République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/MIRA de Bejaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de Fin de cycle

En vue de l'obtention du diplôme de Master Professionnel en Informatique

Option : Administration et Sécurité des Réseaux

Thème

Simulation d'un protocole de surveillance des interfaces d'un routeur

---

Réalisé par :

Mr RAHMOUNE Amer.

Devant le jury composé de :

Président :	Mr	A. SIDER
Examinateur :	Mr	S. AISSANI.
Rapporteur :	Mr	D. TOUAZI

Promotion 2014/2015

Remerciements

*Je tiens tout d'abord à remercier le bon **DIEU** qui m'a donné la santé et le courage d'accomplir ce modeste travail*

Mes remerciements sont adressés également à mes chers parents, pour leur amour, leurs sacrifices et leur patience.

*Je tiens à remercier vivement **Mr TOUAZI Djoudi**, pour m'avoir honoré par son encadrement, pour sa disponibilité, ses orientations, ses precieux conseils et ses encouragements qui m'ont permis de mener à bien ce travail.*

*Je tiens à exprimer ma gratitude aux **membres de jury** pour avoir accepté de juger ce travail.*

Enfin, à tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce modeste travail trouvent ici l'expression de ma sincère gratitude et mes remerciements les plus sincères.

Dédicaces

Je dédie ce modeste travail :

À mes très chèrs parents ;

À mes très chèrs frères Nacer-eddine, Lmine, Houssem et Tarek ;

À mes très chères sœurs Zoulikha, Djahida, Dounia, Kanza et leurs maris ;

Àu beaux enfants de mon frère et au beaux enfants de mes sœurs ;

À toute la famille sans exception ;

À tous mes amis ;

Amer

LISTE DES ABRÉVIATIONS

AODV **A**d Hoc **O**n **D**emand **V**ector

B.A.T.M.A.N **B**etter de **A**pproach **T**o **M**obile **A**d hoc **N**etwork

DSDV **D**estination **S**equenced **D**istance **V**ector

DSR **D**ynamic **S**ource **R**outing

IP **I**nternet **P**rotocol

MPR **M**ulti **P**oint **R**elays

OGM **O**riginator **A**essage

OLSR **O**ptimized **L**ink **S**tate **R**outing

PSCLIR **P**rotocol de **S**urveillance de **C**onnexion des **L**iens d'**I**nterfaces d'un **R**outeur

QOS **Q**ualité of **S**ervice

RCSF **R**éseaux **C**apteurs **S**ans **F**ils

RF **R**adio **F**requency

SB **S**tations de **B**ase

TCP	**T**ransmission **C**ontrol **P**rotocol
UM	**U**nité **M**obile
UWB	**U**ltra **W**ide**B**and
WSNs	**W**ireless **S**ensor **N**etworks
ZRP	**Z**one **R**outing**P**rotocol

TABLE DES MATIÈRES

LISTE DES TABLEAUX

TABLE DES FIGURES

INTRODUCTION GÉNÉRALE

L'évolution récente de la société s'appuie sur des techniques de plus en plus tournées vers la communication, l'image et la mobilité. Le téléphone portable et Internet sont les vecteurs principaux de cette révolution technologique. A côté de ces techniques plutôt liées aux loisirs se développent également des dispositifs pour améliorer notre connaissance du monde extérieur. Les informations recueillies dans la nature par exemple, vont être récupérées pour être intégrées au processus de décision.

Le besoin d'échange rapide d'informations et le développement des communications ont abouti à la création d'Internet qui rend accessible au monde entier une grande quantité de données et de services. Internet suscite une passion croissante, tant dans le domaine de la recherche, de l'éducation que celui des affaires. Ainsi le nombre de personnes qui accèdent à Internet pour leur travail, leurs études ou leurs loisirs augmente sans cesse, de même que les services offerts sur ce réseau (messagerie électronique, moteur de recherche, e-commerce, e-learning, etc.). Dans un avenir proche, Internet va enrichir ses bases de données avec des informations en temps réel directement issues de phénomènes naturels.

L'objectif de ce mémoire est de traiter le problème du routage dans les réseaux informatiques, surtout ceux dont la taille est importante. Le souci principal est de prolonger la vie du système. Pour cela, nous avons proposé la simulation d'un algorithme de routage qui surveille le comportement des liens aux interfaces d'un routeur et calcule leurs probabilités de connexions à des instants donnés.

Dans ce rapport, notre mémoire est organisé comme suit :

- Le premier chapitre présente des généralités sur les réseaux informatiques et les types des réseaux existants.
- Le deuxième chapitre présente une étude comparative entre les protocoles de routage existants, leurs avantages et leurs inconvénients.
- Le troisième chapitre présente le principe de fonctionnement de la solution étudiée

que nous avons proposé (le protocole PSCLIR).

- Le quatrième chapitre présent la simulation pratique du fonctionnement du protocole PSCLIR et les outils utilisés pour le développement.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX SANS FILS

1.1 Introduction

Le développement technologique au cours de ces dernières années à révolutionner un certaine nombre de domaine, notamment celui des réseaux informatique et plus spécialement les réseaux sans fil. Les performances ne cessent d'augmenter et les prix de chuter. Cette avance les a rendus abordable par le grand public. Ce marché de l'équipement sans fil est actuellement en plein essor.

Cette révolution des réseaux informatique devrait se poursuivre au vue des couts des investissements dans le câblage de plus en plus élevés.Le sans fil à toucher beaucoup de domaine mis à part les réseaux informatique, par exemple la téléphonie mobile et les réseaux de capteur sans fil.

Dans ce chapitre, nous donnons un bref aperçu sur les réseaux sans fils (Réseaux ad hoc, Réseaux de capteurs). Nous définissons, par la suite, les réseaux ad hoc et réseaux de capteurs, leurs caractéristiques, leurs contraintes et les différents domaines d'application de ces derniers, Suite à cela, nous exposerons une brève description des particularités des environnements qui sont à l'origine de la naissance des réseaux tolérant aux délais (DTNs). Pour enfin terminer par la définition de ces derniers et une description détaillée des éléments les caractérisant.

1.2 Réseaux mobiles

Un environnement mobile est un système composé de sites mobiles et qui permet à ses utilisateurs d'accéder à l'information indépendamment de leurs positions géographiques.

Les réseaux mobiles ou sans fil, peuvent être classés en deux classes : les réseaux avec infrastructure et les réseaux sans infrastructure.

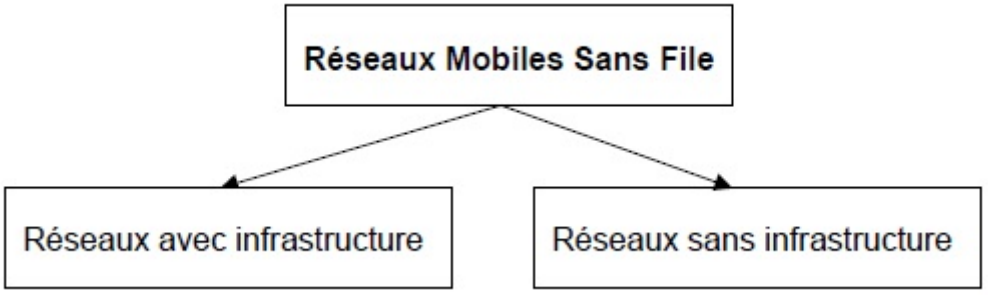

FIGURE 1.1 – Décomposition des réseaux mobiles.

1.2.1 Les réseaux avec infrastructure (cellulaire)

Le modèle de système intégrant des sites mobiles et qui a tendance à se généraliser est composé de deux ensembles d'entités distinctes :

- Les " sites fixes " d'un réseau de communication filaire classique (wired network).
- Les "sites mobiles" (Wireless network).
  Certains sites fixes, appelés stations de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites mobiles ou unité mobile (UM) localisés dans une zone géographique limitée, appelée cellule comme le montre la figure (1.2).
- A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire.
- Une unité mobile ne peut être à un instant donné directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée [2].

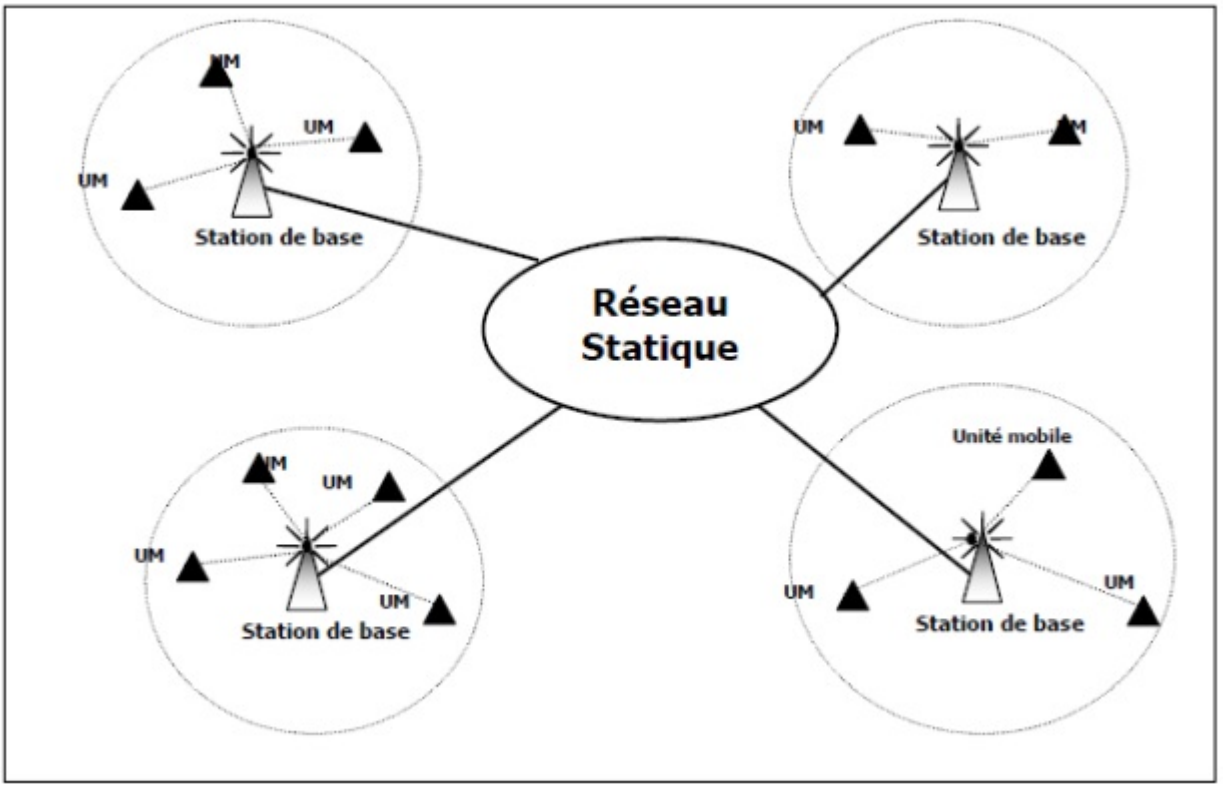

FIGURE 1.2 – Le modèle de réseaux mobile avec infrastructure.

1.2.2 Les réseaux sans infrastructure

Le modèle du réseau mobile sans infrastructure préexistante ne comporte pas l'entité " site fixe ". Tous les sites du réseau sont mobiles et communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil (Figure 1.3).

L'absence de l'infrastructure ou d'un réseau filaire composé de stations de base, oblige les unités mobiles (UM) à se comporter comme des routeurs qui participent à la découverte et à la maintenance des chemins pour les autres hôtes du réseau. Ce type de réseau est appelé : Ad Hoc

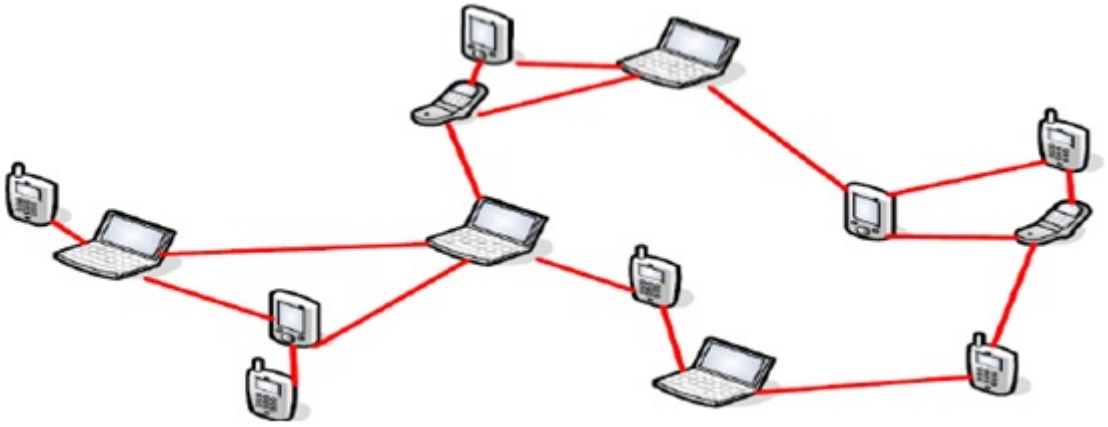

FIGURE 1.3 – Le modèle des réseaux mobiles sans infrastructure.

1.3 Réseaux ad hoc

Un réseau mobile Ad Hoc appelé généralement MANET, consiste en une grande population relativement dense d'unités mobiles qui se déplacent dans un territoire quelconque. Le seul moyen de communication est l'utilisation des "ondes radio" qui se propagent entre les différents nœuds mobiles sans l'aide d'une infrastructure préexistante ou administration centralisée [3]. Dans un réseau Ad Hoc, un nœud peut communiquer directement (mode point-à-point) avec n'importe quel nœud s'il est situé dans sa zone de transmission, tandis que la communication avec un nœud situé en dehors de sa zone de transmission s'effectue via plusieurs nœuds intermédiaires (mode multi-sauts).

