

Application au routage multicritère dans les réseaux ad hoc sans fil

4.1 Introduction

Les réseaux ad hoc sans fil sont des réseaux auto-adaptatifs formés par un ensemble de terminaux sans fil qui établissent une topologie afin de communiquer. L'objectif de la tâche du routage dans ces réseaux est de trouver un chemin efficace (au sens large) entre un nœud source donné et un nœud destinataire. Comme la portée des nœuds est relativement petite, les chemins utilisés pour la communication sont souvent composés de plusieurs nœuds intermédiaires. De plus, la tâche du routage doit optimiser un certain nombre de critères relatifs à la qualité de service QoS (*Quality of Service*), en l'occurrence le délai, le taux de perte, la gigue, etc. Dans ce contexte, la majorité des protocoles de routage introduits dans la littérature prennent uniquement le nombre de sauts comme métrique à optimiser et négligent, par conséquent, les exigences des applications en termes de différents critères de QoS. Cependant, le problème de routage dans les réseaux ad hoc sans fil est typiquement une problématique de décision multicritère, dans ce sens que plusieurs critères, souvent contradictoires, sont impliqués dans le processus de décision. Dans ce chapitre, nous comparerons la méthode de la poupée russe à la méthode de la somme pondérée, dans le cas du routage dans les réseaux sans fil. Ce problème consiste à choisir le meilleur chemin en fonction des exigences

des applications du réseau. Par exemple, une application de type VoIP peut accepter des taux de pertes relativement élevés, contrairement au délai qui doit être petit. D'un autre côté, une application de type transfert de fichier tel que FTP (*File Transfer Protocol*) est sensible au critère du taux de perte, par contre, elle peut accepter des délais relativement élevés. La figure 4.1 illustre schématiquement la problématique de routage multicritère dans les réseaux ad hoc sans fil. La partie droite de la figure illustre la topologie du réseau qui comporte plusieurs chemins entre le nœud source S et le nœud destinataire D . Chaque chemin est caractérisé par ses propres métriques (ou critères) tels que le délai, le taux de perte, la gigue, etc. Quant à la figure de gauche, celle-ci représente les alternatives non-dominées au niveau de la source. Chaque point dans l'espace des critères représente un chemin distinct dans le réseau. Nous pouvons remarquer qu'une problématique existe en ce qui concerne les chemins appartenant au front de Pareto. En effet, le chemin A est bon en termes de délai, mais relativement mauvais en termes de taux de perte. L'inverse est vrai pour le chemin B . De ce fait, le protocole de routage doit s'appuyer sur une méthode de décision multicritère, qui devra gérer le processus du choix des chemins en fonction des exigences des applications routées.

Ce chapitre est organisé comme suit. Premièrement nous exposons, dans la section 4.2, les travaux de recherche qui ont tenté d'incorporer la notion de décision multicritère dans le routage. Ensuite, la section 4.3 donne un aperçu du principe de l'apprentissage par renforcement qui est utilisé dans le protocole de routage. Puis, nous introduisons, dans la section 4.4, le protocole de routage, qui utilise la méthode de la poupée russe, afin de trouver les chemins les plus adaptés aux différentes applications, selon leurs exigences. Ensuite, la section 4.5 montre une évaluation des performances de la méthode de la poupée russe comparée à la méthode de la somme pondérée. Enfin, nous concluons ce chapitre dans la section 4.6.

4.2 État de l'art

La majorité des protocoles de routage introduits dans la littérature utilisent les poids de quantification afin de marquer l'importance des critères de QoS dans le pro-

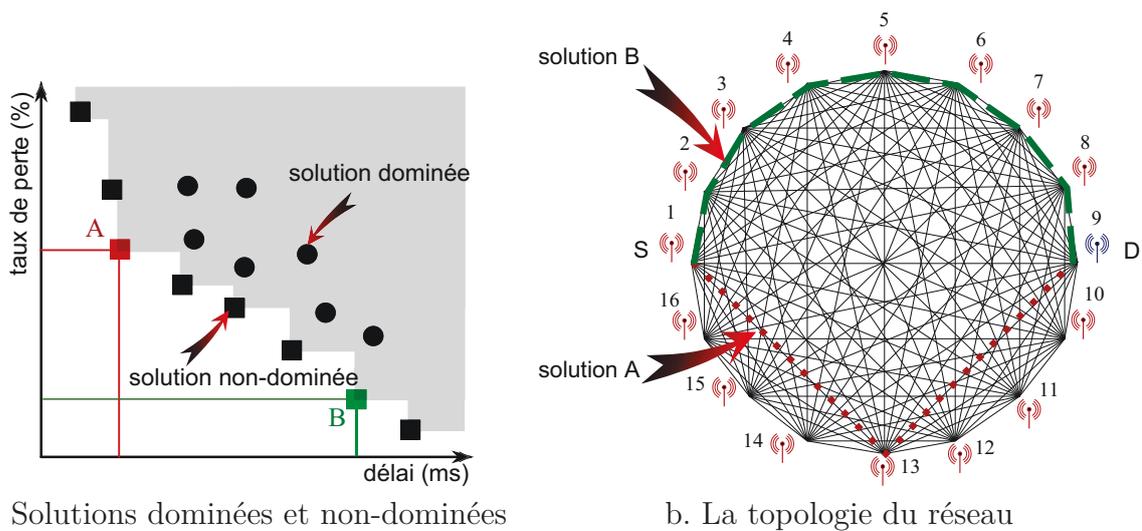


FIGURE 4.1 – Problématique de routage multicritère dans un réseau sans fil

cessus de routage. Une fois que l'importance des critères est quantifiée, une somme pondérée des critères est utilisée afin de transformer le problème de routage multicritère en un problème de routage monocritère. Cependant, dans la majorité des études, aucune indication n'est donnée sur la méthode utilisée pour choisir la valeur des poids de quantification. Dans cette section, nous passerons en revue les études pertinentes qui ont tenté d'apporter des solutions au problème de routage dans les réseaux ad hoc sans fil.

Dans [ED06], les auteurs minimisent une somme pondérée de deux critères, en l'occurrence le délai et la distance entre la source et la destination, afin d'optimiser le processus de routage dans un réseau ad hoc. Les auteurs fixent les valeurs des poids pondérant les deux critères moyennant plusieurs scénarios de simulation. Similairement, dans [MTT⁺06], les auteurs utilisent la méthode de la somme pondérée pour optimiser trois critères, notamment le délai, l'énergie et le taux de perte binaire BER (*Bit Error Rate*) dans un réseau sans fil. Le choix des valeurs des poids utilisés dans cette étude est laissé libre à l'utilisateur. Dans [GDRMB09], les auteurs formalisent le problème de routage comme un problème de décision multicritère et proposent de le résoudre moyennant une somme pondérée. Cependant, aucune méthode n'est proposée pour fixer les valeurs des poids relatifs aux critères. Par contre, les auteurs ont présenté une étude de sensibilité du processus de décision multicritère par rapport aux

valeurs des poids. Les auteurs concluent par la remarque de l'importance de l'estimation des valeurs des poids et observent que ces derniers dépendent considérablement des exigences des applications routées. Dans [SK06], les auteurs traitent le problème du choix de la meilleure connexion réseau comme un problème de décision multicritère. Le but est d'optimiser deux critères contradictoires, en l'occurrence le coût (prix) et le temps de transmission, quand un ensemble d'applications (données, vidéos, voix, etc) sont transférées via plusieurs réseaux de communication. Le problème est résolu par la minimisation de la distance de Chebyshev à un point de référence. Comme dans les études présentées précédemment, les auteurs affirment que les poids utilisés dans l'expression de la distance doivent être introduits par le décideur en personne. Cependant, les auteurs défendent leur travail, en stipulant que le choix de la solution finale est peu dépendant des valeurs des poids, contrairement au choix de la solution de référence. Dans [REM08], les auteurs ont utilisé la distance de *Tchebychev* au vecteur de critère idéal pour sélectionner un chemin parmi l'ensemble des chemins non-dominés qui existent entre une source et une destination dans un réseau de données. L'ensemble des chemins non-dominés est obtenu au moyen d'un algorithme dédié [GBR06, Mar84], quant au vecteur idéal, il est obtenu en prenant la valeur optimale de chaque critère parmi l'ensemble des chemins non-dominés. Bien que cette méthode permet de résoudre la problématique de prise de décision multicritère en temps réel, elle souffre néanmoins des inconvénients relatifs à la méthode de la distance à un objectif de référence, notamment le choix des valeurs des poids de pondération utilisés dans l'expression de la distance de *Tchebychev* (voir la section 2.4.3).

