sur le codage réel. Par la suite, une route est une succession des nœuds, comme le montre le tableau 5.1.

On suppose ici que 5 est le nœud source, 9 celui de destination et (13, 7, 2) sont les nœuds intermédiaires par cette solution.

Il est noté que, puisque le réseau étudié est dynamique, le codage qui le représente l'est aussi. En effet, une solution peut avoir un nombre de nœuds variable, c'est pour cela la taille d'un individu n'est pas fixe.

5.3.2.1.4 Fonction d'évaluation

L'étape la plus sensible dans la conception d'un système par évolution génétique est la définition de la fonction de fitness qui dépend du problème à résoudre et de l'espace de recherche qui en découle. Dans le cadre de notre adaptation, le chromosome (route) est évalué

5	13	7	2	9
---	----	---	---	---

TABLE 5.1: Exemple d'une solution par l'algorithme génétique

suivant les critères de coût par la formule suivante :

$$F = f_1 + f_2 \quad (5.3)$$

5.3.2.1.5 Initialisation de la population

L'algorithme génétique démarre en général avec une population initiale aléatoire de N individus (chaînes, chromosomes) dans le codage retenu. Mais dans notre implémentation des informations a priori sur le problème sont disponibles via la diffusion des requêtes RREQ (figure 5.6), il paraît bien évidemment naturel de générer les individus dans un sous domaine qui respecte les contraintes de routage, afin d'accélérer la convergence.

5.3.2.1.6 Opérateurs génétiques classiques

Trois opérateurs jouent un rôle très important dans la réussite et la convergence d'un algorithme génétique : sélection, croisement et mutation.

A) Sélection

La sélection permet d'identifier les meilleurs individus d'une population qui seront gardés pour former la nouvelle génération et éliminer progressivement les mauvais individus afin d'éviter la dégénérescence. Plusieurs techniques de sélection sont utilisées dans la modélisation des algorithmes génétiques qui peuvent être déterministes ou stochastiques [63]. La méthode la plus connue et la plus utilisée est la sélection par roulette pondérée de Goldberg [64]. Avec cette méthode chaque individu (chromosome) a une chance d'être sélectionné proportionnelle à sa performance, donc les individus ayant la plus grande valeur de fitness, auront plus de chance d'être sélectionnés pour former une nouvelle génération. Dans une population de N

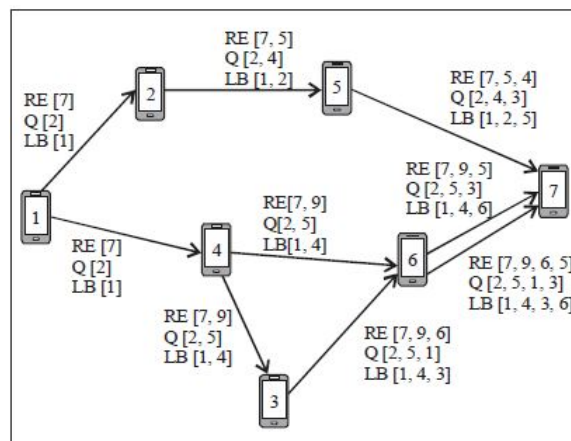


FIGURE 5.6: Exemple de propagation des requêtes RREQ

individus, la fonction de sélection est la suivante :

$$P_s(x_i) = \frac{F(x_i)}{\sum_{j=1}^N F(x_j)} \quad (5.4)$$

Nous appliquons cette méthode dans notre algorithme (figure 5.7).

B) Croisement

En général, le croisement consiste à appliquer un processus avec une certaine probabilité P_c aux individus précédemment sélectionnés. La probabilité de croisement définit la proportion de parents (dans la population courante) qui vont subir une opération de croisement. Après la comparaison de différentes méthodes de croisement, nous avons opté pour celle du croisement en un point pour générer de nouveaux individus (enfants). Le chromosome (parent) est divisé en deux parties, qui seront ensuite recombinaisonnées pour avoir de nouveaux enfants (figure 5.8).

Dans l'exemple de la figure 5.8 l'opérateur de croisement prend en entrée un couple d'in-

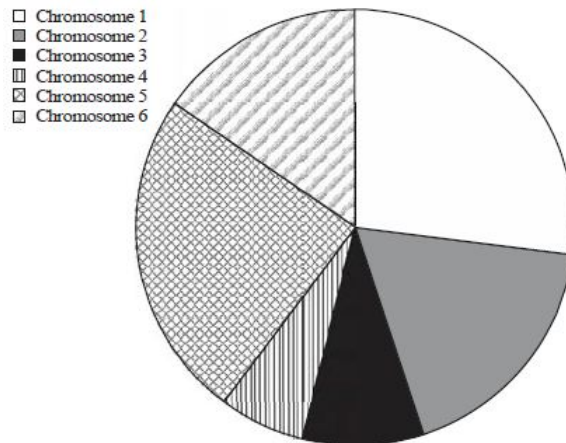


FIGURE 5.7: Exemple de sélection par roulette

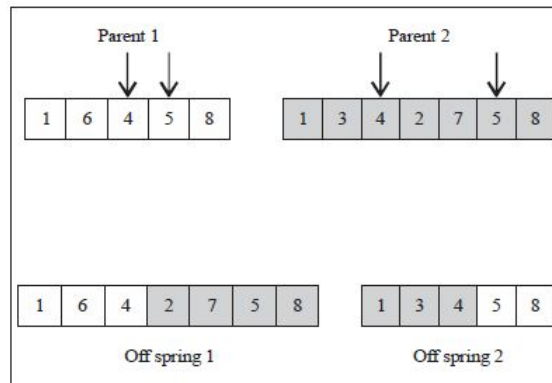


FIGURE 5.8: Exemple de croisement

dividus parents (Parent1 et Parent2) et renvoie un couple d'individus enfants (offspring1 et offspring2) obtenus en choisissant aléatoirement un point de croisement dans les chromosomes et en recopiant dans le fils offspring1 les gènes de parent1 jusqu'au point de croisement puis en complétant avec les gènes de Parent2. On effectue l'opération symétrique pour offspring2.

C) Mutation

Une fois la phase de croisement effectuée, l'opérateur de mutation est appliqué avec une petite probabilité P_m qui est appelée taux de mutation. Cet opérateur joue le rôle d'un (élément perturbateur). Cela nous permet de maintenir une certaine diversité dans la population et d'explorer plus d'espace de recherche. Dans notre algorithme un gène (nœud) sera sélectionné aléatoirement et sa valeur sera changée selon la base topologique (figure 5.9).

5.3.2.2 Algorithme dynamique

Pour appliquer l'algorithme dynamique de routage, nous modélisons notre réseau mobile Ad Hoc par un graphe $G = (V, E)$ où V est l'ensemble des nœuds et $E \subset V \times V$ l'ensemble des liens entre ces derniers. Nous supposons d'une part que le graphe est sans boucle. D'autre part, une paire quelconque de nœuds ne peut être connectée que par au plus un seul lien. Ceci justifie la notation $e = (V1, V2)$ pour parler de l'unique liaison de $V1$ à $V2$. On prendra bien soin de noter que $(V2, V1)$ est une liaison similaire de e .

Etant donnée une paire de nœuds distincts (u, v) nous appelons chemin ou route de u à v une séquence de nœuds $(V1, V2, \dots, Vn)$ telle que $(Vq, Vq + 1) \in E$, $V1 = u$ et $Vn = v$. Les relations entre les éléments peuvent être pondérées, ou valuées, c'est-à-dire que chaque arc du graphe est muni d'un poids, ou coût, défini par :

$$f(i, j) = \frac{H}{\text{mean}(EN_i, EN_j)} + T_q E(H) \quad (5.5)$$

S'il existe un chemin menant d'un nœud u à un nœud v alors son coût égal la somme des

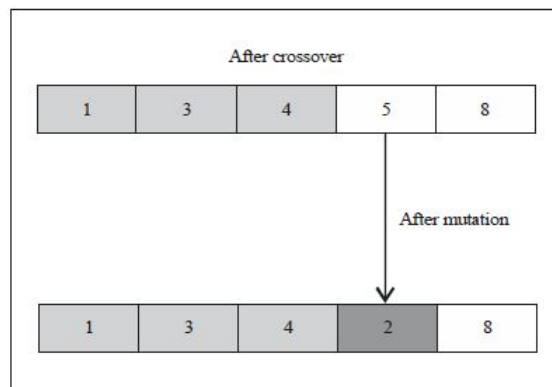


FIGURE 5.9: Exemple de mutation

valeurs des arcs ou arêtes traversées pour joindre les deux nœuds.

$$C(p) = \sum_{i=1}^k f(v_{i-1}, v_i) \quad (5.6)$$

Donc, le coût du chemin le plus court de u à v est :

$$\delta(u, v) = \begin{cases} \min \{C(p)\} & \text{s'il existe un chemin } p \text{ de } u \text{ à } v \\ \infty & \text{sinon} \end{cases} \quad (5.7)$$

Pour calculer les chemins de coût minimal ou plus courts chemins d'un nœud de départ u vers tous les autres nœuds, on utilise l'algorithme Bellman Ford dont le pseudo code est donné par l'algorithme 3.

Algorithm 3 Bellman-Ford(G,s)

```
for all edges  $\in G(V)$ 
 $D(V) \leftarrow \infty$ 
 $Parent[V] \leftarrow -1$ 
 $D(s) \leftarrow 0$ 
For  $i \leftarrow 1$  to  $|G(V)| - 1$ 
for each edge  $(u, v) \in G(E)$ 
If edge can be relaxed
 $D(v) \leftarrow D(u) + \text{weight of edge } (u, v)$ 
 $Parent[v] \leftarrow u$ 
for each edge  $\in G(E)$ 
If edge can be relaxed
Return false
Return true
```

5.4 Modèles de simulation & métriques de performance

5.4.1 Modèle de trafic

Pour analyser le comportement de nos protocoles de routage, nous avons choisi des sources de trafics à débit constant CBR (Constant Bit Rate) associées au protocole UDP, avec variation du nombre de sources CBR et le nombre de nœuds. Le taux d'émission des paquets est fixé à 8 paquets par seconde.