1.3.1 Caractéristiques des réseaux Ad Hoc

Les réseaux Ad Hoc sont caractérisés principalement par :

- **Topologie dynamique :** la topologie des réseaux ad hoc change rapidement et aléatoirement, ceci est causé par la mobilité arbitraire des nœuds du réseau. Le changement de la topologie change les routes entre les nœuds et provoque la perte de paquets [4].
- **Bande passante limitée :** une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé (ondes radio). Ce partage fait que la bande passante réservée à un hôte soit modeste [5].
- **Contraintes d'énergie :** les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries ou les autres sources consommables. Le paramètre

d'énergie doit être pris en considération dans tout contrôle fait par le système.

- **Sécurité physique limitée :** les réseaux mobiles Ad Hoc sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.
- **Erreur de transmission :** les erreurs de transmission radio sont plus fréquentes que dans les réseaux filaires.
- **Interférences :** les liens radios ne sont pas isolés, deux transmissions simultanées sur une même fréquence utilisant des fréquences proches peuvent interférer entre eux.
- **Absence d'infrastructure :** les réseaux Ad Hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructures préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue [5].
- **Nœuds cachés :** ce phénomène est très particulier à l'environnement sans fil. Les nœuds ne s'entendent pas à cause d'un obstacle qui empêche la propagation des ondes. Les mécanismes d'accès au canal vont permettre alors à ces nœuds de commencer leurs émissions simultanément. Ce qui provoque des collisions au niveau du nœud.
- **Qualité de service :** de nombreuses applications ont besoin de certaines garanties relatives par exemple au débit, au délai ou encore à la gigue. Dans ces réseaux Ad Hoc, ces garanties sont très difficiles à obtenir. Ceci est dû à la nature du canal radio d'une part (interférences et taux d'erreur élevés) et au fait que des "liens" entre des mobiles peuvent avoir à se partager les ressources. [6]

1.3.2 Domaine d'applications des Réseaux ad hoc

Les réseaux ad hoc sont rapides et faciles à déployer, ils sont particulièrement intéressants pour les applications militaires ou l'installation d'infrastructure fixe est souvent impossible, ils peuvent être aussi utilisées dans :

- **Les opérations de recherche et de secours :** En cas de tremblement de terre, de feux ou d'inondation, dans le but de remplacer rapidement l'infrastructure détruite.
- **L'informatique embarquée :** Dans des véhicules communiquant par exemple.
- **Les entreprises :** Dans le cadre d'une réunion ou d'une conférence.
- **Les gares et aéroports :** Pour la communication et la collaboration entre les membres du personnel.

1.4 Les réseaux de capteurs

On vient de parler des environnements mobiles et de réseaux avec et sans infrastructure, et on verra dans ce qui suit les réseaux de capteurs.

1.4.1 Définition d'un capteur

Un capteur est le dispositif qui transforme une grandeur physique observée (température, position, humidité, etc) en une grandeur utilisable (intensité électrique, position d'un flotteur) [7]. Pour cela, il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre.

FIGURE 1.4 – Quelques exemples de capteurs.

1.4.2 Définition d'un réseau de capteur

Un réseau de capteurs sans fil (RCSF) est un type particulier de réseaux mobiles ad hoc MANET. Il est composé d'un ensemble de dispositifs très petits, nommés nœuds capteurs, variant de quelques dizaines d'éléments à plusieurs milliers. Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil [8].

1.4.3 Architecture d'un capteur

Pour bien comprendre le fonctionnement et les autres concepts des réseaux de capteurs, on s'intéresse à leur architecture et à celle de leurs composants.

- **Architecture matérielle :** Deux entités sont fondamentales dans le fonctionnement d'un capteur : l'unité de traitement qui est le cœur physique et l'unité de communication [9] [10].

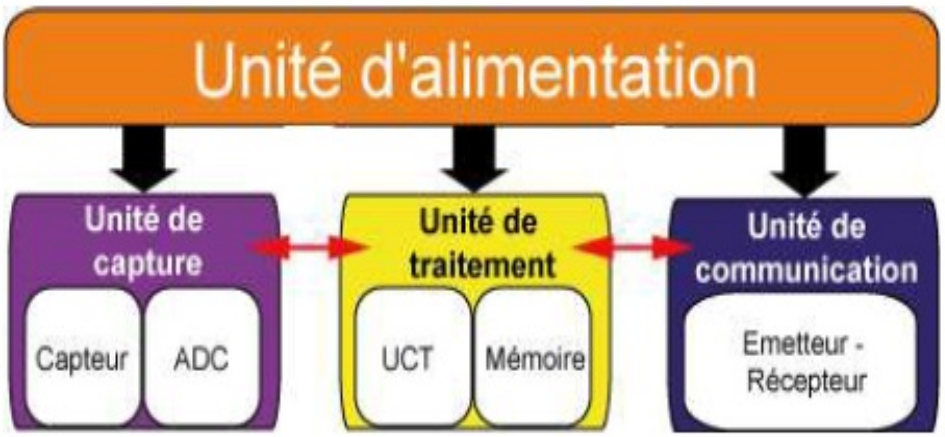

FIGURE 1.5 – Architecture d'un capteur sans fil

  - **Unité de traitement :** c'est l'unité principale du capteur. Elle est généralement représentée par un processeur couplé à une mémoire vive. Son rôle est de contrôler le bon fonctionnement des autres unités.
  - **Unité d'acquisition :** Composée d'un capteur qui obtient des mesures sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui convertit l'information relevée et la transmet à l'unité de traitement. [11]
  - **Unité de communication :** elle a pour fonction de transmettre et recevoir l'information. Elle est équipée d'un couple émetteur/récepteur pour communiquer au sein du réseau. Il existe cependant d'autres possibilités de transmission (optique, infrarouge, etc.).
  - **Unité d'alimentation :** c'est un élément primordial de l'architecture du capteur, c'est elle qui fournit en énergie toutes les autres unités. Elle correspond le plus souvent à une batterie ou une pile alimentant le capteur, dont les ressources limitées en font une problématique propre à ce type de réseau puisque ces derniers sont généralement déployés dans des zones non accessibles [12].
- **Architecture Logicielle :** La contrainte énergétique des capteurs exige l'utilisation de systèmes d'exploitation légers tels que TinyOS ou Contiki [13]. Cependant, TinyOS reste toujours le plus utilisé et le plus populaire dans le domaine des RCSF.

Il est libre et est utilisé par une large communauté de scientifiques dans des Simulations pour le développement et le test des algorithmes et protocoles réseau.

1.4.4 Architecture d'un réseau de capteur

Le processus d'acheminement de l'information des capteurs à la station de base peut prendre quatre formes. Dans les architectures à plat, les capteurs peuvent communiquer directement avec la station de base en utilisant une forte puissance (figure 1.6 (a)), ou via un mode multi-sauts avec des puissances très faibles (figure 1.6 (b)), alors que dans les architectures hiérarchisées, le nœud représentant le cluster, appelé cluster-head, transmet directement les données à la station de base (figure 1.6 (c)), ou via un mode multi-saut entre les cluster-heads (figure 1.6 (d)) [20].

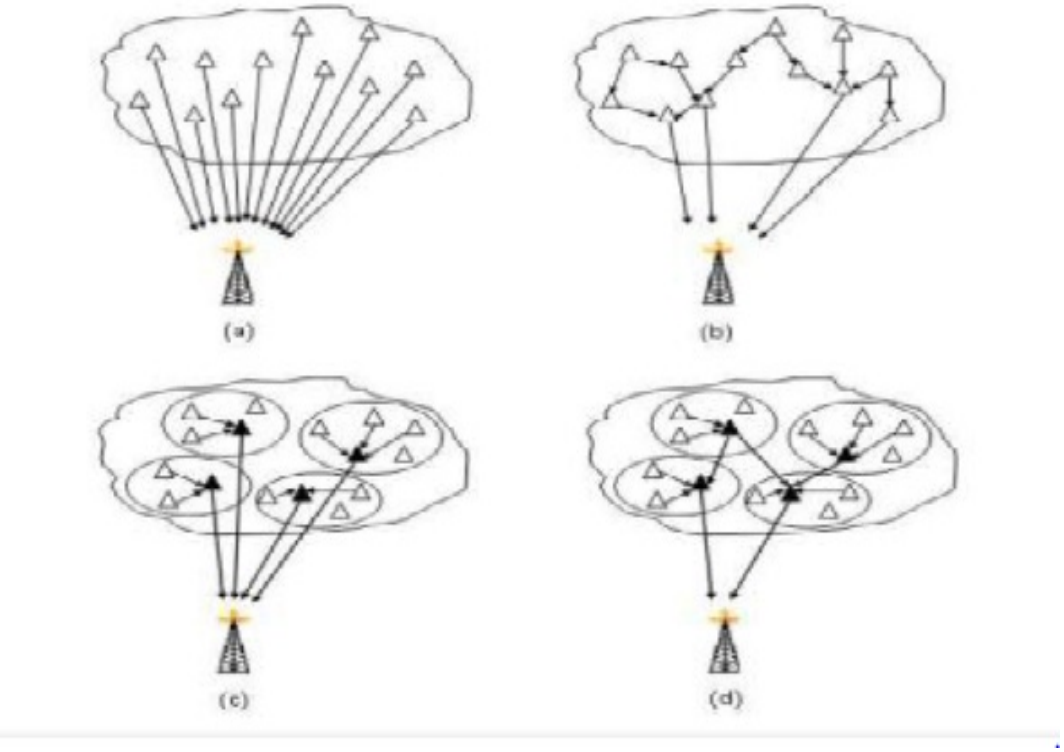

FIGURE 1.6 – Architecture de communication dans les RCSF.

1.4.5 Caractéristiques des réseaux de capteurs

Un réseau de capteurs présente des caractéristiques particulières comparativement aux autres réseaux sans fil. Dans cette section, nous présentons les principales caractéristiques de ces réseaux :

- **Sans infrastructure :** les RCSF appartiennent à la famille des réseaux sans fil

sans infrastructure. Les capteurs sont généralement déployés aléatoirement dans des zones hostiles ce que nécessite qu'ils doivent s'auto-configurer et s'auto-organiser sans intervention humaine.

- **Scalabilité (Passage à l'échelle) :** dans les RCSF, les capteurs sont déployés généralement en grand nombre pour garantir la couverture totale de la zone d'intérêt et faire face aux pannes puisque les capteurs peuvent cesser de fonctionner pour différentes causes. Nous pouvons avoir dans certains cas des RCSF de haute densité dont la taille dépasse mille capteurs voire un million de capteurs.
- **Interférences :** la notion d'interférences apparaît dans la plus part des réseaux sans fil en particulier dans les RCSF où deux capteurs voisins peuvent transmettre dans la même bande de fréquences, ce qui peut causer des interférences.
- **Topologie dynamique :** les capteurs sans fil peuvent être placés sur des objets mobiles par exemple sur des animaux pour les surveiller à distance sans perturber leur comportement. Ce type de scénario génère une topologie qui n'est pas statique dite dynamique [14].
- **Contrainte d'énergie, de stockage et de calcul :** la caractéristique la plus critique dans les RCSF est la limite des ressources énergétiques car la plupart des capteurs sont dotés de piles à énergie limitée. A cet effet, dans la plupart des travaux de recherche, une problématique est traitée conjointement avec l'économie de l'énergie.

1.4.6 Domaine d'applications des Réseaux de capteurs Sans Fil

Les réseaux de capteurs sans fil (RCSF) ont un champ d'application vaste et diversifié. Ceci est rendu possible par leur cout faible, leur taille réduite, le support de communication sans fil utilisé et la large gamme des types de capteurs disponibles. Un autre avantage est la possibilité de s'auto-organiser et d'établir des communications entre eux sans aucune intervention humaine, notamment dans des zones inaccessibles ou hostiles. Ce qui accroit davantage le nombre de domaines ciblés par leur application (environnement, catastrophes naturelles, bâtiments intelligents, la santé, l'agriculture, l'industrie...etc.). Nous présentons dans ce qui suit les domaines les plus ciblés par les RCSF [15].

- **Domaine militaire :** les RCSF sont le résultat de la recherche militaire. Ils sont utilisés dans la surveillance des champs de bataille pour connaitre exactement la position, le nombre, l'armement (chimique, biologique, nucléair etc.), l'identité et le mouvement des soldats et ainsi empêcher leur déploiement sur des zones à risques. [16]

- **Domaine civil :** apparus dans plusieurs contextes notamment dans la surveillance des habitations (concept de bâtiments intelligents), des infrastructures, des installations et des zones à risques. Leur utilisation permet de réduire considérablement le budget consacré à la sécurité des humains tout en garantissant des résultats sûrs et fiables.
- **Domaine agricole et environnemental :** les réseaux de capteurs sans fil sont très utiles dans la protection de l'environnement. Ils peuvent être utilisés pour la détection des feux de forêts, des inondations, de la surveillance des volcans, du contrôle de la qualité de l'air, le déplacement des animaux...etc. [17]
- **Domaine industriel :** le suivi des chaines de production dans une usine, détection des dysfonctionnements de machines, suivi du mouvement des marchandises dans les entrepôts de données, suivi du courrier, des colis expédié etc.
- **Domaine de la santé :** un moyen très efficace pour le domaine médical et le suivi temps-réel de l'état des patients, notamment ceux atteints de maladies chroniques Ils permettent de collecter des informations physiologiques de meilleure qualité pouvant être stockées pour une longue durée ou alors détecter des comportements anormaux chez des personnes âgées ou handicapées comme les chutes, les chocs, les cris...etc. [18]
- **Applications domestiques :** les capteurs peuvent être embarqués dans des appareils électroménagers (aspirateurs, micro-ondes, climatiseurs, réfrigérateurs...etc.) et d'interagir entre eux et avec un réseau externe pour assurer un meilleur contrôle à distance de ces appareils par leur propriétaire [19].