Dans [RD04], les auteurs ont tenté de résoudre le problème du routage *multicast* en utilisant les algorithmes génétiques multiobjectifs (MOGA) [FF⁺93]. Le but de ce travail est de construire un arbre de routage *multicast* qui optimise deux critères en l'occurrence le délai et la bande passante. Le choix du meilleur arbre *multicast* est effectué par le décideur à la fin de l'optimisation (quand l'algorithme génétique trouve les solutions non-dominées) en fonction des exigences des applications. Dans [Ish06], les auteurs utilisent une solution de référence fournie par le décideur, afin de guider le processus de recherche lors de l'étape de sélection de l'algorithme génétique

multiobjectif (MOGA). Cependant, un choix adéquat de la solution de référence est exigé, car la méthode en dépend grandement.

4.3 Aperçu sur l'apprentissage par renforcement

L'apprentissage par renforcement [Sut98] est une branche de l'intelligence artificielle qui permet de reproduire le mécanisme d'apprentissage des animaux par des ordinateurs. En effet, un animal apprend à connaître son environnement au moyen du mécanisme action/récompense, qui est aussi le fondement de l'apprentissage par renforcement. La figure 4.2 illustre le schéma général de fonctionnement d'un système utilisant l'apprentissage par renforcement. Dans ce contexte, le système est représenté par un agent composé de plusieurs états, qui ne sont pas nécessairement connus à l'avance. L'agent choisit les meilleures actions qui maximisent l'espérance de ses gains et se déplace d'un état à un autre, en fonction de la réponse de l'environnement. Mathématiquement parlant, soit \mathcal{S} l'ensemble des états possibles dans le système. A chaque instant t , l'agent perçoit un état $s_t \in \mathcal{S}$ de son environnement et sélectionne une action $a_t \in \mathcal{A}(s_t)$, où $\mathcal{A}(s_t)$ est l'ensemble des actions possibles à l'état s_t . En retour, l'agent reçoit de son environnement une récompense immédiate $r_{t+1} \in \mathcal{R}$, où \mathcal{R} est l'ensemble des récompenses, puis se déplace à l'état s_{t+1} (voir la figure 4.2). A chaque instant t , l'agent suit une stratégie (ou politique) donnée, notée $\pi_t(s, a)$, qui est la probabilité que l'agent choisisse l'action a_t sachant que ce dernier est dans l'état s_t . Les méthodes d'apprentissage par renforcement adaptent la politique de l'agent en fonction de son expérience dans le temps. Pour ce faire, les méthodes d'apprentissage par renforcement définissent une fonction de valeur (*value function*) qui tente de prévoir l'espérance des gains futurs que l'agent peut recevoir s'il se trouve dans un certain état, ou s'il exécute une certaine action en se trouvant dans un certain état. Plus précisément, la fonction de valeur $V^\pi(s)$ d'un état s suivant la politique π est définie, par exemple, comme suit :

$$V^\pi(s) = E_\pi \{R_t/s_t = s\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}/s_t = s \right\}$$

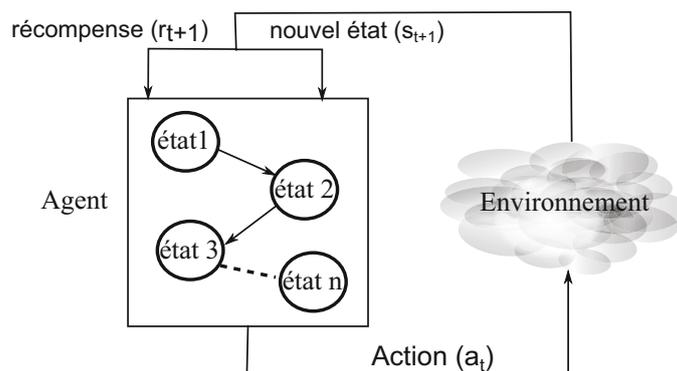


FIGURE 4.2 – Principe de l'apprentissage par renforcement

où $E_{\pi} \{ \cdot \}$ est l'espérance mathématique. $\gamma \in [0, 1]$ est un nombre réel qui quantifie l'importance des récompenses futures dans le processus d'apprentissage par renforcement : une valeur de γ proche de 1 signifie que les récompenses futures sont importantes dans le calcul de la fonction de valeur. $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$.

De même, la fonction de valeur d'un couple état/action (*action-value function*) $Q^{\pi}(s, a)$ suivant la politique π est définie comme suit :

$$Q^{\pi}(s, a) = E_{\pi} \{ R_t / s_t = s, a_t = a \} = E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} / s_t = s, a_t = a \right\}$$

Les valeurs $V^{\pi}(s)$ et $Q^{\pi}(s, a)$ peuvent être estimées en calculant leur moyennes empiriques au cours du temps. Pour ce faire, il suffit que l'agent suive la politique π puis calcule progressivement la moyenne empirique en fonction des récompenses qu'il reçoit de son environnement. La moyenne convergera dans le temps vers la valeur $V^{\pi}(s)$, respectivement $Q^{\pi}(s, a)$, si le nombre d'occurrences de l'état s , respectivement couple état/action (s, a) est assez grand. Notons que ce procédé d'estimation de la valeur de $V^{\pi}(s)$ et $Q^{\pi}(s, a)$ est appelé la méthode de Monte Carlo. Il existe d'autres méthodes qui permettent aussi d'estimer les valeurs des fonctions de valuation, notamment les méthodes dites *TD-learning* (*Temporal-Difference Learning*) ou encore *Q-Learning* [Sut98].

Le but de l'agent est de constituer une politique optimale π^* qui permet de maximiser ses gains. En d'autres termes, l'agent aspire à constituer une politique π^* dont la fonction de valeur $V^*(s)$ est définie comme suit :

$$V^*(s) = \max_{\pi} V^{\pi}(s)$$

La politique π^* doit aussi maximiser la fonction de valeur état-action comme suit :

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

Finalement, nous pouvons formaliser l'espérance des gains que percevra un agent se trouvant dans l'état s et exécutant l'action a , suivant la politique optimale π^* , comme suit :

$$Q^*(s, a) = E \{r_{t+1} + \gamma V^*(s_{t+1}) / s_t = s, a_t = a\} \quad (4.1)$$

L'expression 4.1 découle directement du principe d'optimalité de *Bellman* [Sut98].

4.3.1 Application au problème de routage multicritère

Dans le cas de la problématique de routage, tous les nœuds du réseau sont considérés comme des agents. De plus, pour chaque application dans le réseau, tel que la VoIP, la vidéo ou le transfert de fichier, il existe une politique de routage π spécifique qui satisfasse plus au moins ses exigences en termes de QoS. Notons que le problème de recherche de l'ensemble des routes multicritères optimales (front de Pareto) est un problème NP-difficile [WC96]. Par conséquent, notre approche de routage fondée sur le principe d'apprentissage par renforcement est une heuristique qui aspire à approcher le front de Pareto du problème en un temps raisonnable.