5.4.2 Modèle de mobilité

Afin de décrire l'impact de la densité sur les performances de nos protocoles de routage, nous avons fait varier le nombre de nœuds entre 5 et 25 nœuds. Et pour décrire l'impact de la mobilité nous avons fait varier le temps de pause des nœuds mobiles entre 3 et 25 secondes.

les nœuds mobiles du réseau utilisent le modèle de mobilité Random Waypoint et se déplacent dans une surface de dimension 1000m x 1000m.

5.4.3 Modèle de propagation

Le modèle de propagation utilisé est le Two-ray-ground puisqu'il donne des résultats plus justes que le modèle de propagation en espace libre, plus particulièrement quand la distance est assez grande. De plus, un module d'antenne omnidirectionnel (Omni Directionan Antenna Module) est utilisé dans la couche physique.

La file d'attente du noeuds de réseau peut contenir jusqu'à 50 paquets et nous avons considéré le paramètre DropTail /PriQueue pour donner la priorité aux paquets de contrôle.

5.4.4 Métriques de performance

Pour évaluer nos algorithmes de routage pour les réseaux mobiles Ad Hoc, nous avons choisi les paramètres de performances suivants :

- Le délai moyen de bout en bout (End to End Delay).
- Le taux de livraison normalisé (Packet delivery ratio).
- La charge normalisée du routage (Normalizing routing overhead).
- Durée de vie du réseau.

5.5 Résultats et discussion

5.5.1 Taux de livraison normalisé (Packet delivery ratio)

Dans la figure 5.10, nous étudions l'impact de la charge du réseau (nombre de nœud) sur le taux de réception de paquets. Pour cela, nous avons varié le nombre de nœud de 5 à 25.

Nous remarquons que pour une faible densité (5 à 15 nœuds) les trois protocoles donnent des taux de réception de paquets inférieur par rapport au cas d'une densité plus élevée (20 à 25 nœuds). En effet, comme la connectivité du réseau est faible étant donné que la densité est faible, les protocoles ne trouvent pas des routes pour atteindre certaines destinations et surtout ceux qui sont éloignés.

Chapitre 5. Optimisation multi-objectif des services dans les réseaux mobiles Ad Hoc

En général, les protocoles DP-AODV et GA-AODV, possèdent un taux de livraison de paquets important. Ils performent mieux comparés au protocole AODV.

Dans la figure 5.11 nous avons calculé le taux de délivrance de paquets en fonction du temps de pause de nœuds. Ce dernier prend ces valeurs comme suit : 3,7,15,20 et 25s. Le résultat montre que plus le temps de pause augmente, ce qui signifie que les nœuds se déplacent peu, le taux de réception de paquets augmente pour les trois protocoles. nous remarquons aussi que nos protocoles sont plus efficace, ils peuvent augmenter l'efficacité jusqu'à 3% par rapport à AODV.

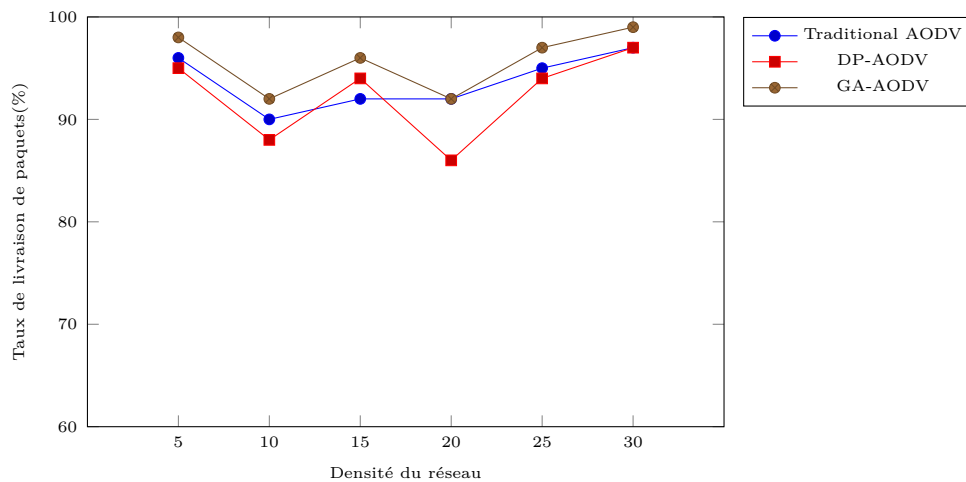


FIGURE 5.10: Taux de livraison de paquets vs densité

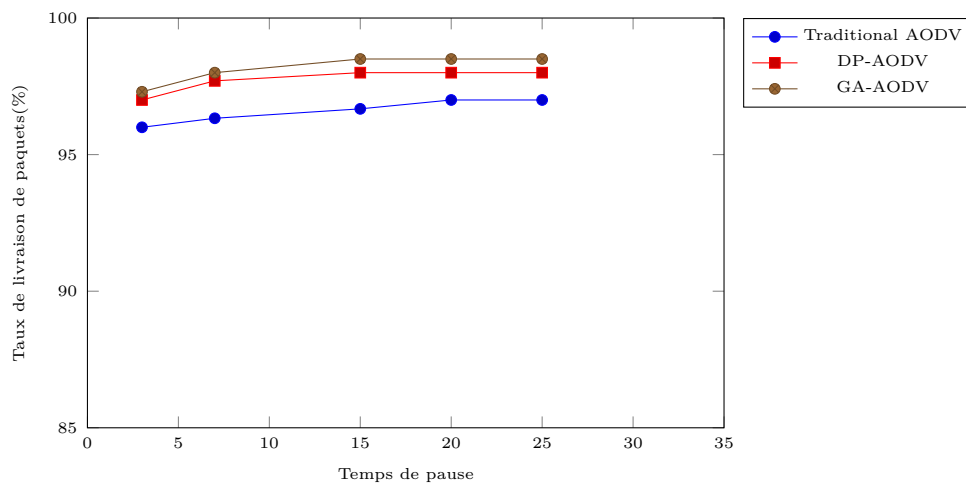


FIGURE 5.11: Taux de livraison de paquets vs mobilité

5.5.2 Délai de bout en bout

Le délai de bout en bout caractérise le fonctionnement d'un protocole de routage. Il représente le temps nécessaire pour acheminer un paquet de sa source jusqu'à atteindre sa destination. Dans nos simulations ce temps est calculé comme la moyenne des temps de réponse de chaque connexion de données CBR.

Les figures 5.12 et 5.13 montrent le délai de bout en bout en fonction du temps de pause et le nombre des nœuds dans le réseau.

Les deux figures montrent que les protocoles modifiés DP-AODV et GA-AODV assurent de bonnes performances en termes de délai de bout en bout par rapport à la version standard du protocole AODV, et ce pour tous les nombres des nœuds et les valeurs de temps de pause. Précisément, pour un environnement à forte mobilité, le délai enregistré par les protocoles

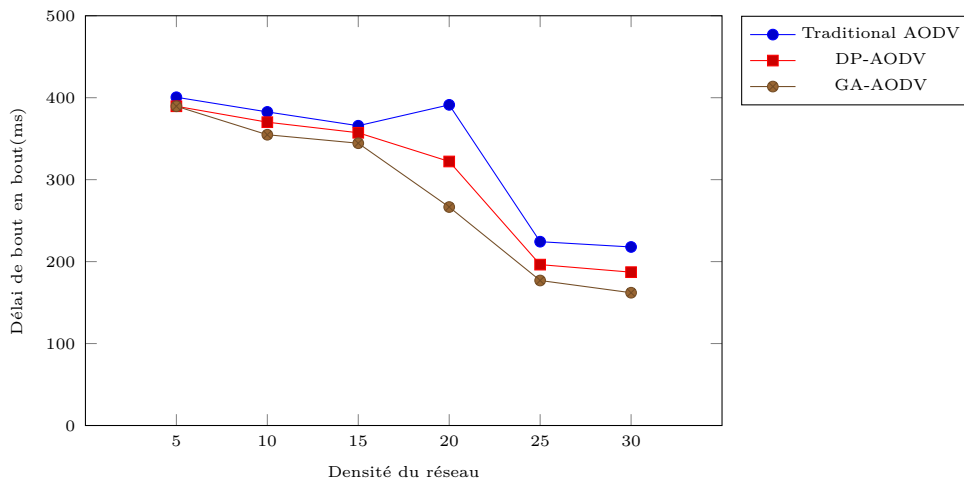


FIGURE 5.12: Délai de bout en bout vs densité

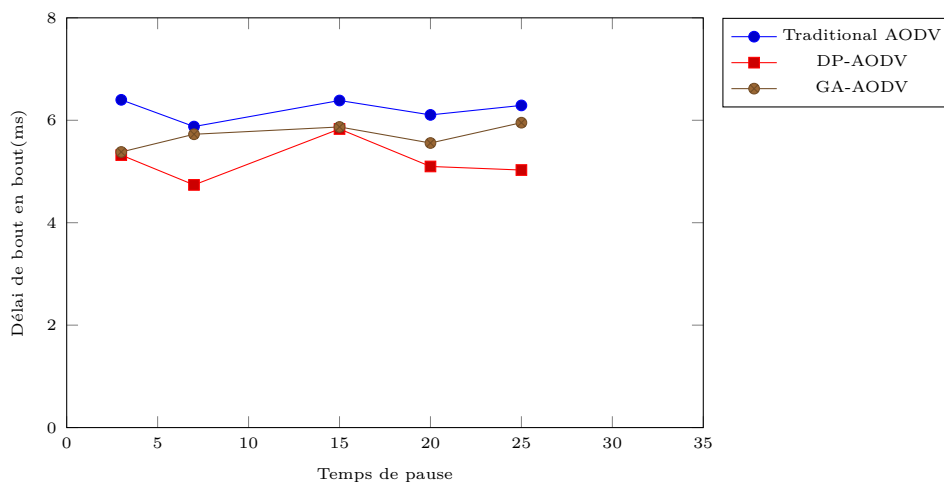


FIGURE 5.13: Délai de bout en bout vs mobilité

DP-AODV et GA-AODV est d'environ 5,2ms (amélioration de 0,9ms) en comparaison avec le protocole standard. Dans le cas statique, le délai diminue à 4,25 ms pour DP-AODV (amélioration de 0.8ms par rapport à AODV original). En comparaison avec AODV, il est intéressant de noter que nos protocoles proposés DP-AODV et GA-AODV offrent de bons résultats dans les topologies dynamiques, cela s'explique par le fait que nos protocoles considèrent, en plus de la qualité de la liaison l'énergie résiduelle de chaque nœud intermédiaire qui résulte des routes ayant une durée de vie plus importante, par conséquent on évite le temps de recherche des nouvelles routes.