1.5 Réseaux tolérants aux délais (DTN)

Un réseau tolérant aux délais est un réseau de plusieurs réseaux régionaux, c'est un overlay au-dessus de ces réseaux régionaux incluant le réseau Internet. Un DTN supporte l'interopérabilité entre les réseaux [21] :

- En s'accommodant de longs délais entre (ou dans) les réseaux régionaux.
- En traduisant les caractéristiques de communication entre les réseaux régionaux. Les DTNs accommodent la mobilité et l'énergie limitée des appareils de communication sans fil.

  Notons que, les technologies sans fil DTN sont diverses, nous trouvons alors :

– La radio fréquences (RF) : Une forme de communication sans fil qui permet de transmettre l'information d'un terminal à une station de base, qui à son tour la transmet à un ordinateur hôte.

– LUWB : Une technologie radio ultra large bande, utilisée pour la communication haut débit sur courte distance, et avec une très faible puissance. La bande passante UWB est définie comme ayant une largeur d'au moins 500 MHz.
– La liaison dégagée optique (Laser) : Une technologie qui a un avantage économique certain sur les solutions filaires où la seule condition pour l'installer est de garantir une vue dégagée des obstacles entre les deux points. Ce qui impose des émetteurs/-récepteurs sur des points hauts, des fixations fiables interdisant tout mouvement des matériels et l'absence d'éléments perturbateurs tels que : le flux d'air d'une bouche d'aération ou la poussière.

1.6 Caractéristiques des réseaux DTN

- **Une connectivité intermittente :** s'il n'y a pas de chemin de bout en bout (partitionnement du réseau), un protocole comme TCP/IP ne peut pas fonctionner. D'où le besoin de nouveaux protocoles.
- **Des Délais longs et variables :** Dus au problème de propagation et au temps d'attente dans les files des nœuds intermédiaires. Par conséquent, les protocoles Internet et les applications qui comptent sur le retour d'acquittement rapide ne pourront pas fonctionner.
- **Vitesse de transmission asymétrique :** Les protocoles Internet supportent une asymétrie modérée. Néanmoins dans le cas d'asymétrie importante, cela empêche le bon fonctionnement des protocoles conventionnels.
- **Taux d'erreur important :** Les erreurs de bits sur une liaison exigent des corrections (en ajoutant des bits et du traitement) ou la retransmission du paquet complet (donc plus de trafic réseau). Pour un taux d'erreur donné sur un lien, moins de retransmissions sont nécessaires quand il s'agit du cas de retransmission saut par saut que du cas de retransmission de bout-en-bout.

1.7 Conclusion

Dans ce chapitre, nous avons défini et décrit brièvement ce qu'est un réseau Ad-hoc et ses caractéristiques ainsi qu'un réseau de capteurs sans fil qui est un type particulier de réseau Ad-hoc. Nous avons décrit le capteur, ses fonctionnalités et son architecture. Nous avons cité les caractéristiques d'un réseau de capteurs et présenté quelques applications. Un état de l'art sur les réseaux tolérants aux délais a été présenté. Dans lequel nous avons introduit la notion de DTN. Dans la suite, nous présenterons plusieurs protocoles de routage utilisés dans les réseaux sans fils.

CHAPITRE 2

ETAT DE L'ART SUR LES PROTOCOLES DE ROUTAGE

2.1 Introduction

Lors de la transmission d'un paquet de données d'une source vers une destination, il est nécessaire de faire appel à un protocole de routage qui acheminera correctement le paquet par le meilleur chemin. Les nœuds d'un réseau de capteurs sont dotés de plusieurs fonctionnalités telles que la recherche et la maintenance d'un chemin qui satisfait certaines contraintes.

Dans ce chapitre nous allons présenter les différents protocoles de routage dans les réseaux mobiles et les réseaux de capteurs sans fil et détailler les protocoles de routage basés sur la qualité de service.

2.2 Définition du routage

Le routage est une méthode d'acheminement des informations vers la bonne destination à travers un réseau de connexion donné, il consiste à assurer une stratégie qui garantit, à n'importe quel moment, un établissement de routes correctes et efficaces entre n'importe quelle paire de nœuds appartenant au réseau. Ce qui assure l'échange des messages d'une manière continue. Vu les limitations des réseaux ad hoc, la construction des routes doit être faite avec un minimum de contrôle et de consommation de bande passante.

2.3 Objectifs du routage

L'objectif majeur du routage est la conservation des métriques de la QoS comme le délai de transmission, la consommation d'énergie, etc. Le routage doit être capable d'acheminer les informations avec les critères suivants :

- Une économie d'énergie (pour les RCsf).
- En minimisant la charge du réseau, afin de maximiser la qualité totale d'informations pouvant être transmises sur le réseau dans son ensemble.
- En étant fiable, et en particulier, tolérant aux pannes des nœuds du réseau.

2.4 Classification des protocoles de routage

Vue la difficulté du routage dans les réseaux, les stratégies existantes utilisent une variété de techniques afin de résoudre ce problème. Suivant ces techniques, plusieurs classifications sont apparues parmi lesquelles nous allons citer :

Les Protocoles de routage proactifs, les Protocoles de routage réactifs et les protocoles de routage hybride.

2.4.1 Protocoles proactifs (table driven)

Sont caractérisés par le fait que les nœuds tiennent à jour une table de routage de l'ensemble du réseau. Chaque nœud envoie périodiquement des messages sur l'ensemble du réseau afin d'y informer des variations de topologie. L'avantage de ce type de protocole est qu'à tout moment, une route entre deux nœuds est connue et peut être utilisée pour acheminer des paquets. L'inconvénient est la réaction assez lente aux changements de topologie : si un lien "casse" entre deux nœuds, le réseau doit attendre de recevoir l'information sur le changement de topologie pour être averti du problème.

2.4.1.1 OLSR

Comme son nom l'indique, le protocole OLSR (Optimized Link-State Routing) [22] est une version optimisée du principe de routage à état de liens, mais destinée aux réseaux sans fil. Comme expliqué précédemment, un des problèmes du principe de routage à état de liens est le risque de congestion lié à l'échange massif de messages. Le principal objectif de ce nouveau protocole est de limiter cet échange de messages tout en garantissant que chaque nœud ait en sa possession les informations suffisantes pour choisir les meilleurs next-hops. Le protocole OLSR propose de sélectionner, pour chaque nœud du réseau,

un sous-ensemble de voisins qui se chargeront de retransmettre les messages de contrôle de topologie reçus. Les autres nœuds, qui ne font pas partie de ce sous-ensemble devront uniquement traiter l'information reçue sans la retransmettre.De plus, le nombre de voisins déclarés dans chaque message est limité. Grâce à ces optimisations, la charge transitant sur le réseau est sensiblement réduite. Ce qui implique indirectement une diminution du risque d'interférence de signal [23].

2.4.1.2 Relais Multipoint

Les MPR (MultiPoint Relays) sont des nœuds chargés de la retransmission des messages de contrôle de topologie. Tout nœud « n » du réseau doit choisir un ou plusieurs de ses voisins comme MPR de telle sorte que tous les nœuds situés à deux sauts du nœud d'origine « n » soient accessibles via un de ses MPR. Chaque nœud choisi comme MPR ajoute le nœud « n » à sa liste de sélecteurs MPR.

La principale difficulté réside dans le choix des MPR. Moins les MPR sont nombreux, moins la densité de trafic est importante. Différentes approches sont possibles pour déterminer le choix des MPR. Il est difficile de trouver une approche optimale : un algorithme trop simple choisirait des MPR peu efficaces, mais un algorithme trop complexe aurait un impact sur les performances du protocole. Ce problème s'apparente à celui de couverture d'un graphe et comme expliqué par A. Qayyum et al. [23], la recherche de MPR est un problème NP-complet. Dans leur rapport, ces chercheurs proposent plusieurs heuristiques. Celle de base consiste à sélectionner comme premier MPR le voisin qui couvre le plus de nœuds à deux sauts qui ne sont atteignables par aucun autre voisin. Ensuite, il reste à choisir les MPR restant en commençant par ceux couvrant le plus de nœuds à deux sauts. La Figure 2.1 illustre la diminution de l'envoi de messages de contrôle de topologie grâce à l'utilisation des MPR. Cette diminution a comme conséquence de réduire le taux d'erreur des paquets à la réception (car moins de risques d'interférences), mais aussi le temps nécessaire pour que tous les nœuds reçoivent les messages. Le rapport de recherche indique que ce temps peut être divisé par deux.

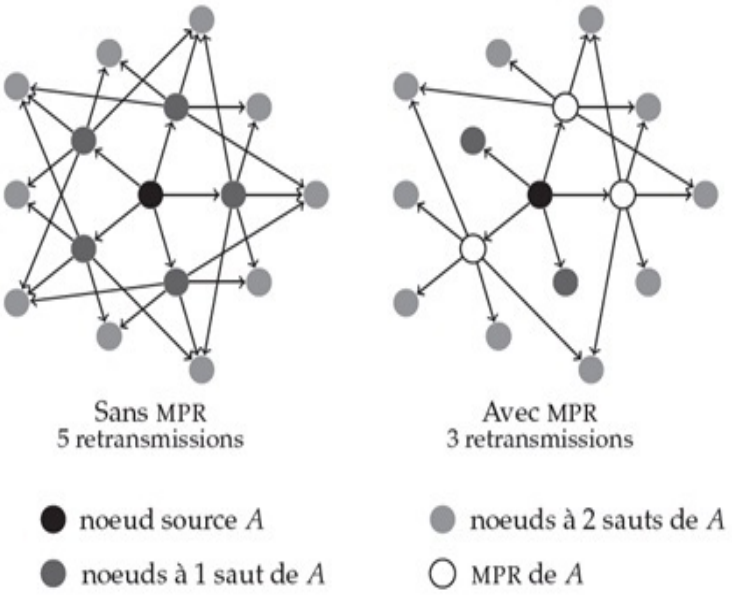

FIGURE 2.1 – Impact de l'utilisation des MPR dans OLSR (les nœuds blancs sont les MPR du nœud central) ;

- **Découverte du voisinage**

  La découverte du voisinage se fait par l'intermédiaire de messages HELLO. Périodiquement, chaque nœud broadcast un message HELLO contenant l'ensemble de ses voisins bidirectionnels, l'ensemble de ses voisins unidirectionnels et l'ensemble de ses voisins choisis comme MPR. Ces messages ne doivent jamais être retransmis par les nœuds qui les reçoivent. Les messages HELLO ont deux rôles :

– déterminer le type de liens existant entre un nœud et ses voisins (lien bidirectionnel ou unidirectionnel).

– déterminer les voisins qui serviront de MPR.

  Pour comprendre comment ces deux rôles sont remplis, il est nécessaire d'introduire les différents cas de figure qui peuvent se présenter lors de la réception d'un message HELLO. Supposons qu'un nœud B reçoit un message HELLO broadcasté par un nœud A.

  a. Si B et A ne se connaissent pas, c'est-à-dire que B n'a encore reçu aucun message HELLO de la part de A et inversement, alors B ajoute le nœud A à sa liste de voisins unidirectionnels.

  b. Si A connaît B, c'est-à-dire que A à déjà reçu un message HELLO de B (et l'a donc ajouté à sa liste de voisins unidirectionnels ou bidirectionnels), cela signifie que B a reçu un message contenant son propre identifiant dans la liste des voisins unidirectionnels ou bidirectionnels de A. Dans ce cas, A et B sont accessibles l'un a

l'autre, et donc B ajoute le nœud A à sa liste de voisins bidirectionnels.

c. Si un autre nœud C différent de A et B est présent dans la liste des voisins bidirectionnels de A, il s'agit d'un nœud accessible depuis B en 2 sauts par l'intermédiaire du nœud A. Un algorithme doit alors déterminer si le nœud A doit être choisi ou non comme MPR de B pour accéder au nœud C. Si c'est le cas, alors B ajoute le nœud A à sa liste de MPR.

d. Si B est présent dans la liste des MPR de A, alors B retient A comme étant un de ses sélecteurs MPR. Ainsi, après un certain nombre de messages échangés, chaque nœud du réseau connaît la présence et l'état des liens avec ses voisins, les voisins qu'il a choisis comme MPR et les voisins qui l'ont choisi comme MPR. La Figure 2.2 illustre ces différents cas de figure.

- **Distribution de la topologie**

  La distribution de la topologie du réseau se fait par l'intermédiaire des messages TC (Topology Control). À intervalles réguliers, chaque MPR broadcast envoie un message TC contenant l'ensemble de ses voisins qui l'ont choisi comme MPR. Chaque message contient également un numéro de séquence pour reconnaître les messages périmés. Les messages TC ne sont retransmis que par les nœuds désignés comme MPR par le voisin ayant transmis le message. Les messages TC ont 2 rôles :

– distribuer la topologie du réseau.
– détecter les changements de topologie du réseau.

Les messages TC peuvent être comparés aux messages LSA du protocole de routage à état de liens. Les deux principales différences sont les suivantes :

a. les messages LSA diffusent sur le réseau l'ensemble des voisins d'un nœud A alors que les messages TC ne diffusent que les voisins ayant choisi le nœud A comme MPR.

b. les messages LSA sont broadcastés et retransmis par tous les nœuds du réseau alors que les messages TC ne sont retransmis que par les nœuds qui sont des MPR du voisin ayant transmis le message.