Soient a, s, n, d la classe de l'application, le nœud source, un nœud voisin et le nœud destination, respectivement. Un nœud r doit choisir un nœud voisin n^* en fonction de l'application a et la destination d de telle façon que le vecteur de critères au niveau de la destination satisfasse les exigences multicritères de l'application a . Les vecteurs de critères sont considérés comme des réalisations d'un vecteur aléatoire \mathbf{c} . Chaque nœud du réseau possède une table de routage (fonction de valeur) dans laquelle il mémorise l'estimation des espérances des vecteurs de critères $\mathbf{Q}(a, d, r, n) = E \{\mathbf{c}/a, d, r, n\}$. Les paramètres a et d sont contenus dans le paquet d'apprentissage introduit dans la sous-

4.3. Aperçu sur l'apprentissage par renforcement

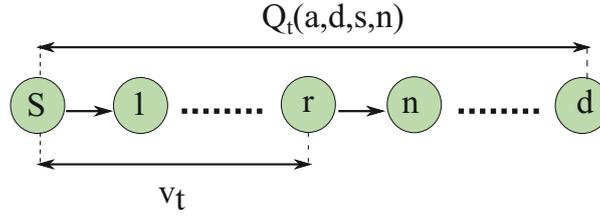


FIGURE 4.3 – Illustration d'un chemin entre la source et la destination

section suivante. L'espérance des récompenses est mise à jour au niveau de chaque source du réseau comme suit :

$$\begin{cases} \mathbf{Q}_t(a, d, s, n) = (1 - \gamma) \mathbf{Q}_{t-1}(a, d, s, n) + \gamma \mathbf{c}_t & \text{si } t > 0 \\ \mathbf{Q}_0(a, d, s, n) = 0 & \text{sinon} \end{cases} \quad (4.2)$$

où \mathbf{c}_t est la récompense immédiate calculée au niveau de la destination pour le paquet numéro t . Les coordonnées de \mathbf{c}_t sont les différents critères du problème traités, en l'occurrence les critères de qualité de service pour l'application de routage.

Finalement, le meilleur voisin n^* pour le nœud r est le voisin qui maximise l'espérance de la qualité de service multicritère $\mathbf{Q}(a, d, r, n)$. Le meilleur voisin n^* est choisi par la méthode de décision multicritère utilisée comme suit :

$$n^* = \arg \max_{n \in \mathbf{N}(r)} u(\mathbf{Q}_t(a, d, r, n) + \mathbf{v}_t) \quad (4.3)$$

Où,

- $u(\mathbf{c})$ est une fonction d'utilité ;
- $u(\mathbf{c}) = m - \Delta(\mathbf{c})$ pour la méthode de la poupée russe, où m est l'index de la plus grande boîte de la poupée russe et $\Delta(\mathbf{c})$ est la distance au système de boîtes englobantes ;
- $u(\mathbf{c}) = 1 - \sum_{i=1}^n w_i c_i$ pour la méthode de la somme pondérée, tel que $\sum_{i=1}^n w_i = 1$ et c_i est la valeur du critère i qui est normalisé de telle sorte que $c_i \in [0, 1]$;
- $\mathbf{N}(r)$ est la liste des voisins directs du nœud r ;
- \mathbf{v}_t est le vecteur des critères relatif au chemin (suivi par le paquet t) entre le nœud source s et le nœud courant r (voir la figure 4.3). $\mathbf{v}_t = 0$ au niveau de la source s .

4.4 Protocole de routage multicritère utilisant l'apprentissage par renforcement

Les méthodes classiques de routage dans les réseaux ad hoc sans fil ne sont pas multicritères, ce qui justifie la conception d'un nouveau protocole de routage. Dans cette section, nous présentons le protocole RLRP (*Reinforcement Learning Routing protocol*) qui est un protocole de routage multicritère fondé sur le principe de l'apprentissage par renforcement.

4.4.1 Aperçu général du fonctionnement du protocole RLRP

Le protocole RLRP opère en deux phases principales, en l'occurrence la phase de découverte et maintenance de routes et la phase de routage des paquets de données. Dans la phase de découverte et maintenances de routes, chaque source (un nœud donné s dans le réseau qui envoie des paquets à une destination d) envoie des paquets spéciaux, appelés "paquets d'apprentissage", afin de découvrir une ou plusieurs routes vers la destination. Pour ce faire, le nœud source s choisit aléatoirement un voisin de sa liste des voisins directs et lui envoie un paquet d'apprentissage appelé, *Forward Learning Packet* (FLP) qui est destiné à la destination d . Le voisin direct du nœud s qui reçoit le paquet FLP examine sa table de routage, pour savoir s'il existe ou pas un chemin vers la destination d . Dans le cas où un chemin existe, le voisin direct de la source s envoie le paquet FLP au prochain nœud sélectionné par la méthode de décision multicritère. Dans le cas contraire, le voisin direct de la source sélectionne aléatoirement un nœud dans sa liste de voisins directs et lui envoie à son tour le paquet FLP. Ainsi, de proche en proche, le paquet FLP transite entre les nœuds intermédiaires, jusqu'à ce qu'il arrive au niveau de la destination d . Cependant, le paquet FLP peut être supprimé au cours du temps, s'il passe par un nœud donné pour la deuxième fois (cas d'une boucle) ou si le nombre de nœuds visités dépasse un seuil prédéfini. Notons que, pendant son passage de la source vers la destination le paquet FLP est mis à jour de plusieurs façons. Premièrement, la liste des nœuds déjà visités est stockée dans un champ spécial appelé "mémoire". Enfin, les différents critères, tels que le nombre

4.4. Protocole de routage multicritère utilisant l'apprentissage par renforcement

de sauts, le délai et le taux de perte, sont estimés au fur et à mesure que le paquet FLP s'approche de la destination. Au niveau de la destination, le paquet FLP est transformé en un paquet d'apprentissage, appelé *Backward Learning Packet* (BLP) qui a comme source la destination d et comme destination la source s . Le paquet BLP suit le chemin inverse emprunté par le paquet FLP, en parcourant la liste des nœuds stockés dans la mémoire du paquet FLP, de la fin jusqu'au début, jusqu'à ce qu'il atteigne la source s . Notons qu'à l'image d'un paquet FLP, un paquet BLP peut être perdu en route, à cause des problèmes de transmission. Dans le cas où le paquet BLP atteint la source, cette dernière l'utilise pour mettre à jour sa table de routage au moyen de l'expression 4.2. Si le paquet est perdu, la source pénalise dans sa table de routage le voisin responsable. Ainsi, de proche en proche, les nœuds découvrent des chemins divers vers la destination. Ces chemins optimiseront différemment les critères de qualité de service, ce qui nous amène à une problématique de décision multicritère des chemins en fonction des exigences des différentes applications.