5.5.3 Charge normalisée du routage

Les figures 5.14 et 5.15 illustrent la charge de routage normalisée (NRL : normalized routing load) introduite dans le réseau pour les trois versions du protocole AODV, où le nombre de paquets de routage est rationalisé par rapport aux paquets de données envoyés. Une surcharge assez stable des données de contrôle serait une propriété souhaitable dans les réseaux mobiles Ad Hoc.

On constate que nos protocoles produisent la plus faible quantité NRL par rapport au protocole AODV pour toutes les valeurs de temps de pause et nombre de nœuds. Ces résultats s'expliquent essentiellement par la stratégie de routage d'AODV qui s'appuie sur le nombre de saut entre la source et la destination, le protocole maximise la distance parcourue à chaque saut. Les nœuds se situent souvent à la limite de la portée radio ce qui entraîne un nombre important de ruptures des liens. A chaque rupture de nombreux messages de contrôle sont nécessaires pour rétablir le lien brisé.

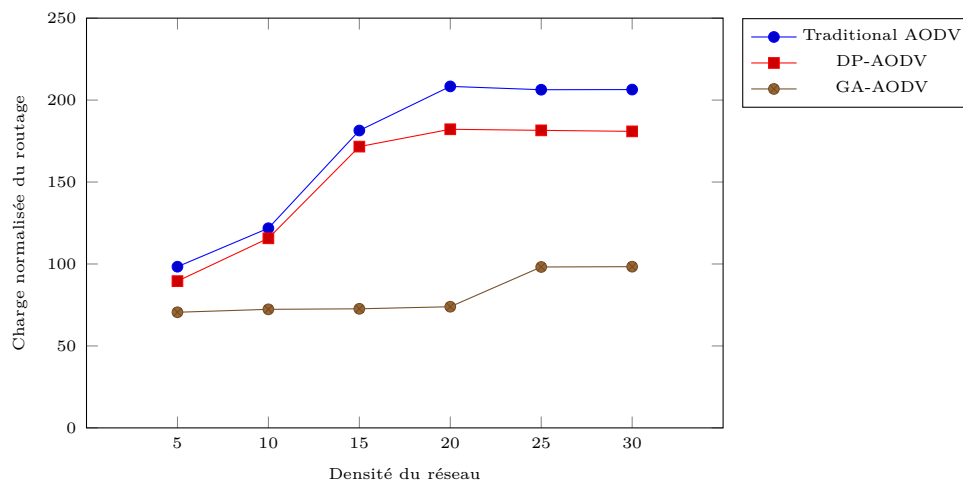


FIGURE 5.14: Charge normalisée du routage vs densité

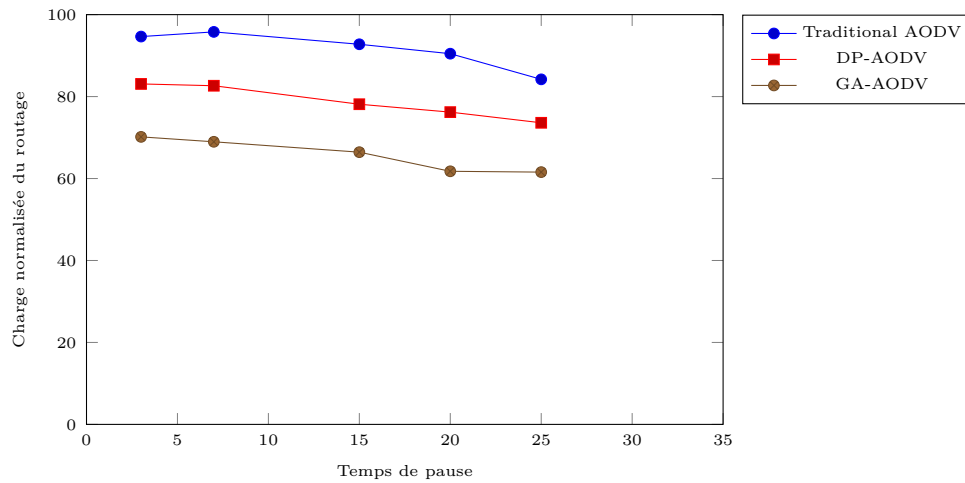


FIGURE 5.15: Charge normalisée du routage vs mobilité

5.5.4 Durée de vie du réseau

Dans la Figure 5.16, nous traçons la durée de vie du réseau en fonction du nombre de nœuds. Nous remarquons que les durées de vie obtenues par GA-AODV est bien supérieures à celui d'AODV et DP-AODV. Ceci est dû à la bonne gestion des routes par le protocole de routage GA-AODV qui vise à utiliser l'équilibrage de charge pour minimiser la charge d'utilisation des nœuds. Notons aussi que lorsqu'on augmente la densité du réseau les performances en terme de durée de vie du réseau démunie ce qui est tout à fait logique.

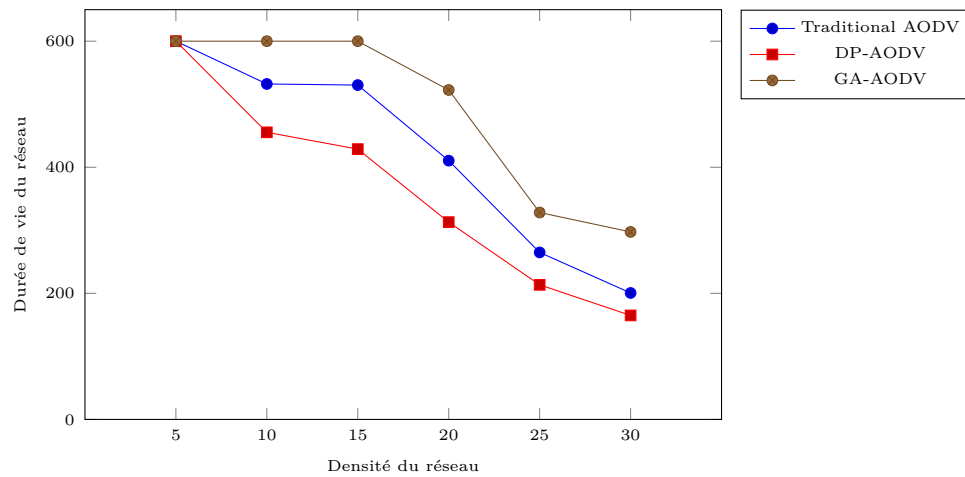


FIGURE 5.16: Durée de vie du réseau vs densité

D'après les résultats de la figure 5.17 nous constatons que plus les nœuds se déplacent avec une grande vitesse, plus l'énergie consommée par les nœuds devient importante pour tous les protocoles. Cependant, AODV consomme plus d'énergie comparativement à nos approches. Cela démontre qu'une prise en compte de l'énergie résiduelle des nœuds dans le processus de

roulage équilibre mieux la consommation énergétique du réseau.

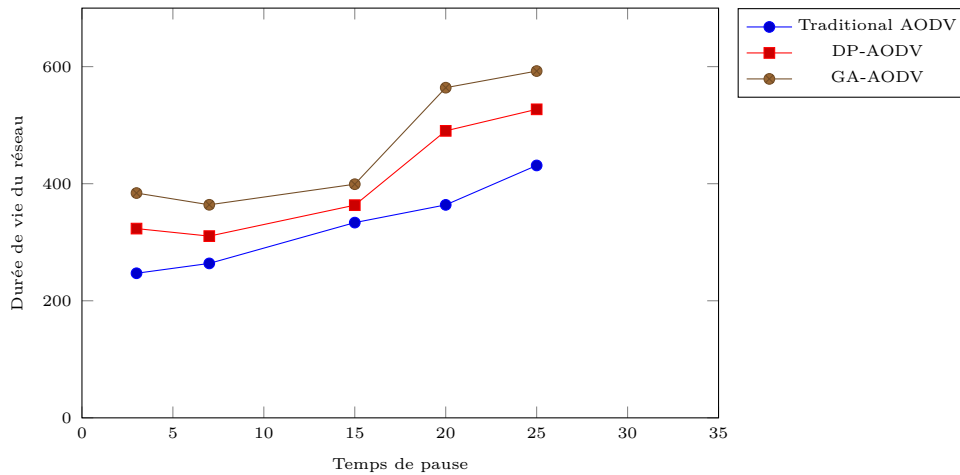


FIGURE 5.17: Durée de vie du réseau vs mobilité

5.6 Conclusion

Dans ce chapitre nous avons proposé et implémenté deux approches de routage Ad Hoc. L'objectif principal de ces approches est d'optimiser les performances du réseau en réduisant d'avantage le nombre de paquets de contrôles qui peut réduire les collisions, le délai de bout en bout, la consommation d'énergie et augmentant le taux de réception de paquets dans les réseaux mobiles Ad Hoc.

Les poids des liens nécessaires pour déterminer les meilleures routes sont calculés en fonction du métrique multicritère (nombre de saut, énergie résiduelle, délai et la file d'attente des nœuds). Ces métriques permettent aux nos protocoles d'améliorer la QoS par rapport au protocole AODV qui utilise des mécanismes classiques tels que le nombre de sauts comme métrique de routage.

Dans la première approche, la sélection des routes est basée sur l'algorithme de Bellman Ford appelée DP-AODV, La deuxième est basé sur l'algorithme génétique appelée GA-AODV.

Les résultats de simulation avec le simulateur réseau NS2 indiquent que nos variantes améliorées du protocole AODV, en particulier la variante GA-AODV permet d'établir le meilleur compromis de performance et donc de cerner la qualité des liens au plus juste.

Nous remarquons aussi que l'utilisation des critères de contrôle des opérateurs génétiques accélère la convergence du protocole GA-AODV par rapport à DP-AODV.

Conclusion et perspectives

Dans les dernières années, la recherche dans le domaine des réseaux Ad Hoc a connu une croissance remarquable. Les réseaux Ad Hoc offrent la commodité d'une mobilité sans contrainte et un déploiement rapide. Ils sont caractérisés en particulier par :

- Absence d'une infrastructure fixe.
- Une structure dynamique.
- Des ressources limitées.