La première différence a l'avantage de diminuer la taille des messages diffusés sur le réseau en limitant le nombre de voisins déclarés dans un même message. La deuxième différence permet de limiter le nombre de retransmission des messages TC. Ces deux différences permettent de diminuer la charge sur le réseau tout en garantissant que les messages soient diffusés à tous les nœuds du réseau et qu'un nombre suffisant de nœuds nécessaires au routage soient déclarés. Ce protocole considère que la perte de messages TC n'est pas contraignante vu que ceux-ci sont broadcastés périodiquement.

Comme pour le principe de routage à état de liens, chaque nœud génère une carte du réseau à partir des messages TC reçus. Étant donné que tous les voisins ne sont pas déclarés dans un message TC, la carte ne contient qu'une partie du réseau. Néanmoins, cette carte simplifiée garantit que toutes les destinations accessibles par le principe de routage à état de liens le sont aussi via cette méthode.

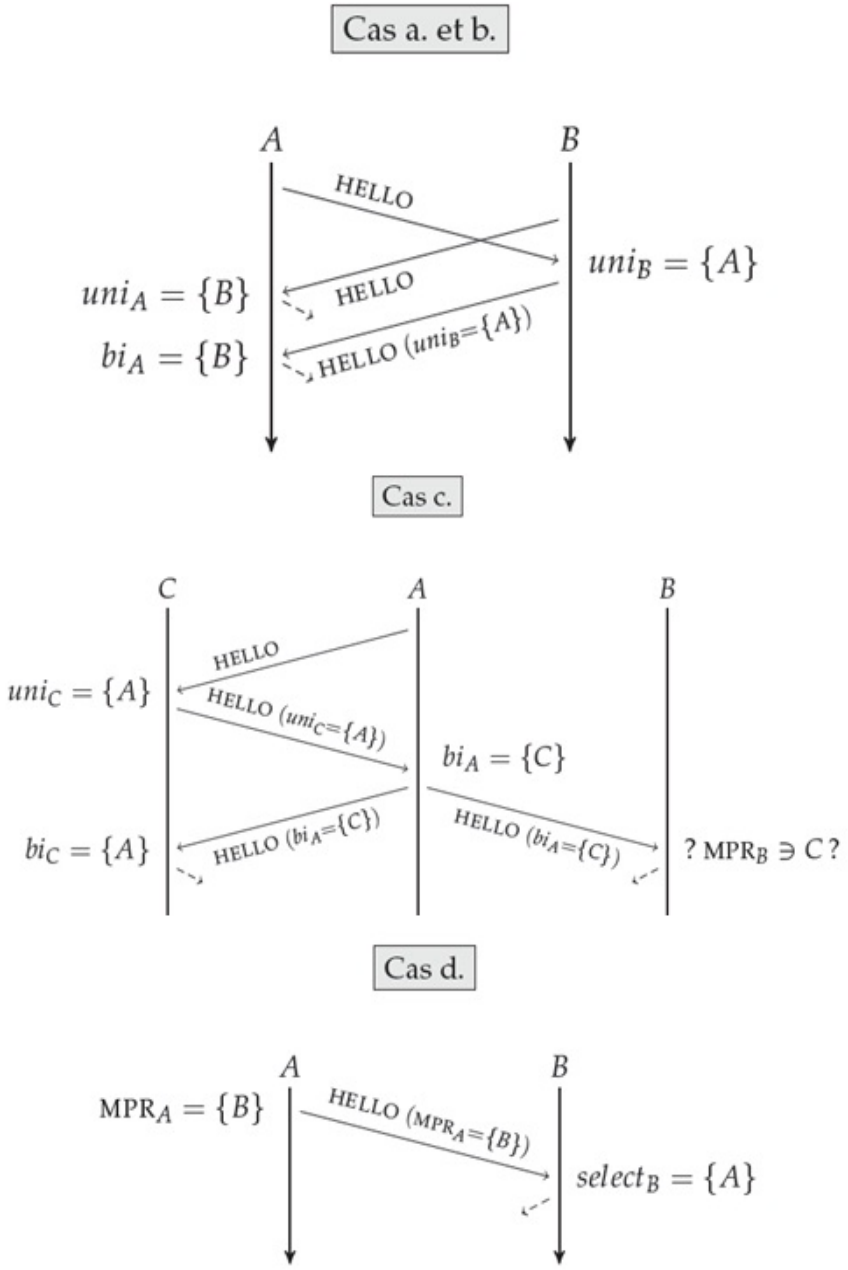

FIGURE 2.2 – Découverte du voisinage dans OLSR.

- **Détermination des meilleures routes**

  Comme pour le principe de routage à état de liens, un algorithme de calcul de plus

court chemin est exécuté sur la carte du réseau afin de déterminer les meilleurs next-hops pour atteindre chaque destination du réseau. À partir de là, des tables de routage sont générées ou mises à jour lors de changements de topologie détectés via les messages TC.

2.4.1.3 B.A.T.M.A.N

Le protocole B.A.T.M.A.N. (Better Approach To Mobile Ad hoc Network) [14] est une approche destinée à remplacer peu à peu le protocole OLSR et vise à combler ses principaux défauts ; il ne s'agit toutefois pas d'un protocole à état de liens. Le problème d'OLSR est qu'il ne tient pas compte, dans ses choix de meilleurs next-hops, de la perte possible de paquets (à cause de défaillances de liens, brouillage du signal, etc.). L'approche proposée par B.A.T.M.A.N. est de choisir le next-hop situé sur la route la plus fiable, c'est-à-dire celle ayant le moins de risque de perte de paquets. Pour ce faire, chaque nœud garde un historique contenant, pour tout autre nœud du réseau, tous les numéros de séquence de tous les messages reçus de la part de chaque voisin. Le voisin ayant le plus de numéros de séquence dans sa table est considéré comme next-hop le plus fiable pour atteindre le nœud d'origine du message.

- **Originator Message**
  L'OGM (Originator Message) est l'unique type de message qui transite au sein du réseau. Ce message est broadcasté périodiquement par tous les nœuds afin de découvrir la topologie du réseau. Un OGM contient entre autres l'identifiant du nœud d'origine du message, un numéro de séquence pour déterminer les messages périmés et un flag indiquant si le type de lien existant avec le dernier nœud ayant envoyé le message. Chaque message est retransmis par tous les nœuds afin de se répandre sur tout le réseau.
- **Découverte du voisinage**
  La découverte des voisins s'effectue grâce au flag présent dans les messages OGM indiquant si un lien avec un voisin est unidirectionnel ou bidirectionnel. La façon de déterminer le type de lien existant avec un voisin est similaire à l'échange du message HELLO dans le protocole OLSR.
- **Distribution de la topologie**
  Le contrôle de la topologie est la principale fonction des messages OGM. Chaque nœud tient en mémoire les informations suivantes pour chaque destination :

– l'adresse de la destination.
– le numéro de séquence le plus à jour, permettant de détecter les réceptions dupliquées

d'un même message.

– un tableau contenant, pour chaque voisin bidirectionnel ayant transmis un OGM, l'ensemble des numéros de séquence reçus.

Le dernier tableau signifie que chaque nœud compte le nombre de messages OGM différents reçus d'un même voisin pour une même destination. Le voisin pour lequel le plus de messages différents a été reçu est considéré comme se trouvant sur la route la plus fiable pour atteindre la destination.

- **Détermination des meilleures routes**

  À partir des informations récoltées de ses voisins, chaque nœud choisit comme next-hop le voisin qui lui a envoyé le plus de paquets avec des numéros de séquence différents. Le paquet est envoyé à ce voisin, qui répète lui aussi la même opération, jusqu'à ce que le paquet arrive à destination. Le protocole B.A.T.M.A.N. n'est encore qu'à l'état d'ébauche et présente encore quelques problèmes. Entre autres, celui ci ne possède pas de système de détection de boucles de routage et ne tient pas compte des coûts des liens.

2.4.1.4 DSDV

Le protocole DSDV (Destination Sequence Distance Vector) [25] est une version améliorée du protocole à vecteur de distances. Les optimisations proposées permettent de mettre fin au problème de boucle et de comptage à l'infini et diminuent la charge sur le réseau.

- **Types de messages**

  Deux nouveaux types de messages sont introduits : les full-dumps qui transportent la table de routage complète mais qui sont envoyés plus rarement, et les mises à jour incrémentales qui informent uniquement de changements de topologie apportés depuis le dernier full dump. Les mises à jour incrémentales ont l'avantage de réduire la quantité de données transmise sur le réseau.

- **Numéro de séquence**

  Le problème de comptage à l'infini est réglé grâce à l'introduction de numéros de séquence. Chaque entrée dans les tables de routage est associée à un numéro de séquence pair choisi par la destination de l'entrée. Chaque destination génère donc un numéro de séquence pair qu'il envoie en même temps que sa table et qu'il incrémente à chaque envoi. Lorsqu'un voisin reçoit le vecteur de distances, il compare le numéro de séquence associé au vecteur de distances avec celui de l'entrée correspondant à

l'envoyeur dans sa table de routage. L'entrée n'est mise à jour que si le numéro de séquence est supérieur. Si les deux numéros de séquence sont égaux, la route avec le coût minimum est choisie. Si un nœud détecte qu'un voisin devient inaccessible, il incrémente de 1 le numéro de séquence de l'entrée correspondante dans la table de routage pour rendre le numéro impair, signifiant que la route vers la destination est inaccessible. Le vecteur de distances est propagé et provoque des mises à jour par les voisins étant donné que le numéro de séquence a augmenté.

En comparaison avec les protocoles à vecteur de distances, la taille des messages est sensiblement réduite. Par contre, le nombre de messages envoyés est plus important. Il est donc nécessaire d'avoir une bonne fréquence d'envoi de mises à jour incrémentales.

2.4.2 Protocoles réactifs (on demande)

Sont caractérisés par le fait que, contrairement aux protocoles proactifs, les nœuds ne tiennent pas à jour de tables de routage. Lorsqu'un nœud souhaite contacter un autre nœud, il fait une demande de route à ses voisins, qui eux-mêmes demandent à leurs voisins, jusqu'à tomber sur un nœud connaissant l'information. Le principal inconvénient est le temps nécessaire pour obtenir l'information sur une route. Par contre, l'avantage est que ce type de protocole réagit très bien aux changements de topologie : le nœud faisant la demande de route obtient toujours une route à jour. De plus, l'espace mémoire nécessaire pour stocker les tables de routage est faible.

2.4.2.1 AODV

Le protocole AODV (Ad hoc On Demand Distance Vector) [26] est une variante du protocole DSDV en y intégrant le concept des protocoles réactifs. Pour rappel, le protocole DSDV est un protocole proactif, ce qui signifie que chaque nœud conserve une table de routage de l'ensemble des destinations du réseau, ce qui nécessite une place mémoire non négligeable. Le protocole AODV propose de tirer parti des bénéfices de DSDV tout en évitant de surcharger la mémoire des nœuds. Si un nœud souhaite contacter une destination du réseau et qu'il ne connaît pas encore le next-hop à emprunter, il diffuse une demande de route sous la forme d'un message RREQ (Route Request). La réponse est ensuite reçue via un message RREP (Route Response). Une table de routage temporaire est créée dans laquelle sont enregistrés les next-hops pour chaque destination demandée. Une entrée dans la table de routage reste présente tant que la route est utilisée. Si un nœud détecte un lien défaillant sur une route, il en informe les nœuds de la route via un

message RERR (Route Error). La suite présente plus en détails les caractéristiques de ce protocole.

- **Demande de route**

  Chaque nœud qui souhaite contacter une destination du réseau qui n'est pas dans sa table de routage diffuse en broadcast un message RREQ contenant la source du message, la destination à atteindre et un numéro de séquence pour détecter les réceptions dupliquées d'un même message. Un nœud qui reçoit ce message enregistre le voisin qui le lui a envoyé et rediffuse le message à son tour. L'ensemble des messages envoyés crée ainsi un ensemble de routes temporaires allant de tout nœud ayant reçu le message RREQ au nœud d'origine l'ayant envoyé. Lorsqu'un nœud possédant l'information demandée reçoit le message RREQ, celui-ci renvoie un message RREP contenant l'information demandée sur la destination. Le message RREP est renvoyé au nœud d'origine via la route temporaire par laquelle le message RREQ est arrivé. Chaque nœud de cette route qui reçoit le RREP enregistre le voisin à contacter pour atteindre la destination et retransmet le RREP jusqu'au nœud d'origine. Si un nœud reçoit plusieurs RREP, il garde celle de coût de minimal via l'équation de Bellman-Ford. Les informations sur la route ainsi produite sont conservées par chaque nœud concerné jusqu'à ce que celle-ci ne soit plus utilisée ou devienne invalide. Si un nœud présent sur une route détecte un lien invalide, il diffuse un message RERR le long de la route jusqu'à la source pour informer que cette route n'est plus valide. Dans ce cas, si le nœud source souhaite encore contacter la destination, il doit relancer le processus de demande de route. Les figures 2.3 et 2.4 illustrent le concept de demande de route.
- **Boucles et comptage à l'infini**

  Pour éviter l'apparition de comptage à l'infini et de boucles, chaque route est associée à un numéro de séquence choisi par le nœud destination, comme pour DSDV. Un numéro pair signifie que la route est valide, un numéro impair signifie qu'une route n'est plus valide. Lorsqu'un nœud reçoit deux réponses ayant des numéros de séquences différents, il choisit la route ayant le numéro le plus élevé, si les deux sont pairs. Si les deux numéros de séquences sont égaux, le nœud choisit la route ayant le moins de sauts. Si un nœud reçoit une réponse avec un numéro de séquence impair, la route associée est oubliée par le nœud. Chaque nœud doit donc enregistrer deux numéros de séquence : un pour détecter les réceptions dupliquées de messages RREQ et un pour détecter les nouvelles routes.