4.4.2 Présentation détaillée

Vue statique du protocole RLRP

La figure 4.4 illustre les différentes composantes qui forment le protocole RLRP. Comme nous pouvons le constater sur le diagramme, le protocole RLRP comprend une classe principale, appelée RLRP, qui est instanciée dans tous les nœuds du réseau. Cette classe porte l'identificateur du nœud en question. La classe RLRP englobe plusieurs autres classes qui représentent l'aspect statique du système :

- Voisin : un nœud voisin est identifié par son identificateur et l'instant auquel il est rentré en contact avec le nœud courant.
- Table de routage : chaque nœud du réseau maintient une table de routage pour chaque application. Chaque application utilise à son tour sa propre méthode de décision multicritère. La table de routage maintient aussi la liste des routes découvertes par l'envoi des paquets d'apprentissage. Chaque route est indexée par son premier nœud, qui est à son tour un voisin direct du nœud courant.
- Paquet : la classe Paquet généralise les trois formes de paquets utilisés par le

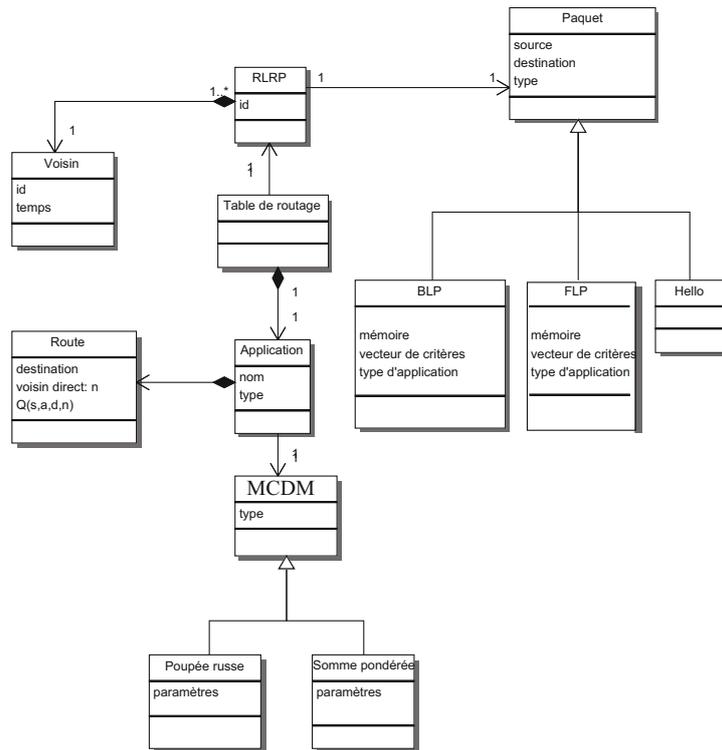


FIGURE 4.4 – Diagramme de classe du protocole RLRP

protocole RLRP, en l’occurrence les paquets FLP, BLP et “Hello”. Le paquet “Hello” est employé afin de découvrir la liste des voisins directs du nœud courant. La classe Paquet encapsule les attributs source, destination et le type de paquet. Les paquets d’apprentissage FLP et BLP encapsulent les propriétés importantes suivantes : type d’application, liste des nœuds intermédiaires déjà visités et le vecteur de critères.

Vue dynamique du protocole RLRP

Dans cette sous-section, nous présentons l’aspect dynamique du protocole RLRP notamment la mécanique qui permet de découvrir la topologie du réseau et le routage des paquets de données. En plus des différentes structures de données présentées ci-dessus, le protocole RLRP utilise des *timers* afin de réguler le mécanisme d’apprentissage et de découverte de voisins directs. La figure 4.5 illustre le fonctionnement de la classe principale du protocole RLRP. Cette dernière simule le comportement d’un nœud du réseau en fonction des événements qu’il reçoit de son environnement. D’abord,

4.4. Protocole de routage multicritère utilisant l'apprentissage par renforcement

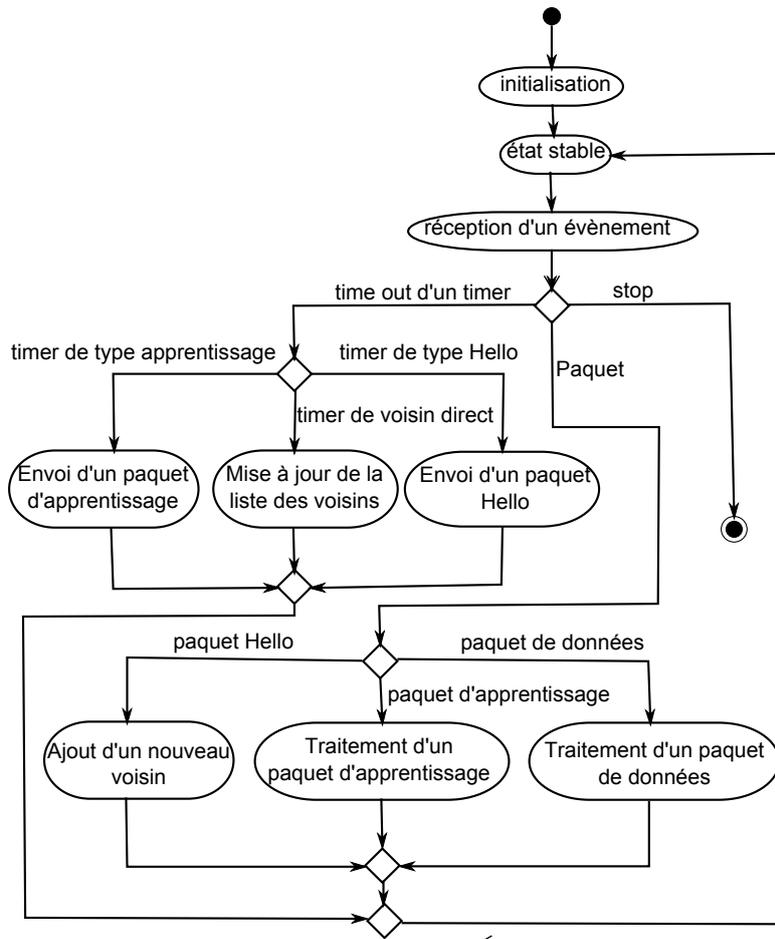


FIGURE 4.5 – Diagramme d'activité de la classe RLRP

le protocole commence par une étape d'initialisation où des valeurs sont attribuées à tous les paramètres notamment les valeurs des différents *timers*, à savoir le *timer* d'envoi de message hello et le *timer* de la mise à jour de la liste des voisins. Ensuite, le protocole passe à l'étape de stabilité, où le protocole est prêt à recevoir tous les événements de son environnement. Finalement, chaque réception d'un événement déclenche un processus réactif qui contribue à la mise à jour des différentes structures de données du protocole RLRP, en l'occurrence la table de routage et la liste des voisins directs.

- Envoi d'un paquet "Hello" : le nœud courant crée un paquet "Hello", initialise le champ source avec son identificateur, le champ destination par l'adresse de *broadcast* puis envoie le paquet à travers l'interface de la couche réseau.
- Envoi d'un paquet d'apprentissage : d'abord, le nœud courant sélectionne une

application et un voisin direct aléatoirement. Puis, il initialise les champs source, destination avec son identificateur et la destination courante et envoie le paquet à travers l'interface entre les couches réseau et liaison de données.

- Mise à jour de la liste des voisins : le nœud courant examine la liste des voisins directs. Les voisins qui n'ont pas envoyé de message "Hello" depuis une durée de temps déterminée (typiquement, trois fois l'intervalle de temps nécessaire à l'envoi d'un message "Hello") seront supprimés de la liste des voisins directs.
- Ajout d'un nouveau voisin : à la réception d'un paquet "hello", le nœud courant examine sa liste des voisins directs. Si le voisin n'existe pas déjà dans la liste, alors le nouveau voisin y est inséré. Dans le cas contraire, la date courante de la réception d'un message "Hello" pour le voisin courant est mise à jour.
- Traitement d'un paquet d'apprentissage : premièrement, le nœud courant examine le champ type de paquet d'apprentissage reçu, qui peut être soit un paquet FLP ou BLP. Dans le cas où le paquet reçu est un paquet FLP, le nœud courant examine le champ destination. Si le paquet lui est destiné, alors le nœud courant crée un nouveau paquet de type BLP et le renvoie à la source du paquet FLP. Dans le cas où le paquet reçu est un paquet BLP, le nœud courant l'oriente à un voisin direct, s'il existe un chemin vers la destination au niveau du nœud courant, ou l'oriente vers un voisin sélectionné aléatoirement dans le cas contraire.
- Traitement d'un paquet de données : à la réception d'un paquet de données, le nœud courant examine sa destination. Dans le cas où la destination finale du paquet de données est le nœud courant, le paquet est envoyé à la couche transport. Dans le cas contraire, le nœud courant examine sa table de routage pour savoir s'il existe un chemin vers la destination ou pas. Dans le cas où un chemin existe, le paquet de données est orienté vers le voisin direct indexant le chemin trouvé ; dans le cas contraire, le paquet de données est supprimé.