Les ressources comprennent entre autres l'énergie résiduelle des nœuds, la puissance de transmission de données et la bande passante de communication. Cependant, ces propriétés rendent le problème de routage dans les réseaux ad hoc un axe de recherche capital.

Dans le but d'améliorer le routage dans les réseaux mobiles Ad Hoc, nous avons présenté quatre contributions dont l'objectif est d'améliorer la qualité de communications entre les nœuds du réseau en évitant le plus possible les collisions, en réduisant les interférences, augmentant la durée de vie du réseau et en limitant, de façon considérable, les paquets de contrôle induits par les procédures de découverte de routes utilisées par les protocoles de routage.

Dans notre première contribution, nous avons étudié l'impact respectivement des modèles de propagation radio et l'état du canal de transmission sur les performances de différents protocoles de routage. Pour ce faire, nous avons proposé de valider les résultats de simulations avec des modèles proche de la réalité en intégrant la probabilité d'erreur de transmission, caractérisé par les deux états de Markov. Comme prévu, le protocole AODV a montré des meilleures performances par rapport aux protocoles DSDV, DSR et OLSR.

Dans la deuxième contribution, nous avons proposé, une stratégie qui vise à étendre la durée de vie du réseau par une modification de la métrique de routage. La métrique de notre protocole de routage (E-AODV) est basée sur l'énergie résiduelle des nœuds et le nombre de saut possible entre l'émetteur et le récepteur, afin de permettre la distribution du trafic sur un maximum de nœuds du réseau, de sorte à avoir un niveau de consommation d'énergie relativement égale pour tous les nœuds et ainsi augmenter la durée de vie du réseau. Les résultats de simulation sur NS-2 ont montré que notre variante améliore le protocole AODV en diminuant le nombre des messages de contrôles et la consommation d'énergie, ainsi en augmentant le taux de paquets délivrés dans le réseau et sa durée de vie.

Dans la troisième contribution, nous avons proposé un nouvel algorithme de routage multi-interfaces multicanaux pour améliorer l'utilisation des canaux par rapport aux problèmes que nous avons identifiés. Nous avons remarqué que la plupart des méthodes d'accès multicanal qui ont été proposées traitent principalement le cas des réseaux mono-saut, c'est pourquoi nous avons proposé une méthode d'accès multi-interfaces multicanaux adaptée à une topologie multi-saut, qui passe à l'échelle, que l'on pourra également simuler afin de vérifier ses

performances. Dans le but de valider notre approche, nous avons comparé la performance de cinq protocoles de routage face à la mobilité et la densité, avec et sans considération de l'énergie résiduelle des nœuds. Les résultats de simulation ont démontré que notre extension performe mieux.

Dans la quatrième contribution nous avons présenté un mécanisme d'optimisation des réseaux ad hoc tout en garantissant une attribution efficace des canaux et assurant la connectivité des nœuds mobiles. Pour ce faire, nous avons opté pour deux méthodes, une méthode heuristique qui se contente de trouver une solution réalisable au problème de routage et constitue une alternative très intéressante pour résoudre les problèmes d'optimisation de grande taille et une méthode qui permet de résoudre ce problème de façon exacte via l'utilisation de l'algorithme dynamique de Bellman Ford. Dans nos approches, nous avons utilisé une métrique combinée (délai de bout-en-bout, taille de la file, énergie résiduelle des batteries et nombre de saut) pour choisir les meilleurs chemins au lieu d'un seul paramètre qui est le nombre de sauts. Les résultats de simulation ont montré que notre modèle est efficace en termes de consommation d'énergie, d'affectation de canaux et de garantir la connectivité globale de réseau.

En perspectives, nous proposons, d'améliorer encore plus la performance du protocole AODV en fusionnant d'autres métriques de routage dans une unique implémentation. Il sera pertinent d'étudier aussi l'effet de rapport signal à bruit SINR sur l'affectation de canaux et contrôle de puissance. Aussi, l'étude menée dans notre travail suppose, à cause de la limitation de l'outil de simulation, que quelques paramètres (portée radio, capacité des interfaces) sont identiques pour tous les nœuds mobiles du réseau, chose qui n'est pas vraie dans la réalité. Tenir en compte de ces paramètres s'avère complexe, mais elle apparaît beaucoup plus pertinente, si on arrive à modéliser ces paramètres dans le simulateur NS2. Donc nous envisageons de se concentrer sur le problème de routage pour améliorer de plus en plus les protocoles proposés et le comparer avec d'autres protocoles de routage existants dans la littérature.

Implémentation et Simulation

A.1 Introduction

NS2 est un simulateur à événements discrets orienté objet basé sur C++ avec une interface textuelle OTCL (Object Tool Command Language). Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation, par contre le langage OTcl fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement d'événements, la configuration du réseau, la collecte de statistiques, etc. . .

L'application NS2 se compose donc de deux éléments fonctionnels : un interpréteur (utilise le langage OTCL) et un moteur de simulation (utilise le langage C++).

L'interpréteur permet à l'utilisateur de créer le modèle de simulation, ce qui revient à assembler les différents composants nécessaires à l'étude. Les composants du modèle de simulation sont appelés objets ou encore instances de classe. Le moteur de simulation effectue les calculs applicables au modèle préalablement construit par l'utilisateur via l'interpréteur. Le modèle de simulation est construit donc à partir de deux arborescences de classes :

- Une arborescence interprétée définie en OTcl : utilisée par l'interpréteur et visible par l'utilisateur.
- Une arborescence compilée définie en C++ : utilisée par le moteur de simulation (le simulateur) et qu'est l'ombre de l'arborescence interprétée.

Ce présent chapitre est consacré aux détails de l'implémentation de nos contributions sous NS2. Pour ce faire, nous présentons d'abord un aperçu du logiciel. Ensuite nous décrivons les différents fichiers concernés par les modifications et enfin, les nouvelles méthodes ajoutées, les changements dans le format des paquets, et le mécanisme de fonctionnement de nos agents de routage.

A.2 Aperçu du logiciel

A.2.1 Principe de base de simulation

Pour simuler un nœud dans NS2, on commence par la création d'un nouvel objet via l'interpréteur OTCL. Le nouvel objet interprété est cloné automatiquement en un objet compilé correspondant dans le moteur de simulation (simulateur).

En réalité on peut avoir des classes qui ne sont que dans l'interpréteur, elles servent à faire par exemple des assemblages (ou agrégations de classes) pour faciliter la manipulation et on peut avoir aussi des classes qui sont purement dans le simulateur et qui ne sont pas visibles de l'utilisateur et servent au fonctionnement interne d'un composant. La figure A.1 montre la dualité des classes de NS2.

A.2.2 Arborescence des classes compilées dans NS2

La figure A.2 représente l'arborescence de classes compilées du simulateur.

A.2.3 Liaisons entre l'interpréteur et le simulateur

La liaison TCL est l'interface de communication entre l'implémentation des objets de simulation sur les hiérarchies compilée et interprétée. Elle permet la transmission des messages de la hiérarchie compilée à la hiérarchie interprétée et vice versa. Le couplage TCL se compose de six classes importantes :

- Tcl

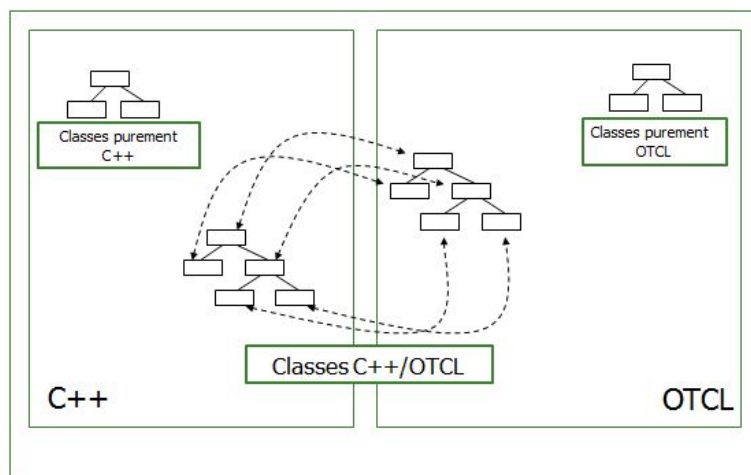


FIGURE A.1: Dualité des classes de NS2

- TclObject
- TclClass
- TclCommand
- EmbeddedTcl
- InstVar

La figure A.3 montre un exemple de communication entre l'implémentation des objets de simulation sur les deux hiérarchies (compilée et interprétée) :

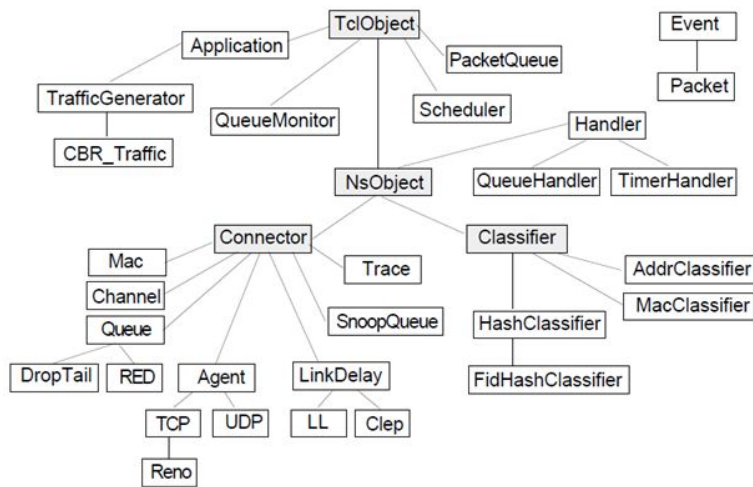


FIGURE A.2: Arborescence des classes C++ du moteur de simulation

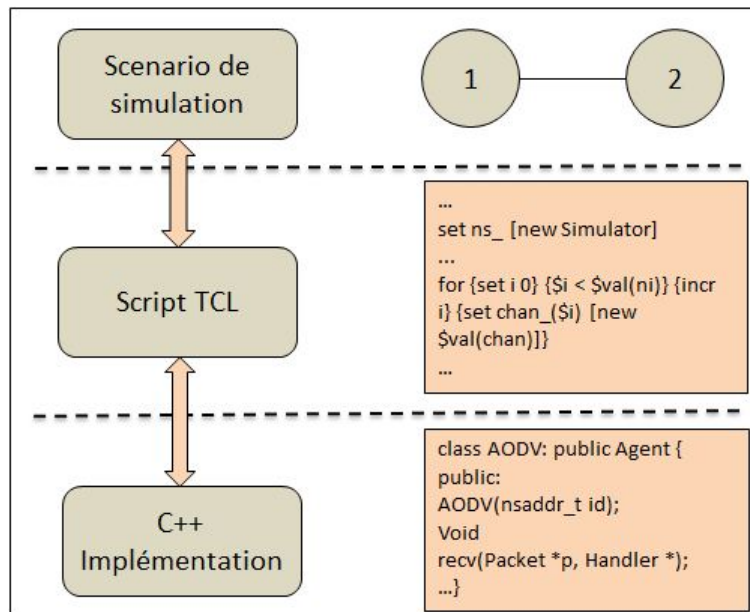


FIGURE A.3: Exemple de communication entre les deux hiérarchies NS2

A.2.4 Caractéristiques d'un nœud sous NS-2

Le nœud constitue l'élément de base de notre conception. Un nœud dans le simulateur NS2 est une classe définie dans OTCL, qui a une adresse et qui contient trois entités : le classifieur, le lien et l'agent (figure A.4).