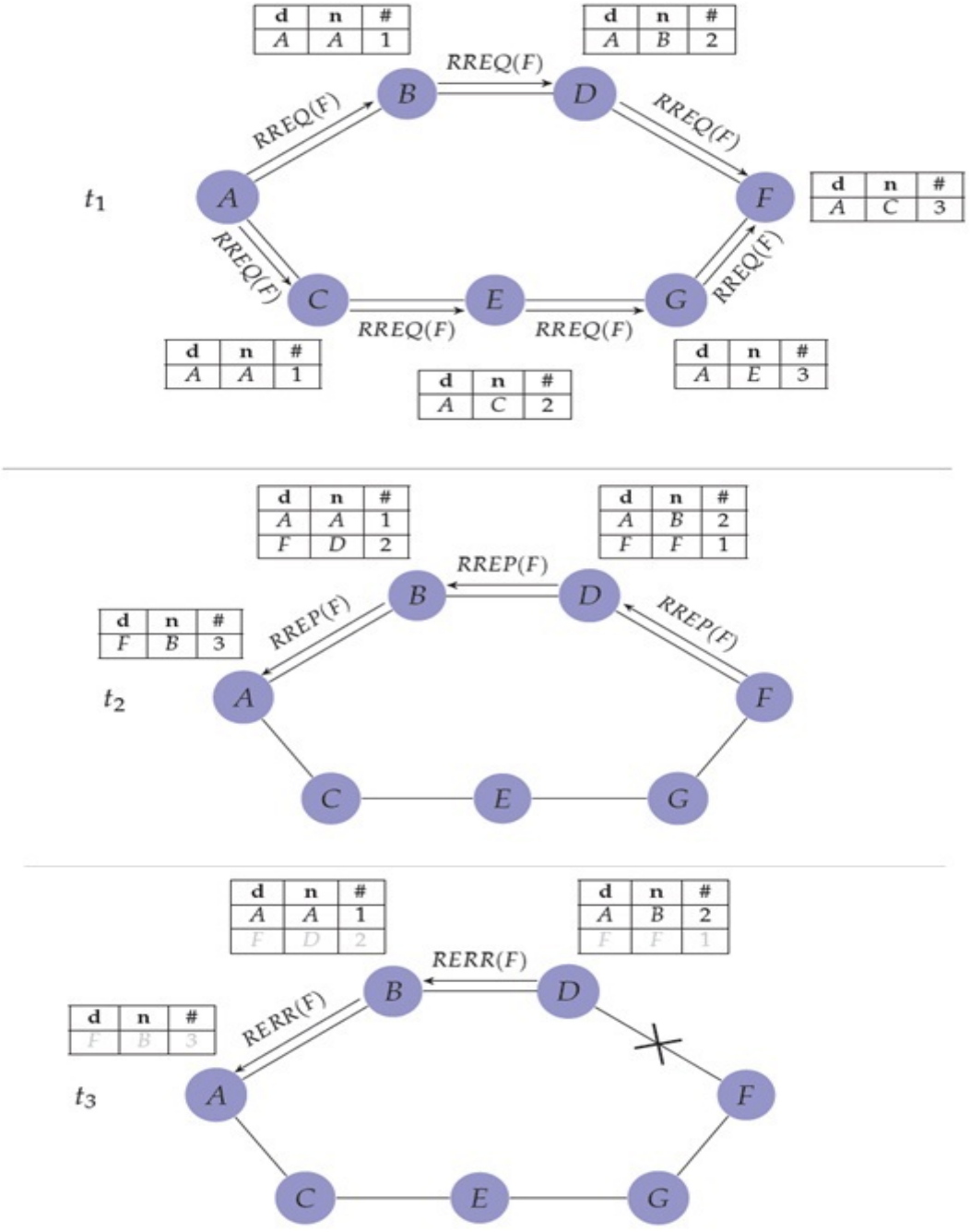

FIGURE 2.3 – Demande de route dans le protocole AODV.

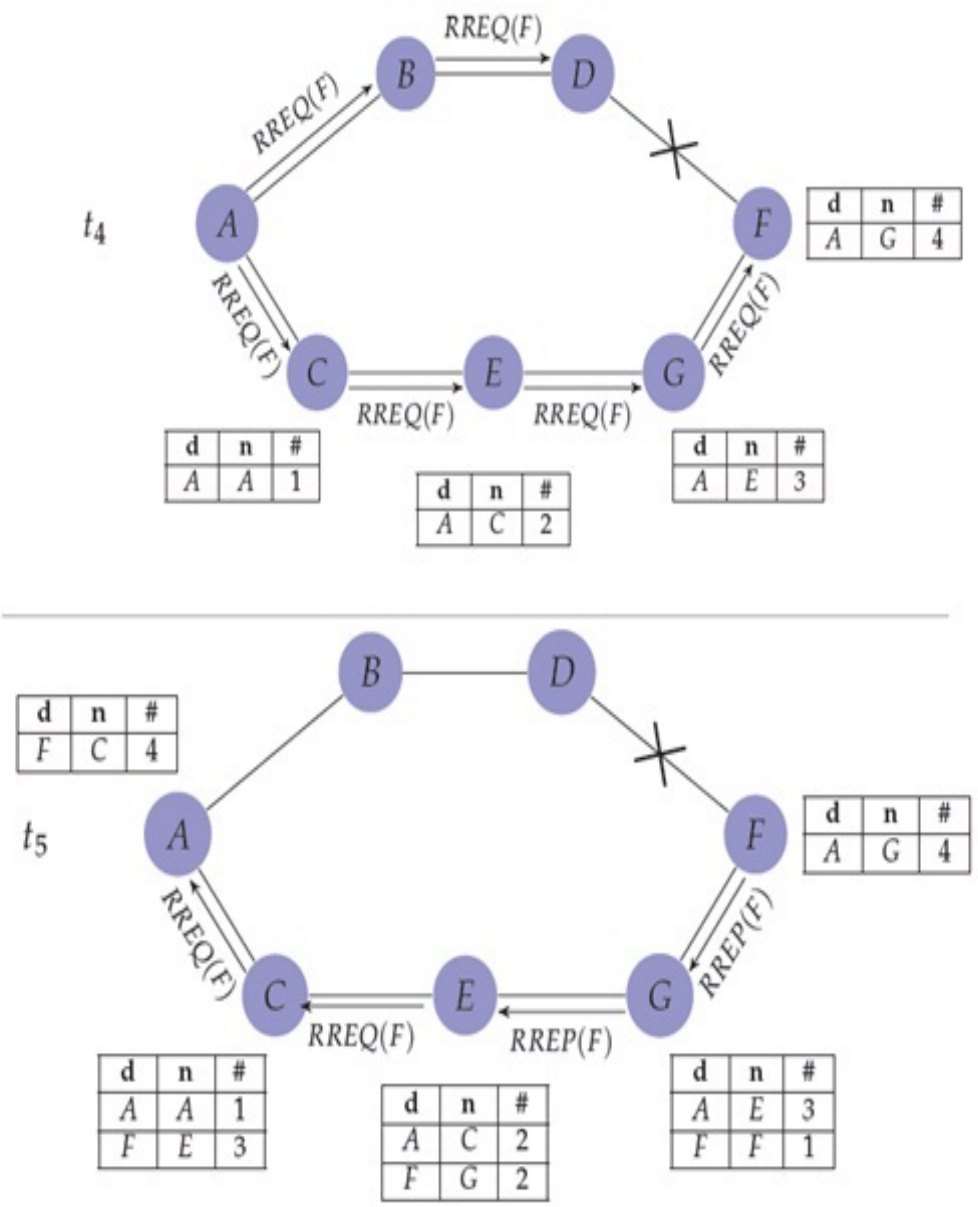

FIGURE 2.4 – Demande de route dans le protocole AODV (suite).

Ce protocole a l'avantage de proposer des routes constamment à jour. De plus, tant que des paquets de données sont envoyés à une destination, la route est gardée en mémoire et le paquet peut être directement envoyé vers la destination. Dans le pire des cas, si des changements de topologie sont fréquents, il suffit de renvoyer des demandes de routes.

2.4.2.2 DSR

Le protocole DSR (Dynamic Source Routing) [27] est un protocole réactif où les routes sont maintenues à la source. Comme pour n'importe quel protocole réactif, DSR crée des routes à la demande lorsqu'une destination doit être atteinte. Contrairement à tous les

protocoles de routage ad hoc présentés ci-dessus, DSR ne stocke jamais de table contenant les next-hops à emprunter pour atteindre différentes destinations. En effet, les routes complètes vers les destinations sont maintenues par chaque nœud source devant envoyer un paquet. Cette manière de faire à l'avantage de faciliter la détection de boucles et de comptage à l'infini. De plus, plusieurs routes vers une même destination peuvent coexister, c'est à la source de faire son choix. Par contre, cette façon de faire crée un overhead sur les paquets envoyés. En effet, la route choisie doit être insérée dans le paquet afin que les nœuds de cette route aient connaissance des next-hops à emprunter. Ce principe se nomme la source routing. Le fonctionnement de ce protocole se découpe en deux parties appelées Route Discovery et Route Maintenance et détaillées ci-dessous.

- **Découverte de route**

  La découverte de routes s'effectue par le broadcaste de messages RREQ comme pour les autres protocoles réactifs. Au fur et à mesure que le message RREQ se propage sur le réseau, la liste des nœuds parcourus est insérée dans le message. Le nœud destination ou un nœud intermédiaire ayant l'information demandée stocke la liste des nœuds formant la route demandée dans un message RREP et l'envoie en retour à la source en remontant la route ainsi créée. Une fois que la source reçoit la réponse voulue, elle stocke la route obtenue dans une table. Si la source reçoit plusieurs RREP, elle choisit la route qu'elle souhaite emprunter. L'avantage de cette pratique est que le nœud source du paquet est totalement libre de choisir la route à emprunter pour arriver à la destination. Avec cette façon de faire, plusieurs routes vers une même destination peuvent exister. Par exemple, un nœud A situé sur la route entre la source X et la destination Y peut choisir une route différente que celle de X s'il souhaite envoyer des paquets à Y.

- **Maintenance de route**

  Pour détecter les défaillances de liens, chaque nœud qui envoie un paquet ou un message différent d'un RREQ doit vérifier que les nœuds suivants sur la route reçoivent bien le paquet. Si ce n'est pas le cas, il faut renvoyer un RERR à la source pour l'avertir de cette défaillance.

2.4.3 Les protocoles hybrides

Ce type de protocole est compromis entres les protocoles proactifs et réactifs et ce sont des protocoles qui d'un côté utilisent une procédure de détermination de route sur demande mais de l'autre un coût de recherche limité.

2.4.3.1 ZRP (Zone routing protocol)

Le protocole ZRP est un model hybride entre schémas proactif et un schéma réactif Le principal problème dans l'élaboration d'un protocole de routage pour réseau ad hoc réside dans le fait que pour déterminer le parcours d'un paquet de données, le nœud source doit au moins connaître les informations permettant d'atteindre ses proches voisins d'un autre côté la topologie d'un tel réseau change fréquemment. De plus comme le nombre de nœuds peut être élevé, le nombre de destination potentiel peut également l'être. Ce que requiert des échange de donnée important et fréquent ; donc la quantité de donnée de mis a jour du trafic peut être conséquent. Cela est en contradiction avec le fait que toutes les mises à jour dans un réseau interconnecté adhoc circulent dans l'air et donc sont coûteuses en ressources. Le protocole ZRP limite la procédure proactive uniquement au nœud voisin et d'autre part, la recherche à travers le réseau (voir figure 2.5 b) est effectué d'une manière efficace dans le réseau contrairement à une recherche générale sur tout le réseau. Sur la figure 2.5(a), le nœud source S est centré dans un grand cercle délimité par des pointillés ayant une zone de routage de rayon 2 dans laquelle il se comporte comme les protocoles proactifs. Par contre sur la figure 2.5 (b), pour atteindre sa destination D, le nœud s doit trouver le chemin avec des algorithmes réactifs parce que D est en dehors de sa zone de routage à rayon 2.[28]

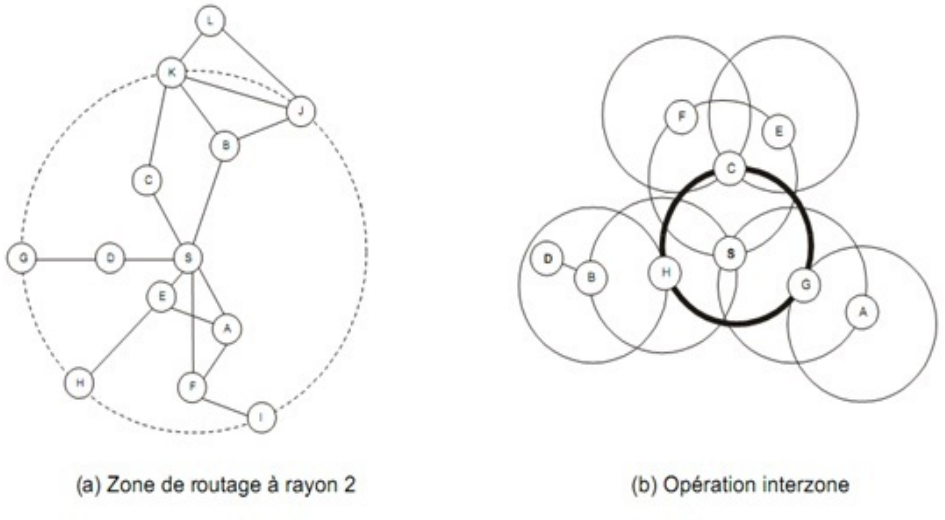

FIGURE 2.5 – Routage dans ZRP.