4.5 Expérimentations

Dans cette section, nous évaluons les performances de la méthode de la poupée russe combinée avec le protocole de routage RLRP dans un cas concret de réseau ad

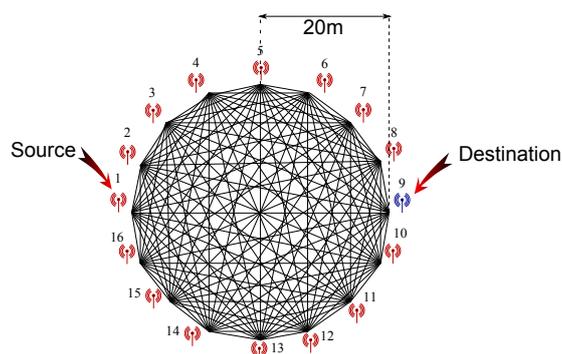


FIGURE 4.6 – Scénario de simulation

hoc. Pour ce faire, nous avons implanté le protocole RLRP, la méthode de la poupée russe et la méthode de la somme pondérée dans le simulateur ns-2 (*network simulator 2*) [BEF⁺00]. Le simulateur ns-2 est un simulateur à événements discrets qui offre une large gamme de fonctionnalités qui permettent de modéliser l'état réel de fonctionnement de plusieurs réseaux, notamment les réseaux ad hoc. Le simulateur ns-2 est très populaire dans le domaine de la simulation des réseaux; en témoignent le grand nombre d'articles scientifiques qui l'utilisent dans l'étape d'évaluation. Dans les sections suivantes, nous présentons les expérimentations réalisées sur trois types de scénarios :

- un problème “jouet” utilisant un ensemble d'alternatives non-dominées prédéfinies,
- un cas de routage dans lequel le modèle *shadowing* est utilisé comme modèle de propagation,
- un cas de routage où le taux de perte est une fonction quadratique de la distance entre l'émetteur et le récepteur.

4.5.1 Scénario

Le réseau est formé de 16 nœuds répartis uniformément sur un anneau de rayon 20 m (Voir la figure 4.6). Ce scénario peut être vu comme un ensemble de points d'accès qui forment un réseau *mesh*. Les nœuds 1 et 9 forment un couple source et destination. Tous les nœuds du réseau, excepté le nœud destination, envoient des pa-

quets d'apprentissage pour explorer la topologie et découvrir des routes multicritère vers la destination. En plus des paquets d'apprentissage, le nœud 1 envoie des paquets CBR (*Constant Bit Rate*) simulant une application VoIP avec un débit de $16Ko/s$. Au niveau de la couche MAC (*Medium Access Control*), nous utilisons le standard IEEE 802.11 avec un débit de transmission de l'ordre de 2 Mbps. Dans ces simulations, nous observons deux critères, en l'occurrence le délai de bout en bout (D), et le taux de perte de bout en bout (L). Le délai de bout en bout d'un paquet donné est la somme des délais individuels (d_i) engendrés par chaque nœud intermédiaire i qui compose le chemin entre la source et la destination (voir la figure 4.7). Les délais individuels d_i sont engendrés par les congestions qui apparaissent au niveau des nœuds intermédiaires pendant la communication. Une congestion se produit quand un nœud donné est surpassé par la réception, transmission et relai des différents paquets appartenant à de multiples applications qui transitent dans le réseau. Le délai de bout en bout est donné comme suit :

$$D = \sum_{i=1}^{n-1} d_i \quad (4.4)$$

où $d_i, i \in [1, n - 1]$ est un nombre réel positif généré par une loi logistique de moyenne $100ms$ et d'écart type $80ms$ [AEES08].

D'autre part, le critère du taux de perte de bout en bout est fonction des taux de pertes l_i des liens composant le chemin emprunté par les paquets allant de la source vers la destination. l_i est provoqué par l'atténuation du signal en fonction de la distance entre les nœuds successifs i et $i + 1$, ainsi que les interférences générées par le voisinage du nœud récepteur. Le taux de perte de bout en bout L est donné comme suit :

$$L = 1 - \prod_{i=1}^{n-1} (1 - l_i) \quad (4.5)$$

l_i sont générés de deux manières, en fonction du scénario testé : premièrement, en utilisant le modèle de propagation *Shadowing* [SAZ07] avec le paramètre “*path loss exponent*” égal à 3 et un écart type égal à 4. Deuxièmement, $l_i = \frac{d^2}{40000}$ pour le scénario incluant un modèle de perte quadratique, avec d exprimée en mètres.

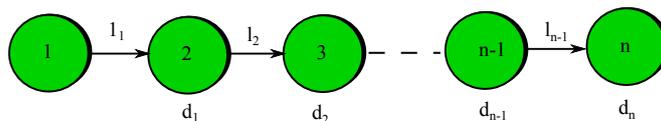


FIGURE 4.7 – Illustration d’une route entre la source et la destination

4.5.2 Structure de la poupée russe

La structure de la poupée russe qui représente les exigences de l’application VoIP est montrée dans la figure 4.8. Les formes des boîtes ont été apprises au moyen de la méthode de classification automatique introduite dans le chapitre 3. L’apprentissage a été effectué sur une expérience MOS simulée contenant 1000 couples (\mathbf{c}_i, q_i) , tels que \mathbf{c}_i sont des vecteurs de critères mesurés au niveau de la destination et q_i est une valeur de qualité subjective. Ces couples (\mathbf{c}_i, q_i) de la base de données MOS ont été représentés sur la figure 4.8.

Les coefficients des différents hyperplans représentant les boîtes sont donnés dans le tableau 4.1. Nous rappelons qu’une boîte est définie par l’intersection de demi-espaces dont les bordures sont des hyperplans. Un hyperplan est défini, dans l’espace des critères, par son équation $a_1x_1 + a_2x_2 = 1$. Les coefficients a_1 et a_2 sont la première et la deuxième colonne des matrices ci-dessous, respectivement :

$$\begin{array}{ccc} \begin{pmatrix} 0,01 & 0 \\ 0 & 2 \\ 1 & 0,0077 \end{pmatrix} & \begin{pmatrix} 0,004 & 0 \\ 0 & 0,52 \\ 0,0033 & 0,33 \end{pmatrix} & \begin{pmatrix} 0,0025 & 0 \\ 0 & 0,25 \\ 0,0018 & 0,18 \end{pmatrix} \\ \text{boîte 0} & \text{boîte 1} & \text{boîte 2} \end{array}$$

TABLE 4.1 – Coefficients des hyperplans composant les différentes boîtes de la poupée russe

4.5.3 Méthodologie de comparaison

Dans ces expérimentations, nous voulons comparer les qualités des routages réalisés par la méthode de la poupée russe et par la méthode de la somme pondérée. Pour ce faire, le résultat d’une expérience de type MOS peut être considéré comme un arbitre qui décide quelle méthode est la meilleure. Cependant, les résultats obtenus à partir