A.2.4.1 Classifier

La fonction principale d'un nœud est d'examiner les champs du paquet reçu, notamment, l'adresse source et l'adresse destination. Selon ces valeurs, le nœud envoie ce paquet sur ses liens de sortie. Ceci est effectué par l'objet classifieur, qui a un rôle de démultiplexeur en quelque sorte.

Il existe plusieurs types de l'objet classifieur :

- Adresse classifieur : Il sélectionne les paquets adressés directement au nœud, et de choisir le lien vers le prochain nœud.
- Port classifieur : Son rôle est de sélectionner l'agent auquel le paquet est destiné.
- Multicast classifieur : Il est utilisé pour classifieur les paquets multicast.

A.2.4.2 Lien

Un lien dans le simulateur NS2 est un objet qui relie deux nœuds. Pour que les paquets puissent aller et venir dans les deux sens, il est nécessaire de créer un lien d'un nœud vers un autre et inversement. Généralement un lien a pour propriétés :

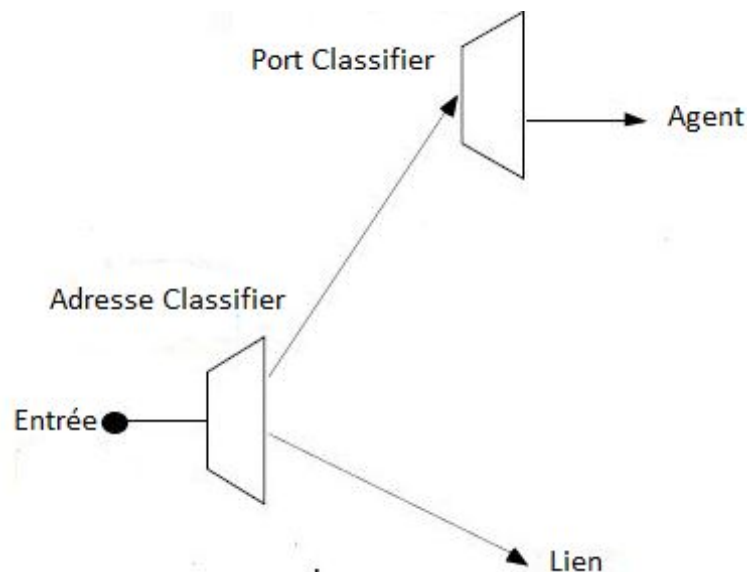


FIGURE A.4: Nœud NS2

La liste de fichiers
/ns2.35/tcl/lib/ns-lib.tcl
/ns2.35/tcl/ns-mobilenode.tcl
/ns-2.35/common/mobilenode.h
/ns-2.35/common/mobilenode.cc
/ns-2.35/mac/channel.cc
/ns-2.35/mac/mac.h
/ns-2.35/mac/mac-802_11.cc
/ns-2.35/aodv/aodv.h
/ns-2.35/aodv/aodv.c
/ns-2.35/aodv/aodv_rtable.h
/ns-2.35/aodv/aodv_paquet.h

TABLE A.1: Fichiers concernés par les modifications

- Un délai de propagation (latence).
- La bande passante.
- Un traitement de la file d'attente.

Exemple de création :

```
% lien entre n0 et n1
$ns duplex-link $n0 $n1 2Mb 20ms Droptail
```

A.2.4.3 Agent

Les agents représentent les points terminaux, là où des paquets de couche réseau sont consommés. Le rôle de l'agent est de fournir l'adresse de destination, les fonctions pour générer les paquets et l'interface à la classe application. Il existe plusieurs types d'agents dans NS2, chacun a un rôle spécifique :

- Agent TCP : pour émettre un trafic TCP.
- Agent UDP : pour émettre un trafic UDP.
- Agent TCPSink : pour la réception du trafic TCP.
- Agent NULL : pour la réception des paquets UDP.

A.3 Implémentation

Dans cette section nous présentons quelques fichiers concernés par les modifications (table A.1).

A.3.1 Implémentation du modèle Gilbert-Elliot

Nous avons implémenté le modèle de Gilbert-Elliot (chaines de Markov à deux états) dans NS2 où les transmissions (les paquets transmis) sont exposées à un taux d'erreur de façon uniformément distribuée et indépendante. Pour cela, on a ajouté une procédure dans l'interface applicative tcl afin de passer les paramètres suivants à NS2 :

- States : Tableau d'états.
- Periods : Tableau des durées d'états.
- Trans : Matrice de transition.
- Transunit : [pkt|byte|time].
- Sttype : Type de transition d'état utilisé : time ou pkt.
- Nstates : Nombre d'états.
- Start : Etat de départ.

```
proc MarkovErr {} {  
    set tmp0 [new ErrorModel/Uniform 0 pkt]  
    set tmp1 [new ErrorModel/Uniform 1 pkt]  
    set m_states [list $tmp0 $tmp1]  
    set m_periods [list 0.985 0.015]  
    set m_transmx { {0.994 0.0060 }  
                   {0.3820 0.618 } }  
    set m_trunit byte  
    set m_sttype time  
    set m_nstates 2  
    set m_nstart [lindex $m_states 0]  
    set em [new ErrorModel/MultiState $m_states $m_periods $m_transmx \  
    $m_trunit $m_sttype $m_nstates $m_nstart]  
    return $em  
}
```

Les valeurs utilisées dans la matrice de transition de notre stratégie sont des mesures réelles (expérimentations déjà citées dans «Janevski,T. : Book Traffic Analysis and Design of Wireless IP. Artech House, Boston »).

A.3.2 implémentation multi-interfaces multicanaux

Dans cette section nous détaillons les changements qui sont nécessaires dans les fichiers de NS2 afin d'étendre le simulateur à être capable d'ajouter plusieurs interfaces par noeud [64].

A.3.2.1 Changement au niveau du code OTCL

Les deux fichiers concernés par la modification du code OTCL sont :

A) Fichier ns-2.35/tcl/lib/ns-lib.tcl

Pour implémenter le modèle multi-interfaces multicanaux dans NS2, nous devons créer quatre nouvelles procédures dans ce fichier :

- Procédure set-interface{nbinterface}

Cette procédure nous permet de spécifier le nombre d'interface de chaque nœud mobile et on va l'appeler avant la création du nœud :

```
Simulator instproc set-interface {nbinterface} {  
  $self instvar numifs_  
  set numifs_ $nbinterface  
}
```

- Procédure get-interface

Cette procédure nous permet de retourner le nombre d'interface définie :

```
Simulator instproc get-interface { } {  
  $self instvar numifs_  
  if [info exists numifs_] {  
    return $numifs_  
  } else {  
    return ""  
  }  
}
```

- Procédure : set-canal {indice canal}

Faire appel à cette procédure permet d'ajouter une interface à un nœud donné :

```
Simulator instproc set-canal {indice canal} {  
  $self instvar chan  
  set chan($indice) $canal  
}
```

- Procédure nb-interface{val}

Cette procédure nous a aidés de changer la valeur locale de numifs_

```
Simulator instproc nb-interface {val} {  
  $self set numifs_ $val  
}
```

Ensuite, on va modifier deux procédures existantes dans le même fichier :

- Procédure node-config args{}

Dans cette procédure nous devons initialiser la variable `chan`, soit en tant que variable unique, si le modèle mono-interface mono-canal est utilisé, soit en tant que tableau, lorsque la multi-interface est activée. De plus, nous ajoutons la variable `numifs` à la liste des arguments passés à cette procédure. Les changements sont indiqués sur le code suivant :

```

Simulator instproc node-config args {
    # Object::init-vars{} is defined in ~tclcl/tcl-object.tcl.
    # It initializes all default variables in the following way:
    # 1. Look for pairs of {-cmd val} in args
    # 2. If "$self $cmd $val" is not valid then put it in a list of
    #     arguments to be returned to the caller.
    #
    # Since we do not handle undefined {-cmd val} pairs, we ignore
    # return value from init-vars{}.
    set args [eval $self init-vars $args]

    $self instvar addressType_ routingAgent_ propType_ macTrace_ \
    routerTrace_ agentTrace_ movementTrace_ channelType_ channel_ numifs_ \
    chan topoInstance_ propInstance_ mobileIP_ \
    rxPower_ txPower_ idlePower_ sleepPower_ sleepTime_ transitionPower_ \
    transitionTime_ satNodeType_ eotTrace_ phyTrace_

if {[info exists channelType_] && [info exists channel_]} {
    error "Can't specify both channel and channelType, error!"
} elseif {[info exists channelType_] && ![info exists satNodeType_]} {
    # Single channel, single interface
    warn "Please use -channel as shown in tcl/ex/wireless-mitf.tcl"
    if ![info exists chan] {
        set chan [new $channelType_]
    }
}
elseif {[info exists channel_]}{
# Si y a une interface, chan est un variable single. Si y a plus 2 interfaces,
chan est une table #(array).
if {[info exists numifs_]}{
set chan(0) $channel_
} else {
set chan $channel_
}
}
}