2.5 Récapitulatif

Voici un récapitulatif des différents protocoles proposés ainsi que leurs avantages et inconvénients respectifs. La Figure 2.6 liste les différents protocoles détaillés ci-dessus et leurs classes respectives. Le Tableau 2.1 reprend les avantages et inconvénients de chacun de ces protocoles.

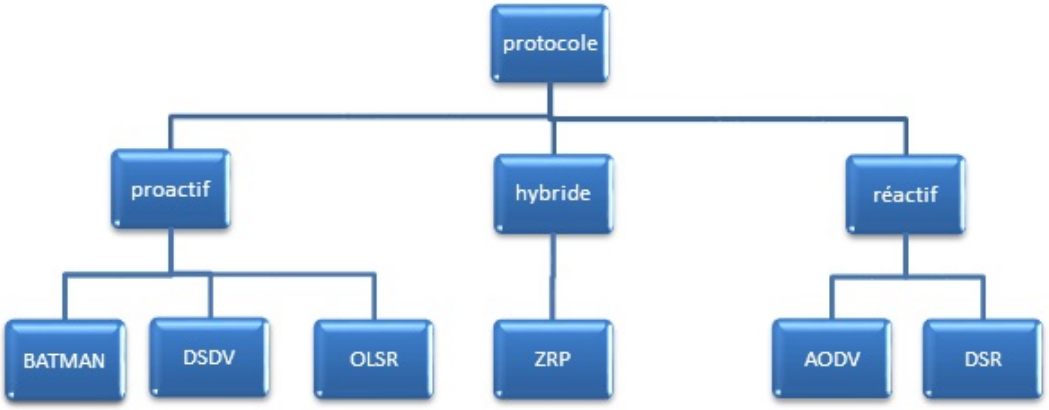

FIGURE 2.6 – Hiérarchie des Protocoles de routage ad hoc .

Protocole	**Spécificités**	**Avantages**	**Inconvénients**
Proactif	maintien constant d'une table de routage	propagation des paquets très rapide, adapté à la mobilité	réaction lente aux changements de topologie, échanges massifs de messages
OLSR	amélioration de LSR via introduction de MPR	diminution du nombre de messages échangés et de leur taille	problème si mobilité très importante
B.A.T.M.A.N.	choix de la route via les voisins ayant transmis le plus de messages	choix des routes les plus fiables	mémoire nécessaire pour stocker l'historique des messages reçus
DSDV	amélioration de DVR via suppression du problème de comptage à l'infini, mises à jour incrémentales pour modifications de topologie mineures	taille des messages diminue	envoi de messages plus important

Réactif	routes obtenues à la demande	routes toujours à jour, pas d'envois excessifs de messages de contrôle	délai nécessaire pour trouver une route, pas optimisé pour la forte mobilité
AODV	choix du next-hop situé sur la route de coût minimum	idem protocoles réactifs	idem protocoles réactifs
DSR	route complète stockée dans chaque noeud, route insérée dans le paquet à envoyer	choix de la route indépendant des autres noeuds, connaissance totale par un noeud de la route empruntée	overhead sur le paquet, mémoire pour stocker les routes

TABLE 2.1 – Avantages et inconvénients des protocoles de routage ad hoc.

2.6 Conclusion

On remarque que les protocoles réactifs envoient beaucoup moins de messages que les protocoles proactifs en cas de faible mobilité. En effet, un message n'est envoyé que si une route doit être trouvée. De plus, une fois qu'une route est trouvée, les paquets suivants sont envoyés très rapidement. Par contre dans un réseau à forte mobilité, le nombre de messages échangés sera bien plus important car il est nécessaire de réaliser de nouvelles demandes de routes plus souvent. Dans ce cas, les temps nécessaires pour qu'un paquet arrive à destination peut être plus important. À l'inverse, les protocoles proactifs permettent une transmission rapide des paquets en cas de faible mobilité, mais sont beaucoup moins efficaces dans les réseaux à forte mobilité. En effet, il est nécessaire d'attendre que de nouvelles informations sur la topologie parviennent aux nœuds pour que ceux-ci mettent à jour leur table de routage. L'avantage de ce genre de protocole est que le nombre de messages envoyés par chaque nœud reste constant, quelle que soit la mobilité du réseau.

CHAPITRE 3

CONCEPTION DU PROTOCOL DE ROUTAGE (PSCLIR)

3.1 Introduction

Vu les inconvénients des Protocoles de routage que nous avons étudié dans le deuxième chapitre et les problèmes d'implémentation de ces derniers, nous proposons un Protocole de routage qui peut couvrir les inconvénients des protocoles de routage existants ou d'améliorer leurs caractéristiques. Pour cela, nous proposons un Protocole de routage qui détermine la probabilité qu'un lien connecté à une interface d'un routeur soit connecté à un moment donné. Ce protocole déterminera à tout moment, lorsqu'un paquet arrive au niveau du routeur courant, quel lien relié à l'une des interfaces de ce dernier, serait le plus disponible à cet instant. Le protocole surveillera en permanence les interfaces par rapport à leur changement d'état, enregistrera tous les instants de changements, calculera la somme des petites durées de connexion durant chaque période d'observation et donnera les probabilités de connexion ou déconnexion à chaque instant ou période.

3.2 Principe de fonctionnement du Protocole (PSCLIR)

Le Protocol PSCLIR (protocole de surveilance de connexion des liens d'interfaces d'un routeur) fonctionne comme suit :

- Le Protocol doit s'installer sur chaque nœud (routeur) du réseau.
- Le Protocole enregistre les dates pour lesquels les états des interfaces (connecté/déconnecté) changent au niveau de chaque nœud et pendant une durée donnée. Puis, il calcule les temps de connexion et de déconnexion des interfaces pour construire

un historique des états de toutes ces interfaces.

- A l'arrivée d'un paquet au niveau d'un nœud intermédiaire sur l'une des interfaces, le nœud correspondant retransmet le paquet vers le nœud suivant. Le Protocole décide d'envoyer le paquet sur l'interface qui a la plus grande probabilité de disponibilité.

3.3 Modèle mathématique du Protocole (PSCLIR)

On représente chaque routeur du réseau par le symbole R et son indice sur le réseau par i. Donc R_1 est le routeur d'indice 1, R_2 est le routeur d'indice 2 et Ri le routeur d'indice i. Soient de même :

I_i est l'interface i d'un routeur.

L_{ij} est le lien qui relie deux routeurs R_i et R_j, $L_{ij} = (R_i, R_j)$.

T_c et T_d représentent respectivement les temps de connexion et de déconnexion.

On représente l'état d'une interface par C si elle est connectée et D sinon.

T_i le temps d'observation du changement d'état à l'instant i.

PB_c probabilité de connexion et PB_d probabilité de déconnexion.

3.3.1 Calcul de l'historique des interfaces

Au début le protocole enregistre les états des interfaces du routeur pendant une durée ou période [T1, T2]. À chaque changement d'état, le protocole enregistre la date et le nouveau passage d'état (C : connecté, D : déconnecté). Le protocole remplit une table comme suit :

temps	Interface I1	Interface I2	Interface I3	.....	.....	Interface Im
T1	**C**	**C**	**D**	.....	.....	**D**
T2	**C**	**D**	**D**	.....	.....	**C**
T3	**D**	**C**	**D**	.....	.....	**C**
T4	**D**	**C**	**C**	.....	.....	**D**
T5	**C**	**C**	**C**	.....	.....	**D**
T6	**D**	**C**	**D**	.....	.....	**D**
T7	**D**	**D**	**C**	.....	.....	**D**
T8	**C**	**C**	**C**	.....	.....	**C**
.	.	.	.	.....	.....	.
.	.	.	.	.....	.....	.
$T_{(n-2)}$	**C**	**D**	**D**	.....	.....	**C**
$T_{(n-1)}$	**D**	**C**	**C**	.....	.....	**D**
$T_{(n)}$	**C**	**D**	**D**	.....	.....	**C**

TABLE 3.1 – Calcul de l'historique des interfaces.

3.3.2 Calcul des temps de connexion et déconnexion

Après enregistrement des dates de changements d'états des interfaces pendant une durée, le Protocole PSCLIR calcule les temps de connexion T_c et de déconnexion T_d pour chaque interface I_i comme suit :

L'interface I1

C	T3-T1		T7-T5		.....		.....	
D		T5-T3		.....		.....		T(n)-T (n-1)

L'interface I2

C	T2-T1		T7-T3		.....		...	T(n)-T (n-1)
D		T3-T2		T8-T7		.....		

L'interface I3

C		T6-T4		.....		.....		T(n)-T (n-1)
D	T4-T1		T7-T6		.....		.....	

.

.

.

.

.

L'interface Im

C		T4-T2		.....		.....		...	
D	T2-T1		T8-T4		.....		...		T(n)-T (n-1)

FIGURE 3.1 – Calcul des temps de connexion et déconnexion.

3.3.3 Décision d'acheminement des paquets

Après le calcul de ces séries de données historiques de dates de connexions et de déconnexions des interfaces, et durant lequel un paquet demeure pendant Δt dans le routeur en arrivant sur l'une des interfaces, Le protocole calcule la probabilité de disponibilité

d'une des autres interfaces du routeur. Le paquet est ainsi dirigé vers l'interface appropriée, ayant une probabilité de disponibilité la plus grande. Cette probabilité est calculée par rapport à l'historique enregistré dans la table selon la formule suivante $\mathsf{PB_c} = \mathsf{T_c}/\Delta\mathsf{t}$ (probabilité de connexion) et $\mathsf{PB_d} = \mathsf{T_d}/\Delta\mathsf{t}$ (probabilité de déconnexion).

3.4 Implantation du Protocole (PSCLIR) sur un réseau

Pour implémenter le Protocole (PSCLIR), deux approches sont possibles

- Approche proactive : envoi périodiquement des tables de routage.
- Approche réactive : envoi des tables de routage à la demande.

3.4.1 Implantation du Protocole (PSCLIR) approche proactive

Dans cette approche chaque nœud envoie périodiquement à tous les nœuds du réseau, la probabilité de libération de toutes ses interfaces, pour que le nœud qui veut émettre un paquet peut calculer le meilleur chemin pour atteindre la destination.

3.4.2 Implantation du Protocole (PSCLIR) approche réactive

Dans cette approche, chaque nœud qui veut émettre un paquet, demande à ses voisins de lui donner le meilleur saut suivant et ses voisins demandent de leurs voisins et ainsi de suite jusqu'à atteindre le nœud destinataire. Après la réception des meilleurs sauts, le nœud peut calculer le meilleur chemin pour atteindre la destination.

3.5 Optimisation de la taille des paquets envoyés et les calculs effectués

Pour réduire la taille des paquets envoyés et les calculs effectués pour calculer le meilleur chemin en prend un schéma de réseau sur lequel on peut trouver des informations redondantes qu'on peut éliminer. Pour cela nous étudiant le schéma de la figure suivante :

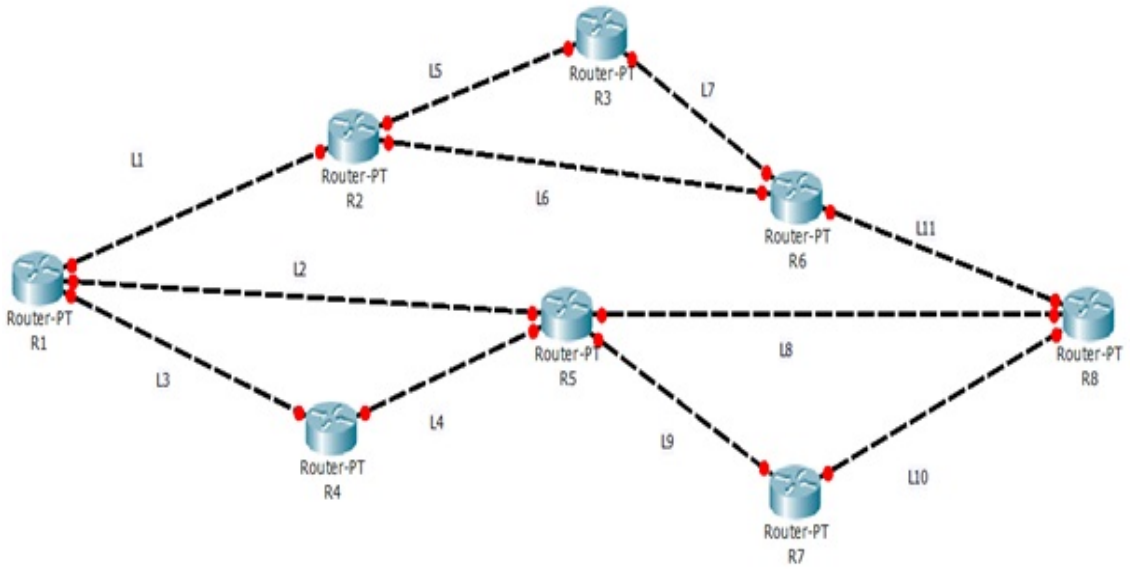

FIGURE 3.2 – Exemple illustrant l'architecture d'un réseau.

On remarque sur le graphe de ce réseau que l'état d'un lien est représenté par les états des interfaces de deux routeurs. Donc, on peut représenter l'état d'un lien par l'état d'une des deux interfaces du premier routeur ou du deuxième routeur.