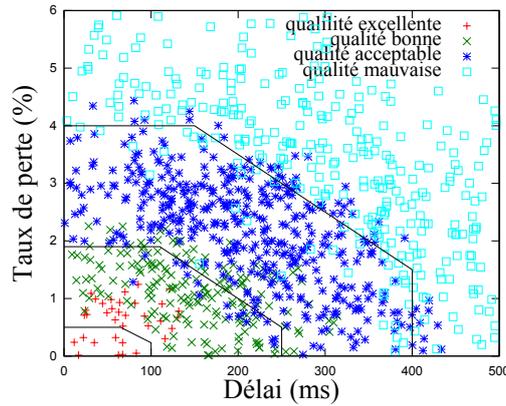


FIGURE 4.8 – Structure de la poupée russe pour l’application VoIP

d’un MOS sont aléatoires : pour un même vecteur de critères mesuré à destination, plusieurs utilisateurs peuvent donner différentes opinions subjectives. Par conséquent, la comparaison doit se faire sur la base d’une distribution de probabilité empirique, qui représente la qualité du routage. De ce fait, pour utiliser la notion de fonction de répartition empirique, les niveaux de qualité : excellent, bon, acceptable et mauvais sont représentés par une variable aléatoire entière prenant les valeurs : 3, 2, 1 et 0, respectivement. Deux distributions de probabilité $P(q_i \setminus \mathbf{c}_i)$ et $P(q_j \setminus \mathbf{c}_j)$ estimées à partir du résultat d’une expérience MOS peuvent être comparées au moyen du test statistique de Kolmogorov-Smirnov [Con80] pour déterminer avec un risque d’erreur donné si les échantillons de qualité $\{q_i\}$ and $\{q_j\}$ proviennent de la même distribution de probabilité. Dans ce cas, nous disons que les deux échantillons de qualité sont similaires. Dans le cas contraire, les distributions empiriques sont comparées en utilisant la dominance stochastique du premier ordre [HR69], définie comme suit : soient A et B deux distributions de probabilité, et F_A et F_B leurs fonctions de répartition respectives.

$$A \text{ “domine stochastiquement au premier ordre” } B \Leftrightarrow \begin{cases} \forall y, & F_B(y) \geq F_A(y) \\ \exists y, & F_B(y) > F_A(y) \end{cases}$$

Ainsi, lorsque A domine B , la fréquence des qualités inférieures à y associée à la distribution B est supérieure ou égale à la fréquence des qualités inférieures à y associée

à la distribution A , et ce pour toutes les valeurs de y .

Pour avoir une bonne estimation de $\{q_i\}$ il faudrait un grand nombre d'échantillons de qualité subjective q_i associés aux vecteurs de critères \mathbf{c}_i . Cependant, dans le cas général, la probabilité de tirer plusieurs fois le même vecteur de critères \mathbf{c}_i durant une expérience MOS est égale à zéro, les coordonnées du vecteur de critères \mathbf{c}_i étant définies dans des intervalles de \mathbf{R} . Pour remédier à cette difficulté, nous proposons d'estimer la densité de probabilité $P\{q_i \setminus \mathbf{c}_i\}$ au moyen de la méthode des k plus proches voisins (k-NN, *k-Nearest Neighbors*) [CH67] (Figure 4.9). La méthode k-NN exige la définition d'une distance utilisée dans l'espace des critères, ainsi que le choix de la valeur du paramètre k . Une distance euclidienne pondérée est utilisée dans ce contexte, elle est définie comme suit :

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n \left(\frac{a_i - b_i}{u_i} \right)^2}$$

où u_i est la borne supérieure de la coordonnée i des vecteurs de critères contenus dans le MOS. La taille d'un échantillon $k = 128$ pour une base de données MOS contenant 10000 entrées a été déterminée au moyen de plusieurs expériences préliminaires, qui ont permis de trouver un bon compromis entre le biais et la variance des estimations.

Une simulation consiste à transmettre des paquets de données du nœud source au nœud destination pendant 100 secondes. Pour chaque paquet de données arrivant au niveau du nœud destination, un vecteur de critères \mathbf{c} (délai et taux de perte moyens) est calculé.

Un histogramme de niveaux de qualité (Figure 4.10) est alors obtenu en appliquant la méthode d'estimation k-NN sur le MOS avec le vecteur \mathbf{c} obtenu durant une simulation. Une comparaison de la qualité de routage nécessite une paire de simulations pour un vecteur de poids donné \mathbf{w} associé à la méthode de la somme pondérée. L'une des simulations utilise la méthode de la poupée russe pour obtenir l'histogramme de qualité noté R . L'autre utilise la méthode de la somme pondérée dont l'histogramme de qualité associé est noté W . Les paramètres d_i sont choisis au hasard et sont identiques pour chaque paire de simulations.

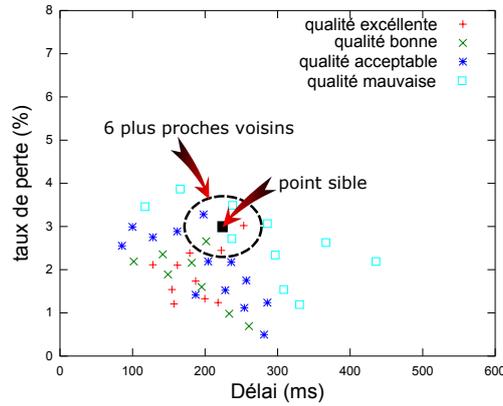


FIGURE 4.9 – Méthode des k plus proches voisins

Pour un vecteur de poids \mathbf{w} , N comparaisons de paires d’histogrammes sont effectuées. Les résultats sont les suivants :

- N_s : le nombre de comparaisons où les échantillons R et W sont similaires,
- N_r : le nombre de comparaisons où R domine W ,
- N_w : le nombre de comparaisons où W domine R ,
- N_n : le nombre de comparaisons où ni R ni W ne domine l’autre histogramme,

avec $N_s + N_r + N_w + N_n = N$. Nous définissons les fréquences relatives $f_s = N_s/N$, $f_r = N_r/N$, $f_w = N_w/N$ et $f_n = N_n/N$.

4.5.4 Résultats et discussion

Dans cette section, nous présentons trois expérimentations. La première a pour but de comparer les résultats obtenus par la méthode de la poupée russe et par celle de la somme pondérée sur un problème “jouet”, en utilisant le MOS construit pour les problèmes de routage. Les deux autres expérimentations donnent des résultats de comparaison pour le problème de routage décrit précédemment.