```

- Procédure `create-wireless-node{}`

Le code otcl suivant montre la deuxième procédure qu'on doit modifier (`create-wireless-node`). Dans notre modèle la procédure `add-interface`, définie dans le fichier `ns-mobilenode.tcl`, doit être appelée autant de fois que le nombre d'interfaces du nœud.

```

Simulator instproc create-wireless-node args {
    $self instvar routingAgent_ wiredRouting_ propInstance_ llType_ \

```

```

macType_ ifqType_ ifqlen_ phyType_ chan antType_ \
energyModel_ initialEnergy_ txPower_ rxPower_ \
idlePower_ sleepPower_ sleepTime_ transitionPower_ transitionTime_ \
topoInstance_ level1_ level2_ inerrProc_ outerrProc_ FECProc_
    rtAgentFunction_ numifs_
...
##### On va remplacer les trois lignes suivantes:
# $node add-interface $chan $propInstance_ $llType_ $macType_ \
# $ifqType_ $ifqlen_ $phyType_ $antType_ $topoInstance_ \
# $inerrProc_ $outerrProc_ $FECProc_
##### Par:
if {[info exist numifs_]}{
for {set i 0} {$i < $numifs_} {incr i} {
# on va ajouter un interface par canal
$node add-interface $chan($i) $propInstance_ $llType_
$macType_ \
$ifqType_ $ifqlen_ $phyType_ $antType_
$topoInstance_ \
$inerrProc_ $outerrProc_ $FECProc_
}
} else {
$node add-interface $chan $propInstance_ $llType_ $macType_
\
$ifqType_ $ifqlen_ $phyType_ $antType_ $topoInstance_ \
$inerrProc_ $outerrProc_ $FECProc_
}
...
}

```

B) Fichier ns-2.35/tcl/lib/ns-mobilenode.tcl

Aucune nouvelle procédure n'a été ajoutée dans ce fichier, mais on a doit faire une certaines modifications aux procédures suivantes :

- Procédure add-interface {}

On va créer une table ARP pour chaque interface.

```

Node/MobileNode instproc add-interface { channel pmodel lltype mactype qtype qlen
    iftype anttype topo inerrproc outerrproc fecproc } {
...
set arptable_($t) [new ARPTable $self $mac]
set arptable $arptable_($t)
...
}

```

- Procédure add-target {agent port}

Tout d'abord, la procédure get-interface qui a été discutée précédemment est appelée, afin que nous puissions évaluer si nous utilisons l'extension multi-interfaces multicanaux. Plus

tard, ce numéro est utilisé pour appeler la nouvelle commande if-queue de l'agent de routage autant de fois que le nombre d'interfaces du nœud.

```
Node/MobileNode instproc add-target { agent port } {
  ...
  # On récupère le nombre d'interfaces par l'objet simulator
  set nbr-interface [$ns get-interface]
  ...
  if { $port == [Node set rtagent_port_] } {
  if {$nbr-interface!=""}{
    for {set i 0} {$i < [Node set nifs_]} {incr i} {
      $agent if-queue $i [$self set ifq_($i)]
    }
  }
  ...
}
```

- Procédure add-target-rtagent {agent port}

Comme nous avons fait précédemment pour la procédure add-target, nous utilisons la procédure get-numifs pour obtenir le nombre d'interfaces du nœud. Ensuite, on utilise le résultat obtenu pour lier notre agent de routage aux entités de la couche liaison de donnée (LL). La variable nbr-interface nous permet de conserver le comportement original du simulateur, puisque le code hérité est toujours valable lorsque cette variable n'a pas de valeur (cas mono-interface mono-canal).

```
Node/MobileNode instproc add-target-rtagent { agent port } {
  ...
  set nbr-interface [$ns get-interface]
  ...
  if { $newapi == "ON" } {
    $agent target $imep_(0)
    $imep_(0) sendtarget $sndT
    # second tracer to see the actual
    # types of tora packets before imep packs them
    if { [info exists toraDebug_] && $toraDebug_ == "ON" } {
      set sndT2 [$self mobility-trace Send "TRP"]
      $sndT2 target $imep_(0)
      $agent target $sndT2
    }
    $sndT target [$self set ll_(0)]
  } else { ;# no IMEP
    if {$nbr-interface != ""} {
      for {set i 0} {$i < [Node set nifs_]} {incr i} {
        set sndT [cmu-trace Send "RTR" $self]
        $agent target $i $sndT
        $sndT target [$self set ll_($i)]
      }
    } else {
      $agent target $sndT
      $sndT target [$self set ll_(0)]
    }
  }
}
```

```
...
}
```

A.3.2.2 Changement au niveau du code c++

Après avoir créé les structures du modèle multi-interfaces multicanaux pour les nœuds mobiles en OTCL, il est nécessaire de les associer correctement à l'arborescence compilée définie en C++. Donc parmi les fichiers qu'on a modifiés :

A) Fichier `/ns-2.35/common/mobilenode.h`

NS2 utilise 2 listes de lien (pointeurs aux nœuds) :

- `prevX_` : Référence pour le nœud précédent.
- `nextX_` : Référence pour le nœud prochain.

Ces pointeurs sont des variables simples, cependant, si nous souhaitons gérer plusieurs canaux, il est nécessaire de créer deux pointeurs tableaux avec autant d'éléments que de canaux dans le scénario de simulation. Ces modifications sont effectuées dans le fichier `mobilenode.h` :

```
class MobileNode : public Node {
...
MobileNode* nextX_[MAXCANAUX]
MobileNode* prevX_[MAXCANAUX]
...
}
```

B) Fichier `ns-2.35/mac/channel.cc`

Pour gérer correctement les différentes interfaces par nos protocoles de routages, on a enregistré l'interface de réception des messages dans le fichier `channel.cc` :

```
Mac802_11::recv(Packet *p, Handler *h)
{
...
if(tx_active_ && hdr->error() == 0) {
hdr->error() = 1;
}
hdr->iface()=addr();
}
```

A.3.2.3 Changement du protocole AODV pour s'adapter au nouveau modèle

L'objectif final de la mise en œuvre du modèle multi-interfaces multicanaux est de l'utiliser par nos agents de routage. Afin d'utiliser cette nouvelle fonctionnalité on a modifié l'architecture du simulateur au niveau de la couche réseau.

Les différents protocoles de routage de nos contributions décident à quelle interface les paquets sortants doivent être acheminés, puisque chaque nœud est associé à plusieurs interfaces. Pour ce faire on a ajouté cinq variables, deux tableaux et deux pointeurs dans le fichier `aodv.h` :

- `MAXINTERFACE` : Nombre maximum d'interface utilisé par un nœud.
- `MAX_ELEMENT` : Nombre maximum des entrées des tables `NeighbourTable` et `nbr_noeud_canalfixe`.
- `int nIfaces`
- `int interface_fixe`
- `int interface_commutable`
- `neighbour_table` : Contient les canaux fixes utilisés par les voisins du nœud.
- `nbr_noeud_canalfixe` : Contient le nombre de nœuds par canal fixe.
- `Ifqueue` : Pointeur de la file d'attente de l'interface réseau située entre le "classificateur" et la "couche liaison de donnée LL" et le pointeur `targetlist` qui va stocker les modules LL de chaque interface.

A.3.3 Implémentation de l'algorithmes génétiques

L'objectif principal de cette section est de formuler le problème de routage à l'aide des algorithmes génétiques permettant de trouver des solutions presque optimales. L'algorithme génétique part en général d'une population appelée population parent choisie aléatoirement. Mais dans notre modélisation, des informations a priori sur le problème de routage sont disponibles via la diffusion des requêtes RREQ, c'est pour ça qu'on génère des individus dans un sous domaine qui respecte les contraintes de routage. Notre algorithme génétique se déroule donc comme suit :

- Création d'une population initiale en se basant sur les paquets RREQ.
- Pour chaque génération.
 - Déterminer une liste d'individus à croiser.
 - Croiser ces individus.
 - Déterminer une liste d'individus à muter.
 - Faire muter ces individus.
 - Injecter ces 2 nouvelles listes d'individus dans la population.
 - Choisir les individus pour la génération suivante.

Pour implémenter notre modèle dans le simulateur NS2, tout d'abord on a ajouté une classe nommée GENETIQUE dans le dossier AODV :

```
class GENETIQUE
{
public:
GENETIQUE(const double probabilite_croisement, const double probabilite_mutation,
          int n, double t[][30], int source, int destination);
int nb_noeuds;
double tt[30][30];
int src;
int dest;
int lncrom;
list<int> *adj; // listes d'adjacence

/* création d'une population initiale*/
void population_initiale ();

/* Création d'une nouvelle population en utilisant le croisement et la mutation
*/
void generation_suivante();

/* meilleur fitness de meilleur chromosome */
double get_meilleurfitness() const;

/* meilleur chemin*/
std::string get_meilleurchemin() const;

const double probabilite_croisement, probabilite_mutation;
int * selection_roulette (double const * const fitness) const;
void mutation(int * const chromosome);
void croisement(const int * const parentA, const int * const parentB, int * const
               offspringA,
};
```

Le code suivant permet d'appeler les méthodes nécessaires pour calculer le plus court chemin entre toutes les paires de nœuds en utilisant l'algorithme génétique :

```
//exemple 30 noeuds
GENETIQUE *gen = new GENETIQUE(0.9, 0.02, 30, t);
size_t generations = 0, generation_sanschangement = 0;
double max_fitness = -1;

while(generation_sanschangement < seuil)
{
gen->generation_suivante();
++generations;
double nv_fitness = gen->get_meilleurfitness();

if(nv_fitness > max_fitness)
{
max_fitness = nv_fitness;
}
```

```
generation_sanschangement = 0;
    }
    else
    {
        ++generation_sanschangement;
    }
}
Path=gen->get_meilleurchemin() ;
```

A.4 Simulation

Nous avons écrit un code TCL pour mettre en place la simulation d'un réseau Ad Hoc. Ce code définit les paramètres de réseau, comme le type de file d'attente (IFQ), le type d'antenne, le modèle de propagation radio, le protocole de routage, le modèle de trafic, le modèle de mobilité, le nombre d'interfaces ...