Le tableau suivant représente les liens existants entre les nœuds du réseau schématisé dans la figure précédente

On remarque d'après le tableau 3.2 qu'il y a une symétrie par rapport à la diagonale principale car le lien $L_{ij} = (R_i, R_j) = L_{ji} = (R_j, R_i)$. Donc, on peut utiliser pour le calcul soit la partie supérieure ou inférieure de la matrice.

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**
R1	/	**L1**	/	**L3**	**L2**	/	/	/
R2	**L1**	/	**L5**	/	/	**L6**	/	/
R3	/	**L5**	/	/	/	**L7**	/	/
R4	**L3**	/	/	/	**L4**	/	/	/
R5	**L2**	/	/	**L4**	/	/	**L9**	**L8**
R6	/	**L6**	**L7**	/	/	/	/	**L11**
R7	/	/	/	/	**L9**	/	/	**L10**
R8	/	/	/	/	**L8**	**L11**	**L10**	/

TABLE 3.2 – Liens entre chaque nœud du réseau.

3.6 Conclusion

Dans ce chapitre nous avons présenté toutes les étapes du fonctionnement de l'algorithme de surveillance des interface d'un routeur et son model mathématique sur lequel il se base pour prendre la décision de l'acheminement des paquets.

Le Protocole que nous avons proposé peut être appliqué dans plusieurs systèmes. On prend l'exemple d'une ambulance qui cherche à trouver le meilleur chemin dans une ville pour atteindre un point d'incendie ou un point d'accident routier.

CHAPITRE 4

SIMULATION DU PROTOCOL DE ROUTAGE (PSCLIR)

4.1 Introduction

Après avoir détaillé le fonctionnement de notre protocole, nous présentons dans ce que suit les résultats de simulation et une interprétation détaillé de l'application réalisée. Pour cela, nous commençons par la présentation de l'environnement de développement de l'application puis on expliquera les fonctionnalités du programme et ses interfaces enfin, on termine par une conclusion.

4.2 Présentation de l'environnement de développement

4.2.1 NetBeans

NetBeans est à l'origine un EDI Java. NetBeans fut développé à l'origine par une équipe d'étudiants à Prague, racheté ensuite par Sun Microsystems. Quelque part en 2002, Sun a décidé de rendre NetBeans open-source. Mais NetBeans n'est pas uniquement un EDI Java. C'est également une plateforme, vous permettant d'écrire vos propres applications Swing. Sa conception est complètement modulaire : Tout est module, même la plateforme. Ce qui fait de NetBeans une boite à outils facilement améliorable ou modifiable. Elle permet de développer tout types d'applications basées sur la plateforme La License de NetBeans permet de l'utiliser gratuitement à des fins commerciales ou non. Les modules que vous pourriez écrire peuvent être open-source comme ils peuvent être closed-source, Ils peuvent être gratuits, comme ils peuvent être payants.

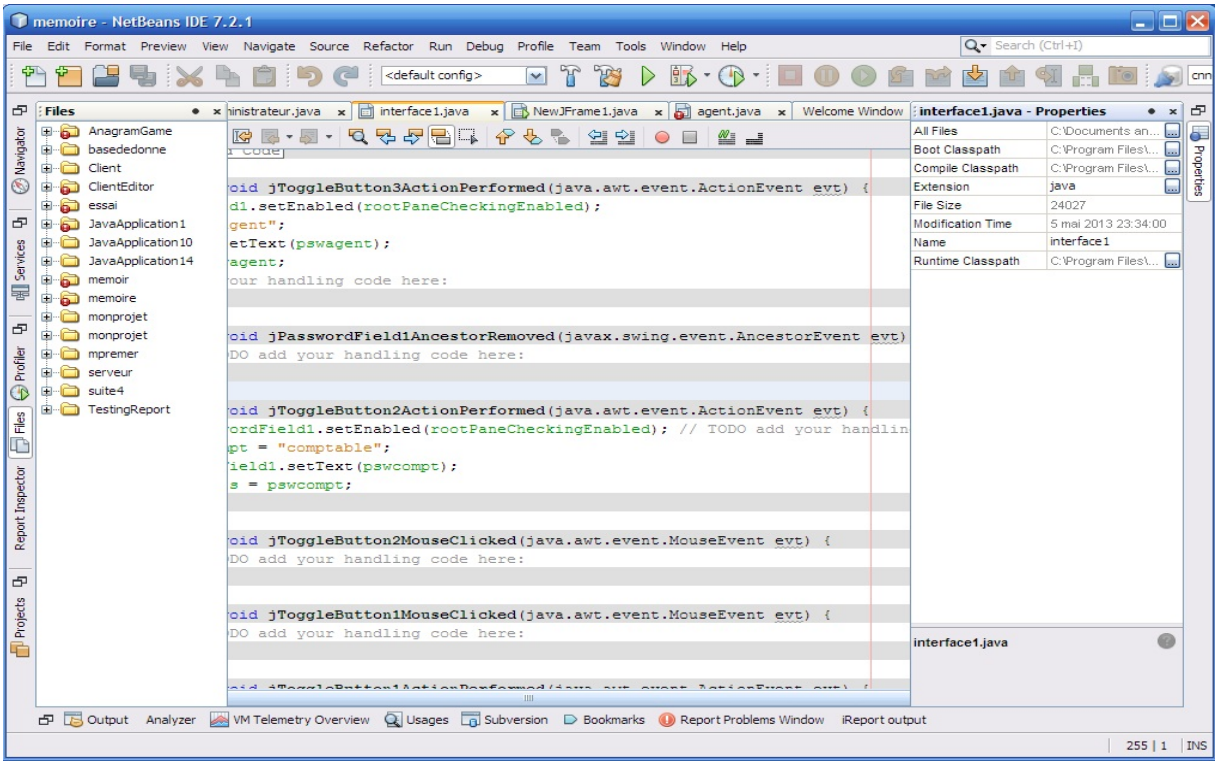

FIGURE 4.1 – Interface de l'IDE netBeans.

4.3 Langage de programmation

Le langage de programmation que nous avons utilisée pour le développement de l'application est le langage JAVA.

4.3.1 Historique du langage

On peut faire remonter la naissance de Java à 1991. À cette époque, des ingénieurs de chez SUN ont cherché à concevoir un langage applicable à de petits appareils électriques (on parle de code embarqué). Pour ce faire, ils se sont fondés sur une syntaxe très proche de celle de C++, en reprenant le concept de machine virtuelle déjà exploité auparavant par le Pascal UCSD. L'idée consistait à traduire d'abord un programme source, non pas directement en Langage machine, mais dans un pseudo langage universel, disposant des fonctionnalités communes à toutes les machines. Ce code intermédiaire, dont on dit qu'il est formé de bytecodes1, se trouve ainsi compact et portable sur n'importe quelle machine ; il suffit simplement que cette dernière dispose d'un programme approprié (on parle alors de machine virtuelle) permettant de l'interpréter dans le langage de la machine concernée [29].

4.3.2 Présentation du langage JAVA

Java est un langage objet permettant le développement d'applications complètes s'appuyant sur les structures de données classiques (tableaux, fichiers) et utilisant abondamment l'allocation dynamique de mémoire pour créer des objets en mémoire. La notion de structure, ensemble de données décrivant une entité (un objet en Java) est remplacée par la notion de classe au sens de la programmation objet. Le langage Java permet également la définition d'interfaces graphiques (GUI : Graphical User Interface) facilitant le développement d'applications interactives et permettant à l'utilisateur de "piloter" son programme dans un ordre non imposé par le logiciel. Le langage est aussi très connu pour son interactivité sur le Web facilitant l'insertion dans des pages Web, au milieu d'images et de textes, de programmes interactifs appelés "applets". Pour des problèmes de sécurité, ces applets sont contrôlées et souvent limitées dans leur interaction avec le système d'exploitation de l'ordinateur sur lequel elles se déroulent : limitation des accès aux fichiers locaux ou aux appels système de la machine. Un programme Java est portable au sens où il peut s'exécuter sur des ordinateurs fonctionnant avec différents systèmes d'exploitation. Les programmes écrits en Pascal ou en langage C sont aussi portables par compilation du code source sur la machine où le programme doit s'exécuter. Java est portable d'une plate-forme produit un langage intermédiaire appelé "bytecode" qui est interprété sur les différentes machines. Il suffit donc de communiquer le bytecode et de disposer d'un interpréteur de bytecode pour obtenir l'exécution d'un programme Java. Les navigateurs du Web ont intégré un interpréteur de bytecode qui leur permet d'exécuter des programmes (applets) Java [30].

4.4 Architecture de l'application

Pour la présentation de la simulation du fonctionnement de notre protocole de routage, nous avons présenté le cas d'un nœud qui a trois interfaces reliées à des liens de communication. Nous avons présenté toutes les étapes de déroulement du protocole.

4.4.1 Enregistrement de l'historique

Pour la simulation de connexion et de déconnexion des interfaces, nous avons créé trois processus (thread) qui s'exécutent en parallèle et qui attribuent aléatoirement des valeurs 1 ou 0 à une variable qui désigne l'état connecté ou non du lien correspondant. Ensuite, on a créé un autre processus qui analyse l'état de la variable (x) pour chaque processus et enregistre cet état dans une table. Une représentation graphique est associée

pour chaque processus pour illustrer le changement aléatoire de connexion et déconnexion des interfaces. La figure 4.2 représente la première interface de l'application

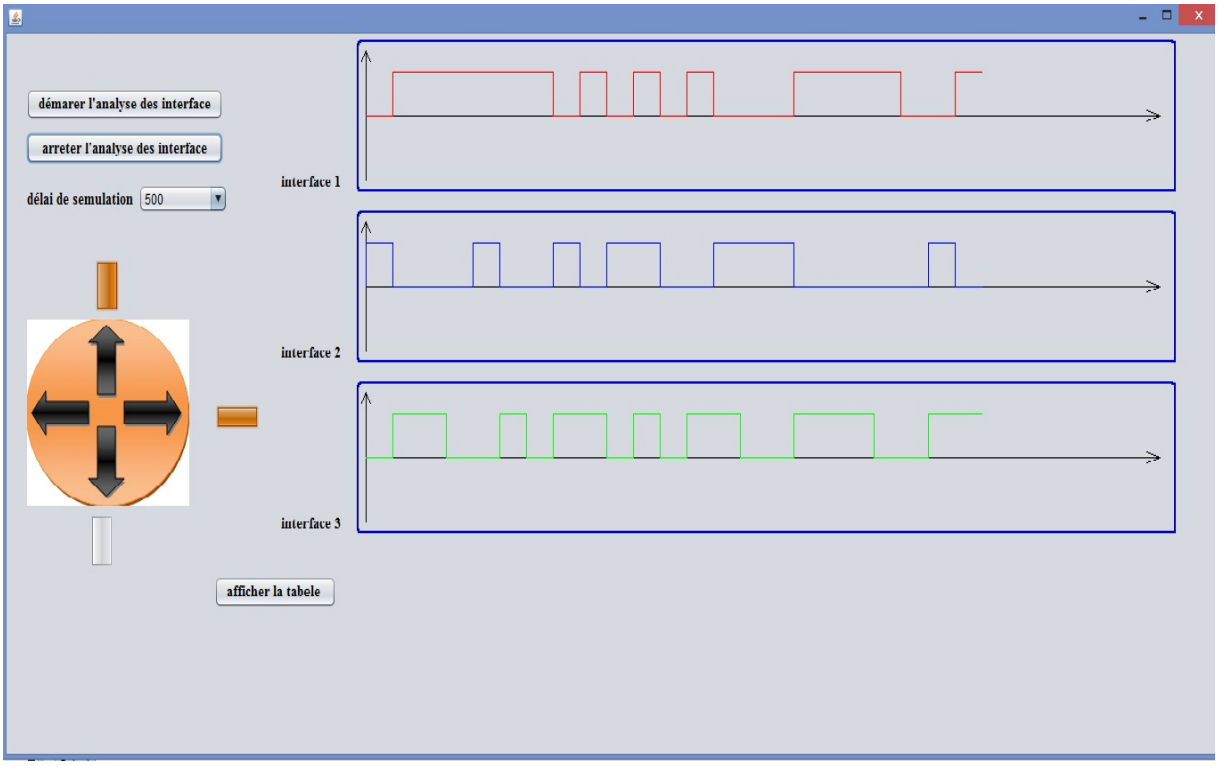

FIGURE 4.2 – Première Interface de l'application.

Au lancement de l'application cette interface apparait.
Pour commencer la simulation, on clique sur le bouton <démarrer l'analyse des interfaces>, les graphes des interfaces commenceront à se dessiner sur les repères. Pour changer la vitesse de simulation, on change le délai de simulation. Et pour arrêter l'analyse on click sur bouton «arrêter l'analyse des interfaces».
Pour afficher les états des interfaces enregistrées, on click sur bouton «afficher la table». L'interface de calcul du temps de connexion et déconnexion s'affiche et on trouve afficher à côté la table d'historique enregistrée.

4.4.2 Calcul du temps de connexion et déconnexion

Après enregistrement de l'historique, on passe au calcul des temps de connexion et déconnexion sur l'interface. Le protocole consulte la table de données sur l'historique enregistré pour calculer les temps de connexion et déconnexion. On clique sur le bouton « Représenter le temps de connexion et déconnexion » un calcul est effectuée puis une représentation graphique est affichée sur les repères.