Problème jouet

Dans cette expérimentation, le système de coordonnées de l’espace des critères est ramené entre 0 et 6 sur chaque axe des critères en divisant les délais par $100ms$ et en multipliant le taux de perte fois 100 pour tous les vecteurs de critère du MOS. Cette

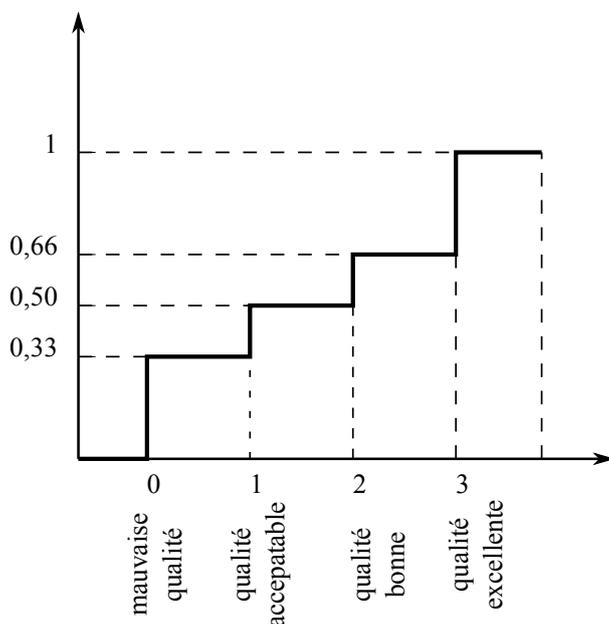


FIGURE 4.10 – Exemple d'une fonction de répartition d'un histogramme de qualité

mise à l'échelle est effectuée pour ramener les deux critères au même ordre de grandeur et faciliter l'analyse. Les méthodes de la poupée russe et de la somme pondérée sont testées au moyen d'un ensemble de solutions non-dominées situées sur un quart de cercle dont le rayon est égale à 1 et les coordonnées du centre sont des nombres réels choisis aléatoirement dans l'intervalle $[0, 6]$ (Figure 4.11). Les fréquences relatives f_s , f_r , f_w et f_n sont déterminées via une comparaison de 600 histogrammes de qualité générés pour chaque poids w_1 associé au critère délai et $w_2 = 1 - w_1$ associé au critère taux de perte. Les valeurs de f_s , f_r , f_w et f_n vs. w_1 sont présentées dans la figure 4.12. Cette dernière montre que les solutions trouvées par la méthode de la poupée russe dominant ou sont similaires à celles trouvées par la méthode de la somme pondérée pour presque toutes les comparaisons. Rarement, les histogrammes de qualité obtenus par la méthode de la somme pondérée dominant ceux fournis par la méthode de la poupée russe : au plus 3 cas sur 600 pour les valeurs de poids w_1 appartenant à l'intervalle $[0, 6; 0, 84]$. Nous avons vérifié que ces cas sont sporadiques. Ils sont probablement dus à des erreurs de première espèce commises par les tests de Kolmogorov-Smirnov. La figure montre également que les histogrammes de qualité pour cette expérimentation sont toujours comparables : $f_n = 0$ pour toutes les valeurs de poids.

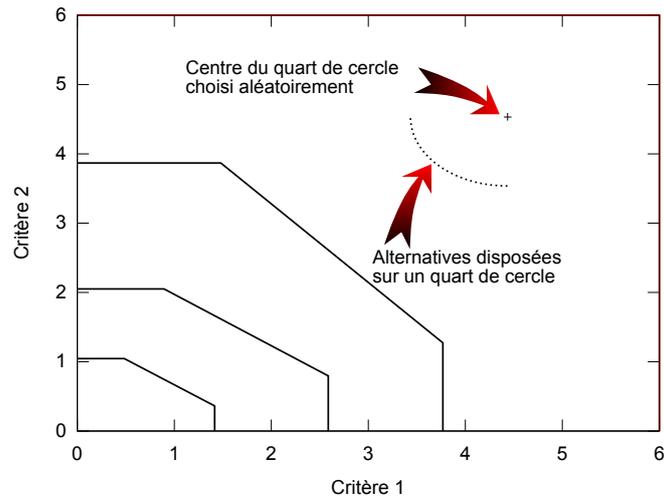


FIGURE 4.11 – Illustration de l'ensemble de solutions non-dominées utilisées dans le problème jouet

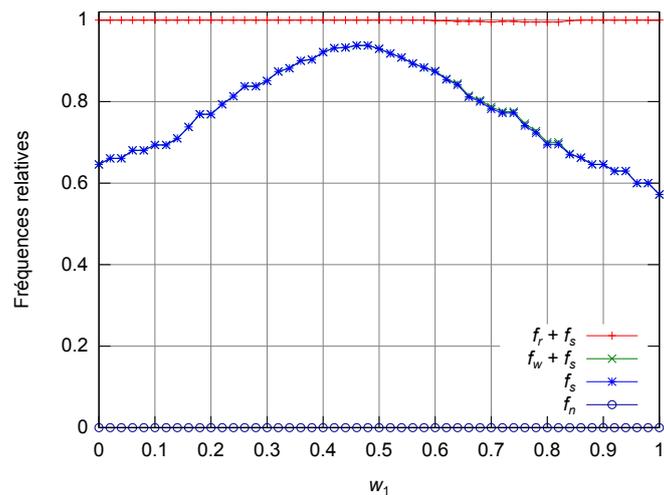


FIGURE 4.12 – Fréquences de dominance relatives dans le cadre du problème jouet : méthode de la poupée russe vs. méthode de la somme pondérée

Problème de routage multicritère avec le modèle *shadowing*

Dans cette section, nous présentons les résultats de simulation relatifs au scénario dans lequel les taux de perte l_i sont engendrés par le modèle *shadowing* (voir la section 4.5.1). La figure 4.14 montre les fréquences relatives $f_s + f_r$, $f_s + f_w$, f_s et f_n vs. w_1 obtenues à partir de 100 paires de simulations au niveau de la table de routage de la source. Nous rappelons que la table de routage de la source s contient les valeurs $\mathbf{Q}(a, d, s, n)$ qui représentent les espérances des vecteurs de critères à la destination

d associés à chaque voisin direct n et application a . Dans ce contexte, le vecteur de critères considéré pour le calcul des histogrammes de qualité R et W est celui donné par l'équation 4.3.

La figure 4.14 montre que les solutions trouvées par la méthode de la poupée russe dominant ou sont similaires à celles trouvées par la méthode de la somme pondérée pour toutes les valeurs de poids. Pour une seule paire de simulation parmi 100, la méthode de la somme pondérée domine la méthode de la poupée russe quand le poids w_1 est égal à 0,75. Pour $w_1 = 0,5$ toutes les paires de simulations donnent des résultats similaires. Cela peut s'expliquer par le fait que la plupart des vecteurs optimaux obtenus avec la méthode de la poupée russe sont ceux qui sont les plus proches des hyperplans obliques, comme le montre la Figure 4.13. Dans ce cas, la méthode de la poupée russe se comporte comme une méthode de somme pondérée avec des poids égaux approximativement à $(0,5; 0,5)$ (voir chapitre 3). Pour les valeurs de poids différentes de $(0,5; 0,5)$, la fréquence des solutions similaires pour les deux méthodes décroît considérablement, dès que la valeur du poids du critère délai (w_1) est supérieure à 0,6 ou inférieure à 0,45.

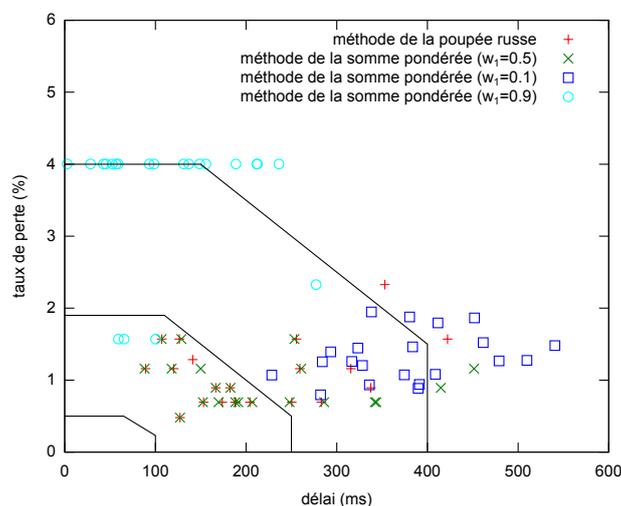


FIGURE 4.13 – Solutions trouvées par les deux méthodes : poupée russe et somme pondérée avec différents poids