A.4.1 Génération de trafic

Pour générer les trafics CBR (Constant Bit Rate) entre les nœuds mobiles du réseau on a utilisé le générateur de trafic de NS2. Celui-ci est disponible sous " ns/indep-utils/cmu-scen-gen" (appelée cbrgen.tcl) :

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate
rate]
```

Par exemple, pour créer un fichier de trafic CBR, ayant un maximum de 30 connexions entre 25 nœuds et un taux de 100 paquets par seconde, on utilise la commande suivante :

```
ns cbrgen.tcl -type cbr -nn 25 -seed 1 -mc 30 -rate 100.0 > test
```

A.4.2 Modèle de mobilité

Dans tous nos simulations on a utilisé le modèle de mobilité "Random WayPoint" qu'est disponible sous le répertoire " ns/indep-utils/cmu-scen-gen/setdest". Pour ce faire, il suffit d'utiliser la commande suivante :

```
./setdest [-n nodes] [-p pausetime] [-s maxspeed] [-t simtime] [-x maxx] [-y maxy
]
```

Exemple : considérons un scénario composé de 25 nœuds qui se déplacent avec une vitesse maximale de 2 m/s , une pause moyenne d'une seconde, la durée de simulation est de 100 secondes et la zone de topologie est limitée à 1000m x 1000m. Donc la ligne de commande correspondante à cet exemple est :