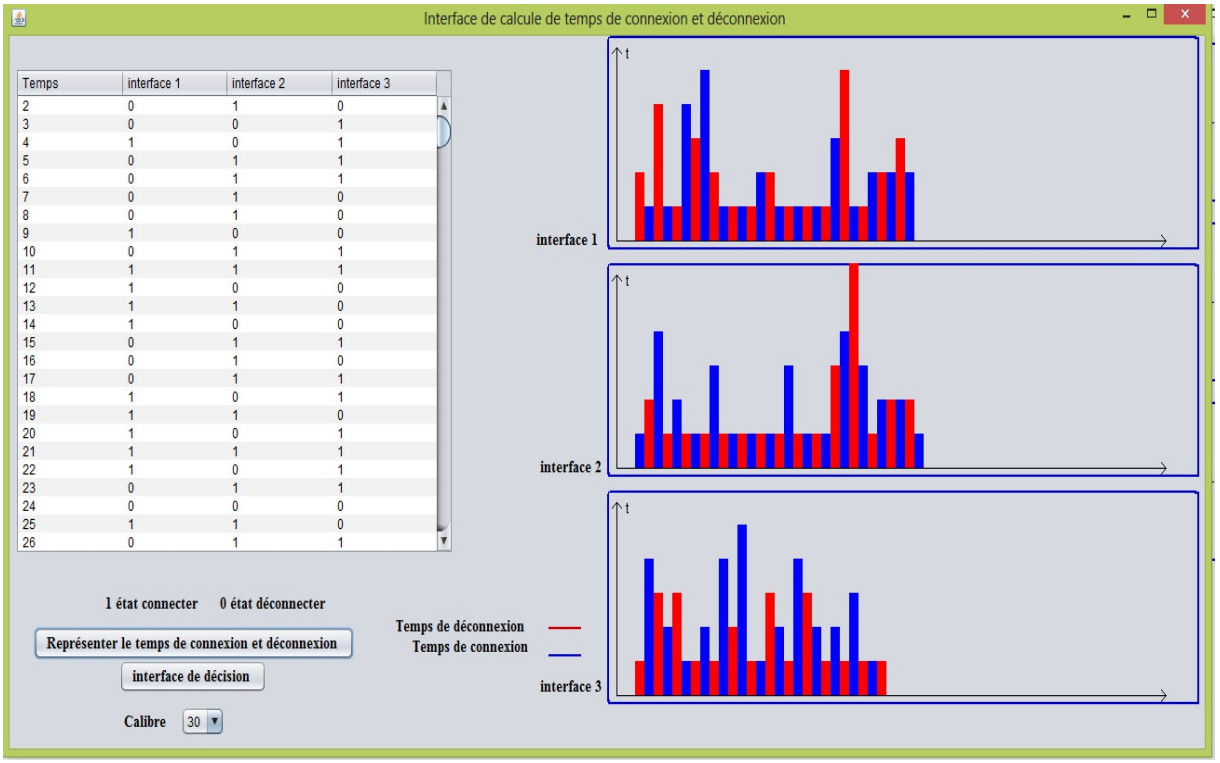

FIGURE 4.3 – Interface de calcul du temps de connexion et déconnexion.

4.4.3 Décision de l'acheminement du paquet

Pour que le programme calcule le lien qui convient lors de l'arrivée de l'un des paquets sur une interfaces et affiche la décision prise, on doit considérer la période $\Delta T = T2 - T1$ dans laquelle le paquet arrivant peut séjourner. On clique sur le bouton «Représenter les temps» et une représentation du temps de connexion et déconnexion pour chaque interface est affichée dans la zone du graphe est illustrée sur la figure 4.4.

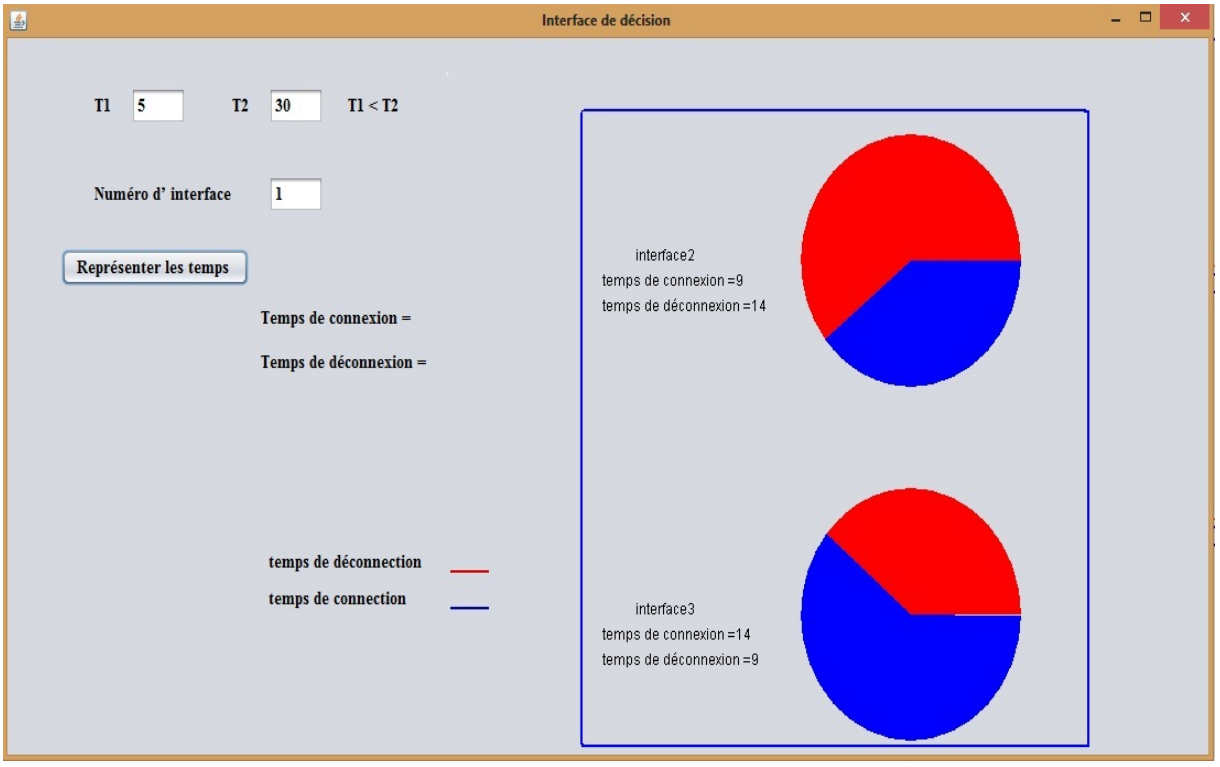

FIGURE 4.4 – Interface de Décision de l'acheminement du paquet.

4.4.4 Conclusion

Dans ce présent chapitre, nous avons présenté l'environnement du travail ainsi le langage utilisé pour simuler le fonctionnement du protocole « PSCLIR » et les résultats de simulation obtenus par notre application.

CONCLUSION GÉNÉRALE

La recherche dans le domaine des réseaux sans fil est en plein essor. Plusieurs protocoles de routage ont été développés ces dernières années. Dans ce mémoire on a passé en revu une étude sur les type de réseaux sans fil existant et une étude comparative sur quelques algorithmes de routage, dans le but de comprendre les étapes à suivre pour implémenter un protocole de routage et les contraintes auxquelles on doit faire face lors de la conception de notre protocole.

Le travail que nous avons effectué (simulation de notre protocole), nous a permis de voir la possibilité d'application de ce protocole dans la réalité.

Grâce à ce projet, nous avons acquis beaucoup de savoirs dont nous citons les plus importants :

- Enrichissement de nos connaissances en réseau informatique.
- Avoir un plan sur lequel on peut se baser pour implémenter un protocole de routage.
- Enrichissement de nos connaissances en ce que concernent le fonctionnement des protocoles de routage et leurs implémentations.
- Enrichissement de nos connaissances et de nous compétence en programmation java (Graphic 2D les threads).
- Enrichissement de nos connaissances dans le domaine de programmation des system distribué.

Et comme perspective, il y a lieu d'améliorer les performances de ce protocole, d'ajouter des fonctionnalités si nécessaire, de le tester dans la réalité et de l'implémenter pour assurer sa validité. Enfin nous souhaitons que cette modeste étude contribue à l'exploitation de ce protocole pour améliorer d'autres protocoles de routages dans d'autres types de réseaux différents.

BIBLIOGRAPHIE

[1] M.Khelifi et A.Djabelkhir, École Doctorale en Informatique ReSyD Bejaïa, Algérie.

[2] K. A. Agha et G.Pujolle, Réseaux mobiles et réseaux sans fil, Eyrolles 2002.

[3] D. DHOUTAUT, Etude du standard IEEE 802.11 dans le cadre des réseaux Ad Hoc Thèse de doctorat, L'Institut National des Sciences Appliquées de Lyon, Décembre 2003.

[4] R. Bedouhene et M. Benmedour, Protocole de Connexion des Réseaux Ad Hoc à Internet, Mémoire Magister, université des sciences et de la technologie Houari Boumediene 2004.

[5] N. BOUKHECHEM, routage dans les réseaux mobiles Ad Hoc par une approche a base d'agents, Mémoire Magister, Faculté des sciences et science de l'ingénieur, Université de Constantine 2008.

[6] S. BoukliHacene, Qualité de service, Thèse de Doctorat, Université Djillali Liabes 2012.

[7] F. Brissaud, D. Charpentier, A. Barros et C. Bérenguer. capteur intelligents : nouvelles technologies et nouvelles problématiques pour la Sûreté de fonctionnement, Maîtrise des Risques et de Sûreté de Fonctionnement, Lambda-Mu 16, Avigon , France 2008.

[8] M. Badet et W. Bonneau, Mise en place d'une plateforme de test et d'expérimentation, Projet tutoré (1ière Master Technologie de l'Internet), Mémoire de Master, Université Pau et des pays de l'Adour 2006.

[9] A. Gallais, F. Ingelrest, J. Carle et S.R.David. Maintien de la couverture de surface dans les réseaux de capteurs avec une couche physique non idéale. CFIP, Colloque Francophone sur l'Ingénierie des Protocoles 2006.

[10] A. Gallais. Ordonnancement d'activité dans les réseaux de capteurs : l'exemple de la couverture de surface. Université des sciences et technologies de Lille 2007.

[11] B. Kamal, Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs, Thèse, Université de Franche Comte, 16 décembre 2009.

[12] Y. Younes, Minimisation d'énergie dans un réseau de capteurs, Mémoire de Master, Département d'Informatique, Université Mouloud Mammeri de Tizi-Ouzou, Septembre 2012.
[13] Dunkels, B. Grönvall, et T. Voigt. Contiki: a Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the First IEEE Workshop on Embedded Networked Sensors, Florida, USA 2004.
[14] Abelmajid HAJAMI. Sécurité du routage dans les réseaux sans fil spontanés. Thèse de Doctorat, Ecole Nationale Supérieure d'Informatique, Rabat.
[15] I.F. Akyildiz, W.S. Sankarasubramaniam et E. Cayirci, Wireless Sensor Network: A Survey. Computer networks 2002.
[16] C. Chong et Y. Kumar, Sensor networks: evolution, opportunities, and challenges. Proceedings of the IEEE 2003.
[17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. l. Cayirci, A survey on sensor networks, IEEE Communications Magazine, Août 2002.
[18] Andrews, P. Johnson et D.C, Remote continuous monitoring in the home. Telemedicine and Telecare 2006.
[19] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis et V.Z. Groza. Sensor-based information appliances. IEEE Instrumentation Measurement Magazine December 2000.
[20] M. lehsaini, Diffusion et couverture basées sur le clustering dans les réseaux de capteurs application à la domotique, Thèse de Doctorat, Université de Tlemcen 2009.
[21] Asma BEN MESSAOUD. Classification des protocoles de routages dans les réseaux DTN. 2009.
[22] T. clausen et P. jacquet. Optimized Link State Routing Protocol (OLSR).
RFC 3626 (Experimental). Internet Engineering Task Force, oct 2003.
[23] Q. Amir, V. Laurent et L. Anis. Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks. Anglais. Rapport de recherche RR-3898. INRIA 2000.
[24] A. Neumann, Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.) draft-wunderlich-openmesh-manet-routing. Draft RFC (Experimental). Internet Engineering Task Force, avr 2008.
[25] Charles E. Perkins et B. Pravin. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers: Proceedings of the conference on Communications architectures, protocols and applications. SIGCOMM 94. London, United Kingdom, ACM 1994.

[26] C. perkins, E. belding-royer et S. das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561 (Experimental). Internet Engineering Task Force, juil. 2003.

[27] D. johnson, Y. hu et D. maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental). Internet Engineering Task Force, fév. 2007.

[28] Q. Amir, V. Laurent et L. Anis. Multipoint Relaying : An Efficient Technique for Flooding in Mobile Wireless Networks.Anglais. Rapport de recherche RR-3898. INRIA, 2000.

[29] C. delannoy eyroles .Programmer en Java.5 ème edition,

[30] Michel Divay .la programmation objet en java. Dunod. 2006

Résumé

Les protocoles de routage assurent la connectivité du réseau et maintiennent des routes afin que les données envoyé par une source puissent atteindre leur destination. Parmi les protocoles de routage utilisé actuellement dans l'internet on peut citer : BGP, RIP et OSPF. Cependant avec la croissance important de l'internet un nombre important des entreprises et service public sont devenu dépondant du bon fonctionnement de ces protocoles. Nous présentons dans ce mémoire une proposition d'un protocole de routage dynamique que se base sur la probabilité de libération de lien et la simulation de fonctionnement de ce dernier.

Mots clés : réseau ad hoc, routage, protocole.

Abstract

Routing protocols provide network connectivity and maintain roads so that the data sent by a source to reach their destination. Among the routing protocols currently used in the Internet include : BGP, RIP, and ospf. However, with the significant growth of the Internet a large number of companies and public service have become dependent the proper functioning of these protocols. We present in this paper a proposal of a dynamic routing protocol that is based on the connection release probability and simulation of operation of the latter.

Key words : ad hoc network, routing , protocol.