Il est intéressant d'évaluer la qualité de routage à la destination, au lieu d'une prévision de qualité donnée par la table de routage du nœud source. La figure 4.15

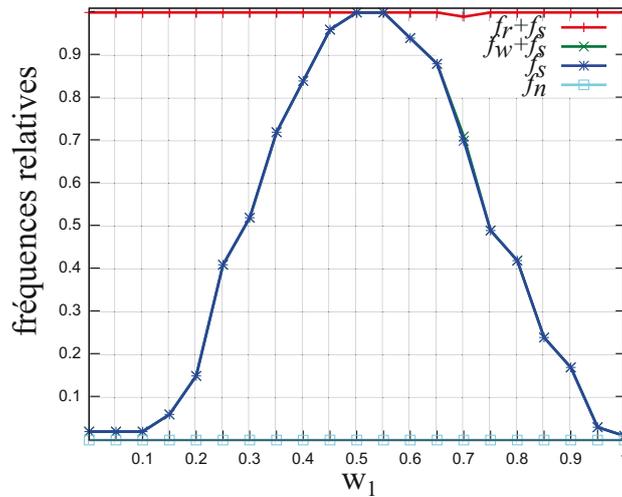


FIGURE 4.14 – Méthode de la poupée russe vs. méthode de la somme pondérée (au niveau de la source)

donne les fréquences relatives calculées à partir des vecteurs de critères évalués à la destination. Les figures 4.14 et 4.15 diffèrent parce que les valeurs des vecteurs de critères au niveau de la destination sont soumises à des variations aléatoires incessantes du délai et du taux de perte, qui sont imprévisibles. Malgré les variations stochastiques des critères, la méthode de la poupée russe domine nettement plus fréquemment la méthode de la somme pondérée pour les valeurs de w_1 en dehors de l'intervalle $[0, 4; 0.6]$.

Problème de routage multicritère avec le modèle quadratique

Dans cette section, nous présentons les résultats de simulation relatifs au scénario dans lequel les taux de perte l_i sont générés avec une fonction quadratique (voir la section 4.5.1). La figure 4.16 renforce les résultats obtenus dans l'expérience précédente, car la grande majorité des solutions trouvées par la méthode de la poupée russe dominant ou sont similaires à celles trouvées par la méthode de la somme pondérée, et ce quelle que soit la configuration des poids. D'autre part, elle montre que la méthode de la poupée russe s'adapte aux variations de l'environnement, car elle trouve à chaque fois de meilleures solutions, sans avoir recours à une éventuelle reconfiguration de sa structure ou de ses paramètres. En effet, nous remarquons, à travers cette expérience, que les meilleurs routages trouvés par la méthode de la somme pon-

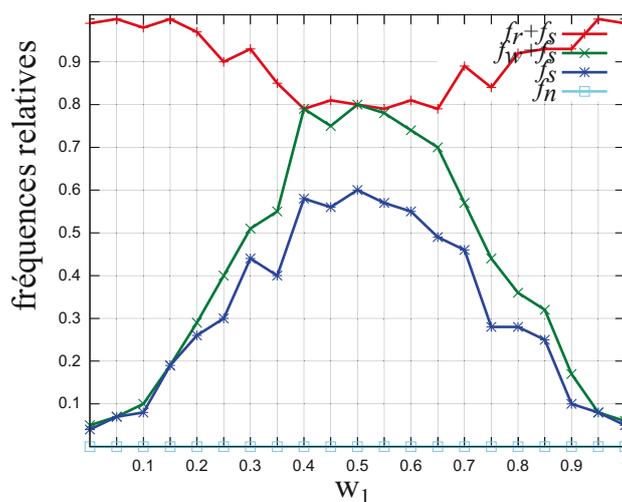


FIGURE 4.15 – Méthode de la poupée russe vs. méthode de la somme pondérée (au niveau de la destination)

dérée correspondent au cas où w_1 est dans l'intervalle $[0, 40; 0, 45]$ contrairement à la configuration optimale relative à l'expérience précédente, où w_1 est dans l'intervalle $[0, 5; 0, 55]$. Ainsi, si nous prenons une valeur de poids $w_1 = 0, 52$ qui est le poids moyen optimal dans l'expérience relative au modèle *shadowing*, nous constatons que ce poids est notablement sous-optimal pour la somme pondérée dans l'expérience relative au modèle quadratique. En effet, la méthode de la poupée russe domine dans 30% des cas la somme pondérée avec $w_1 = 0, 52$ dans l'expérience relative au modèle quadratique, tandis que ce taux de domination est de l'ordre de 10% pour le poids optimal. Cela montre que la méthode de la somme pondérée ne s'adapte pas à différentes conditions de fonctionnement du réseau. Les rares dominations de la méthode de la somme pondérée constatées figure 4.16 dans l'intervalle $[0, 45; 0, 55]$ pourraient être engendrées par des erreurs d'approximation de la base de donnée MOS pour le système de boîtes. La réponse précise à cette question nécessite des investigations supplémentaires.

4.6 Conclusion

Les réseaux ad hoc sans fil sont des réseaux complexes, qui sont capables de s'auto-organiser d'une façon autonome, afin de concevoir une topologie efficace pour la communication. La topologie est exploitée par un protocole de routage dédié qui établit

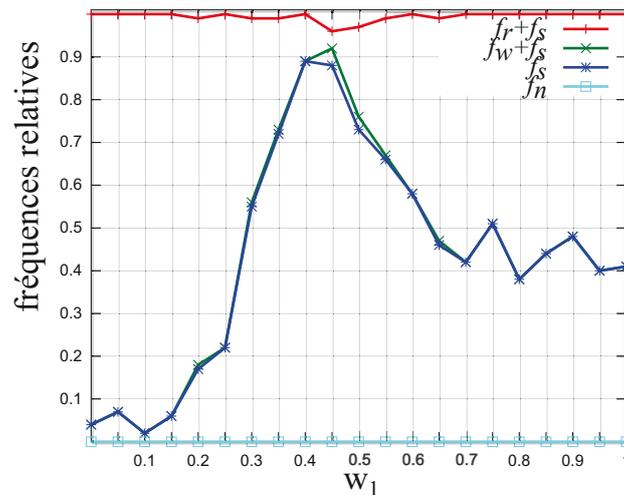


FIGURE 4.16 – Méthode de la poupée russe vs. méthode de la somme pondérée

des routes entre les différents nœuds, afin de faciliter la communication et de maximiser la qualité de service des applications qui transitent dans le réseau. Cependant, le protocole de routage doit alors tenir compte de la complexité de l’environnement, notamment l’aspect multicritère des exigences en QoS des différentes applications.

Dans ce chapitre, nous avons présenté un protocole de routage fondé sur l’apprentissage par renforcement utilisé conjointement avec une méthode de décision multicritère. Ce protocole facilite l’exploration de la topologie et la découverte de multitudes de routes multicritères entre la source et la destination.

Dans ce contexte, nous avons comparé la méthode de la poupée russe avec la méthode de la somme pondérée qui est très utilisée dans la littérature. Nous avons vu, à travers trois expérimentations, que la méthode de la poupée russe domine ou est similaire à la somme pondérée, dans presque 100% des simulations et ce, quelle que soit la configuration des poids utilisés par la méthode de la somme pondérée. De plus, l’utilisation de la somme pondérée, pour être performante, exige une recherche des poids optimaux, selon les variations de l’environnement. En revanche, les décisions prises avec la méthode de la poupée russe s’avèrent meilleures, quel que soit le contexte de fonctionnement. Ainsi, les résultats des comparaisons ont montré que la méthode de la poupée russe répond mieux que la méthode de la somme pondérée à l’aspect dynamique du routage dans un contexte autonome.