```
./setdest -n 25 -p 1.0 -M 2.0 -t 100 -x 1000 -y 1000 > test_modèle_de_mobilité
```

A.5 Conclusion

cette annexe a été consacré aux quelques détails de l'implémentation de nos contributions, il discute les modifications apportées au protocole de routage de base (AODV) dans le but d'introduire la qualité de service dans ce protocole. Il introduit aussi une description des modifications apportées au protocole IEEE 802.11 utilisé au niveau MAC afin d'implémenter le modèle multi-interfaces multicanaux.

Il faut noter ici que les scénarios aléatoires que nous avons réalisés ont pour but de connaître la différence entre le comportement de nos contributions et d'autres protocoles.

Bibliographie

- [1] S. CORSON et J. MACKER. *Mobile Ad hoc Networking (MANET) : Routing Protocol Performance Issues and Evaluation Considerations*. RFC 2501. RFC Editor, 1999.
- [2] T. CLAUSEN et P. JACQUET. *Optimized Link State Routing Protocol (OLSR)*. RFC 3626. <http://www.rfc-editor.org/rfc/rfc3626.txt>. RFC Editor, 2003.
- [3] C. PERKINS, E. BELDING-ROYER et S. DAS. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. <http://www.rfc-editor.org/rfc/rfc3561.txt>. RFC Editor, 2003.
- [4] Zygmunt HAAS, Marc R. PEARLMAN et Prince SAMAR. *The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks*. Internet-Draft draft-ietf-manet-zone-brp-02. Work in Progress. Internet Engineering Task Force, juil. 2002. 13 p.
- [5] Z. J. HAAS et M. R. PEARLMAN. « The performance of query control schemes for the zone routing protocol ». In : *IEEE/ACM Transactions on Networking* 9.4 (2001), p. 427-438.
- [6] David A. JOHNSON David B. and Maltz. « Dynamic Source Routing in Ad Hoc Wireless Networks ». In : *Mobile Computing*. Sous la dir. de Tomasz IMIELINSKI et Henry F. KORTH. Boston, MA : Springer US, 1996, p. 153-181.
- [7] David B. JOHNSON, David A. MALTZ et Yih-Chun HU. *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*. Internet-Draft draft-ietf-manet-dsr-09. Internet Engineering Task Force, 2003. 97 p.
- [8] Charles E. PERKINS et Pravin BHAGWAT. « Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers ». In : *SIGCOMM Comput. Commun. Rev.* 24.4 (oct. 1994), p. 234-244.
- [9] Zygmunt J. HAAS, Marc R. PEARLMAN et Prince SAMAR. *The Interzone Routing Protocol (IERP) for Ad Hoc Networks*. Internet-Draft draft-ietf-manet-zone-ierp-02. 2002. 11 p.
- [10] Zygmunt J. HAAS, Marc R. PEARLMAN et Prince SAMAR. *The Intrazone Routing Protocol (IARP) for Ad Hoc Networks*. Internet-Draft draft-ietf-manet-iarp-01. 2001. 8 p.
- [11] Katia RUNSER. « Méthodologies pour la planification des Réseaux locaux sans-fil ». Thèse de doct. Institut National des Sciences Appliquées de Lyon, 2005.
- [12] Bob O'HARA. *The IEEE 802.11 Handbook : A Designer's Companion*. Institute of Electrical Electronics Engineers (IEEE), 2004.
- [13] Koichiro BAN et Hamid GHARAVI. « IEEE 802.11 FHSS receiver design for cluster-based multihop video communications ». In : *Wireless Communications and Mobile Computing* 2.6 (), p. 595-605. eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcm.85>.
- [14] D. VASSIS et al. « The IEEE 802.11g standard for high data rate WLANs ». In : *IEEE Network* 19.3 (2005), p. 21-26.

-
- [15] I. J. MEEL. « Spread spectrum (SS) introduction ». In : *Sirius Communications -Rotselaar -Belgium, Studiedag Spread Spectrum*. T. 2. 1999, p. 1-33.
- [16] M. L. DOELZ, E. T. HEALD et D. L. MARTIN. « Binary Data Transmission Techniques for Linear Systems ». In : *Proceedings of the IRE* 45.5 (1957), p. 656-661.
- [17] R. W. CHANG. « Synthesis of band-limited orthogonal signals for multichannel data transmission ». In : *The Bell System Technical Journal* 45.10 (1966), p. 1775-1796.
- [18] B. SALTZBERG. « Performance of an Efficient Parallel Data Transmission System ». In : *IEEE Transactions on Communication Technology* 15.6 (1967), p. 805-811.
- [19] L. HANZO et al. *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs and Broadcasting*. Wiley - IEEE. Wiley, 2005.
- [20] M. KRUNZ, A. MUQATTASH et Sung-Ju LEE. « Transmission power control in wireless ad hoc networks : challenges, solutions and open issues ». In : *IEEE Network* 18.5 (2004), p. 8-14.
- [21] P. GUPTA et P. R. KUMAR. « The capacity of wireless networks ». In : *IEEE Transactions on Information Theory* 46.2 (2000), p. 388-404.
- [22] Swetha NARAYANASWAMY et al. « Power Control in Ad-Hoc Networks : Theory, Architecture, Algorithm and Implementation of the COMPOW Protocol ». In : *in European Wireless Conference*. 2002, p. 156-162.
- [23] S. MEGUERDICHIAN et al. « Coverage problems in wireless ad hoc sensor networks ». In : *2nd conference internationale ACM sur les réseaux sans fil et de sensors et leurs applications*. MobiCom '01. San Diego, USA, 2003.
- [24] C.H.R.LIN et M.GERLA. « Asynchronous Multimedia multi hop radio network ». In : *IEEE INFOCOM, Kobe, Japan*. 1997, p. 118-125.
- [25] Suresh SINGH et C. S. RAGHAVENDRA. « PAMAS—Power Aware Multi-access Protocol with Signalling for Ad Hoc Networks ». In : *SIGCOMM Comput. Commun. Rev.* 28.3 (juil. 1998), p. 5-26.
- [26] Kwang-Cheng CHEN. « Medium access control of wireless LANs for mobile computing ». In : *IEEE Network* 8.5 (1994), p. 50-63.
- [27] C. WU et V. LI. « Receiver-initiated Busy-tone Multiple Access in Packet Radio Networks ». In : *SIGCOMM Comput. Commun. Rev.* 17.5 (août 1987), p. 336-342.
- [28] João L. SOBRINHO et A. S. KRISHNAKUMAR. « Real-time traffic over the IEEE 802.11 medium access control layer ». In : *Bell Labs Technical Journal* 1.2 (), p. 172-187. eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/bltj.2023>.
- [29] A. ADYA et al. « A multi-radio unification protocol for IEEE 802.11 wireless networks ». In : *First International Conference on Broadband Networks*. 2004, p. 344-354.
- [30] Jing ZHU et S. ROY. « 802.11 mesh networks with two-radio access points ». In : *IEEE International Conference on Communications, 2005. ICC 2005. 2005*. T. 5. 2005, 3609-3615 Vol. 5.

-
- [31] Jungmin SO et Nitin VAIDYA. « Multi-Channel MAC for Ad Hoc Networks : Handling Multi-Channel Hidden Terminals Using a Single Transceiver ». In : *In ACM MobiHoc*. 2004, p. 222-233.
- [32] Krishna N RAMACHANDRAN et al. « Interference Aware Channel Assignment in Multi-Radio Wireless Mesh Networks ». In : *25th IEEE International Conference on Computer Communications*. 2006, p. 1-12.
- [33] Pradeep KYASANUR et Nitin H. VAIDYA. « Routing and Link-layer Protocols for Multi-channel Multi-interface Ad Hoc Wireless Networks ». In : *SIGMOBILE Mob. Comput. Commun. Rev.* 10.1 (jan. 2006), p. 31-43.
- [34] Andy An-Kai JENG et al. « Release-time-based multi-channel MAC protocol for wireless mesh networks ». In : *Computer Networks* 55.9 (2011), p. 2176 -2195.
- [35] Amjad ALI et al. « A Survey of MAC Protocols Design Strategies and Techniques in Wireless Ad Hoc Networks ». In : *Journal of Communications*. 9.1 (2014).
- [36] Shigang CHEN et K. NAHRSTEDT. « Distributed quality-of-service routing in ad hoc networks ». In : *IEEE Journal on Selected Areas in Communications* 17.8 (1999), p. 1488-1505.
- [37] Zhenhui ZHAI et al. « A reliable and adaptive AODV protocol based on cognitive routing for Ad hoc networks ». In : *2010 The 12th International Conference on Advanced Communication Technology (ICACT)*. T. 2. 2010, p. 1307-1310.
- [38] M TAMILARASI et al. « Integrated Energy Aware Mechanism for MANETs using On-demand Routing ». In : (avr. 2018).
- [39] Anita YADAV, Y. N. SINGH et R. R. SINGH. « Improving Routing Performance in AODV with Link Prediction in Mobile Adhoc Networks ». In : *Wirel. Pers. Commun.* 83.1 (juil. 2015), p. 603-618.
- [40] Xiaoxia QI, Qijin WANG et Fan JIANG. « Multi-path Routing Improved Protocol in AODV Based on Nodes Energy ». In : 8 (fév. 2015), p. 207-214.
- [41] S. AMIRI et al. « Enhanced lifetime maximization algorithm for wireless sensor network ». In : *2011 7th International Wireless Communications and Mobile Computing Conference*. 2011, p. 302-307.
- [42] H. LI. « LEACH-HPR : An energy efficient routing algorithm for Heterogeneous WSN ». In : *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*. T. 2. 2010, p. 507-511.
- [43] N. MEJRI et F. KAMOUN. « Algorithme de routage hiérarchique mheed à plusieurs sauts pour les grands réseaux de capteurs ». In : *4th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications*. 2007, p. 1-7.
- [44] T. ISSARIYAKUL et E. HOSSAIN. « Introduction to Network Simulator NS2 ». In : *Springer, Oct. 2008, ISBN : 978-0-387-71759-3*.
- [45] Edgar W. WOOLARD. « Investigations on the Theory of the Brownian Movement ». In : *The American Mathematical Monthly* 35.6 (1928), p. 318-320.

-
- [46] Tracy CAMP, Jeff BOLENG et Vanessa DAVIES. « A survey of mobility models for ad hoc network research ». In : *Wireless communications and mobile computing 2.5* (2002), p. 483-502.
- [47] Josh BROCH et al. « A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols ». In : *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*. MobiCom '98. Dallas, Texas, USA : ACM, 1998, p. 85-97.
- [48] E. M. ROYER, P. M. MELLIAR-SMITH et L. E. MOSER. « An analysis of the optimum node density for ad hoc mobile networks ». In : *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*. T. 3. 2001, 857-861 vol.3.
- [49] B. LIANG et Z. J. HAAS. « Predictive distance-based mobility management for PCS networks ». In : *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. T. 3. 1999, 1377-1384 vol.3.
- [50] R. WINTER et al. « CrossTalk : cross-layer decision support based on global knowledge ». In : *IEEE Communications Magazine* 44.1 (2006), p. 93-99.
- [51] Raymond KNOPP et al. « Overview of the WIDENS architecture, a wireless ad hoc network for public safety ». In : *SECON 2004, 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks, October 4-7, 2004, Santa Clara, USA*. Santa Clara, ÉTATS-UNIS, oct. 2004.
- [52] Vijay T. RAISINGHANI et Tata INFOTECH. « ECLAIR : An efficient cross layer architecture for wireless protocol stacks ». In : *In World Wireless Congress, SF*. 2004, p. 2004.
- [53] N. KET et S. HIPPARGI. « Modified AODV Energy Aware Routing for optimized performance in mobile ad-hoc networks ». In : *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2016, p. 1030-1034.
- [54] V. KAWADIA et P. R. KUMAR. « A cautionary perspective on cross-layer design ». In : *IEEE Wireless Communications* 12.1 (2005), p. 3-11.
- [55] Vasudev SHAH et S. KRISHNAMURTHY. « Handling Asymmetry in Power Heterogeneous Ad Hoc Networks : A Cross Layer Approach ». In : *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*. 2005, p. 749-759.
- [56] Shigang CHEN et K. NAHRSTEDT. « Distributed quality-of-service routing in ad hoc networks ». In : *IEEE Journal on Selected Areas in Communications* 17.8 (1999), p. 1488-1505.
- [57] Qi XUE et Aura GANZ. « Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks ». In : *Journal of Parallel and Distributed Computing* 63.2 (2003). Routing in Mobile and Wireless Ad Hoc Networks, p. 154 -165.
- [58] A. MUNARETTO et al. « A link-state QoS routing protocol for ad hoc networks ». In : *4th International Workshop on Mobile and Wireless Communications Network*. 2002, p. 222-226.

- [59] L. ROMDHANI et C. BONNET. « A cross-layer on-demand routing protocol for delay-sensitive applications ». In : *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*. T. 2. 2005, 994-998 Vol. 2.
- [60] A. VERES et al. « Supporting service differentiation in wireless packet networks using distributed control ». In : *IEEE Journal on Selected Areas in Communications* 19.10 (2001), p. 2081-2093.
- [61] M. BARRY, A. T. CAMPBELL et A. VERES. « Distributed control algorithms for service differentiation in wireless packet networks ». In : *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*. T. 1. 2001, 582-590 vol.1.
- [62] C. K. BOUNSAYTHIP. « Algorithmes heuristiques et évolutionnaires : application à la résolution du problème de placement de formes irrégulières ». Thèse de doct. université des sciences et technologies de Lille, 1998.
- [63] David E. GOLDBERG. *Genetic Algorithms in Search, Optimization, and Machine Learning*. New York : Addison-Wesley, 1989.
- [64] Ramon Aguero CALVO et Jesus Perez CAMPO. *Adding multiple interface support in NS-2*. 2007.

Liste des Publications

Articles

1. **Faouzi H.**, Mouncif H. and Lamsaadi M., 2017. Improving the QoS for ad hoc Networks Using and Comparing Genetic and Dynamic Programming. *Journal of Applied Sciences*, 17 : 1-11.
2. **Faouzi H.**, Mouncif H. and Lamsaadi M, (2016). AODV energy routing mechanism for multi-channel multi-interface ad hoc networks (EMCMI-AODV) using a dynamic programming algorithm. *International Journal of Mobile Computing and Multimedia Communications*. 7. 1-16. 10.4018/IJMCMC.2016100101.
3. **Faouzi H.**, Er-rouidi M., Moudni H., Mouncif H., Lamsaadi M. (2017) Improving Network Lifetime of Ad Hoc Network Using Energy Aodv (E-AODV) Routing Protocol in Real Radio Environments. In : El Abbadi A., Garbinato B. (eds) *Networked Systems. NETYS 2017. Lecture Notes in Computer Science*, vol 10299. Springer, Cham
4. **Faouzi H.**, Mouncif H., Lamsaadi M. (2015) Performance Evaluation for Ad hoc Routing Protocols in Realistic Physical Layer. In : Bouajjani A., Fauconnier H. (eds) *Networked Systems. NETYS 2015. Lecture Notes in Computer Science*, vol 9466. Springer, Cham
5. Moudni H., Er-rouidi M.,**Faouzi H.**, Mouncif H., El Hadadi B. (2017) Enhancing Security in Optimized Link State Routing Protocol for Mobile Ad Hoc Networks. In : Sabir E., García Armada A., Ghogho M., Debbah M. (eds) *Ubiquitous Networking. UNet 2017. Lecture Notes in Computer Science*, vol 10542. Springer, Cham
6. Er-rouidi M., Moudni H., **Faouzi H.**, Mouncif H., Merbouha A. (2017) A Fuzzy-Based Routing Strategy to Improve Route Stability in MANET Based on AODV. In : El Abbadi A., Garbinato B. (eds) *Networked Systems. NETYS 2017. Lecture Notes in Computer Science*, vol 10299. Springer, Cham
7. Mohamed ER-ROUIDI, Houda MOUDNI, **Hassan FAOUZI**, Hicham MOUNCIF, Abdelkrim MERBOUHA, "Enhancing Energy Efficiency of Reactive Routing Protocol in Mobile Ad-Hoc Network with Prediction on Energy Consumption", *The International Arab Journal of Information Technology (IAJIT)* (pp. 1-7).
8. Mohamed ER-ROUIDI, Houda MOUDNI, **Hassan FAOUZI**, Hicham MOUNCIF, Abdelkrim MERBOUHA, "Improving Performance of Mobile Ad Hoc Network Using Clustering Schemes", *The International Journal of Informatics and Communication Technology (IJ-ICT)* Vol. 6, No. 2, August 2017, pp. 69 75.

Conférences internationales

1. **Hassan Faouzi**, Mohamed Er-rouidi, Houda Moudni, Hicham Mouncif, Mohamed LAMSAADI “Implementation and Comparison of AODV and DSDV in Multi-Channel Multi-Interface Ad Hoc Wireless Networks” , ACIT’2016 Sultan Moulay Slimane University Beni Mellal, Morocco.
2. **Faouzi H.**, Mouncif H., Lamsaadi M. (2015) Performance Evaluation for Ad hoc Routing Protocols in Realistic Physical Layer. NETYS 2015. 13-15 May ; Agadir, Morocco
3. **Hassan Faouzi**, Mohamed Er-rouidi, Houda Moudni, Hicham Mouncif, Mohamed LAMSAADI, “Implementation and Comparison of AODV and DSDV in Multi-Channel Multi-Interface Ad Hoc Wireless Networks” , The International Arab Conference on Information Technology Sultan Moulay Slimane University, Beni-Mellal, Morocco December 6-8, 2016.
4. **Hassan Faouzi** and Mohamed FAKIR, “Automatic exudates detection in diabetic retinopathy images”, The first International Conference On Business Intelligence (CBI’14) ,April 29-30, 2014, Beni Mellal, Morocco.
5. Mohamed ER-ROUIDI, Houda MOUDNI, **Hassan FAOUZI**, Hicham MOUNCIF and Abdelkrim MERBOUHA, “Enhancing Energy Efficiency of Reactive Routing Protocol in Mobile Ad-Hoc Network with Prediction on Energy Consumption”, The International Arab Conference on Information Technology Sultan Moulay Slimane University, Beni-Mellal, Morocco December 6-8, 2016.
6. Houda MOUDNI, Mohamed ER-ROUIDI, **Hassan FAOUZI**, Hicham MOUNCIF, Benachir EL HADADI, “Intrusion Detection System in Mobile Ad-hoc Networks using Machine Learning Techniques”, The Third International Conference on Business Intelligence (CBI’17), March 29-31, 2017, Beni Mellal, Morocco.
7. **Faouzi H.**, Er-rouidi M., Moudni H., Mouncif H., Lamsaadi M. (2017) Improving Network Lifetime of Ad Hoc Network Using Energy Aodv (E-AODV) Routing Protocol in Real Radio Environments. International Conference, NETYS 2017, Marrakech, Morocco, May 17-19, 2017

