**République Algérienne Démocratique et Populaire**
**Ministère l'Enseignement Supérieur et de la Recherche Scientifique**
**Université Hadj Lakhdar BATNA**
**Faculté des Sciences de l'Ingénieur**
**Département Informatique**

# *Etude et analyse de la stabilité des protocoles de routage dans les réseaux ad-hoc*

***Mémoire présenté en vue de l'obtention du Diplôme de Magister***
***en Informatique***
***Option : Informatique Industrielle***

***Présenté par :***
***Kamil CHEBIRA***

***Sous la direction de :***
***Pr. Noureddine DOGHMANE***

***Composition du Jury :***

| | | | |
|---|---|---|---|
| *Dr. ZIDANI Abdelmadjid* | *Maître de conférence* | *Université Batna* | *Président* |
| *Pr. DOGHMANE Noureddine* | *Professeur* | *Université Annaba* | *Encadreur* |
| *Dr. BILAMI Azzedine* | *Maître de conférence* | *Université Batna* | *Examinateur* |
| *Dr. FARAH Nadir* | *Maître de conférence* | *Université Annaba* | *Examinateur* |

***Année universitaire 2006 - 2007***

# Dédicace et remerciements

*Ce modeste travail n'est ni plus ni moins le fruit de parents qui ont su comment faire aboutir ce projet de magister. Ils m'ont toujours encouragé et orienté vers la réussite. Je ne les remercierai jamais assez.*

*A mon encadreur, Mr N. DOGHMANE qui a toujours été présent pour m'éclairer le chemin et qui n'a pas hésité avec ses conseils constructifs.*

*A mon frère Mounis, mes sœurs May et Lydia*

*A mon épouse Lilia et ma fille Yara*

*A mes neveux Macil et Selyane*

*A toute ma famille*

*A tous ceux qui ont participé à l'élaboration de ce travail et qui sauront se reconnaître*

*Je dédie le cœur de cette thèse de magister.*

*Kamil CHEBIRA.*

# Table des matières

## Liste des tableaux

## Liste des graphiques

## Liste des acronymes

- AODV : ad hoc On-Demande Distance-Vector
- BTS : Base Tranceiver Station
- Bluetooth : Technologie de réseau personnel sans fils (noté WPAN pour Wireless Personal Area Network),
- Broadcast Diffusion
- CGSR : Clusterhead Gateway Switch Routing
- CLR Clear packet
- CMU : Carnegie Mellon University
- CSMA/CA : Carrier Sense, Multiple Access/Collision Avoidance
- CSMA/CD : Carrier Sense Multiple Access / Collision Detect
- DAG : Graphe Dirigé Acyclique (Directed Acyclic Graph)
- DECT : Digital Enhanced Cordless Technology
- DSDV : Destination Sequenced Distance-Vector Routing
- DSR : Dynamic Source Routing
- EDGE : Enhanced Data rate for GSM Evolution
- Flooding : Inondation
- GHz Giga Hertz
- GPRS : General Packet Radio Service
- GSM : Global System for Mobile Communications
- HiperLAN : High Performance Radio LAN
- HomeRF : Spécification de réseau sans fil (Shared Wireless Access Protocol-SWAP)
- IEEE : Institute of Electrical and Electronics Engineers
- IEEE 802.11 : Standard pour les réseaux locaux sans fil sans infrastructure
- IETF : The Internet Engineering Task Force
- IMEP : Internet MANET Encapsulation Protocole
- iMode : Internet mode
- INPL : Institut National Polytechnique de Lorraine (Nancy - France)
- IrDA : InfraRed Data Association
- LBNL : Laboratoire National de Lawrence Berkeley
- LSP : Link State Packets
- MAC : Medium Access Control
- MPR Multi-Points Relais
- MRL : Message Retransmission List (Liste des Messages à Retransmettre).
- Mbps : Mega bit pas seconde
- m/h : mètres pas heure

- m/s : Mètres pas seconde
- N.S : Simulateur de réseaux, une collaboration de UC Barcley, LBL, USC/ISI et Xerix PARC.
- NAM : Network Animator
- OLSR : Optimized Link State Routing
- OPNET : Open NETwork
- OSPF : Open Shortest Path First
- Otcl : Object Tool Command Language
- PAN : Personal Area Network
- Python : Langage de programmation interprété, multi-paradigme
- QRY Demande de la source
- R&D Recherches et développements
- RIP : Routing Information Protocol
- Rrep : Route Reply
- Rreq : Route Request
- SIG : Special Interest Group (Microsoft, IBM et Nokia)
- TC : Topology Control
- TCL/TK : Tool Command Language, TK: Toolkit (La bibliothèque graphique de TCL).
- TCP/IP : Transmission Control Protocol/Internet Protocol
- TORA :Tomporally Ordered Routing Algorithme
- Trace : Fichier journal obtenu en sortie (résultat de la simulation).
- UMTS : Universal Mobile Telecomunications System
- VINT :Virtual InterNetwork Testbed
- WAP : Wireless Application Protocol
- WECA :Wireless Ethernet Compatibility Alliance
- WLAN : Wireless Local Area Network appelé aussi WiFi
- WPAN : Wireless Personal Area Network
- WRP : Wireless Routing Protocol
- WiFi : Wireless Fidelity
- ZRP : Zone Routing Protocol

# Introduction

# Introduction

Les protocoles de routage dans les réseaux informatiques sont divers et variés en fonction des types de réseaux. Dans notre étude nous allons nous intéresser particulièrement aux protocoles de routage au sein des réseaux sans fils Ad hoc.

Ce type de réseaux appelé Ad hoc ne dispose guère d'infrastructure fixe comme le cas des réseaux GSM qui nécessitent une installation d'antennes appelées BTS (Base Transceiver Station).

Le support de transmission est les ondes radio. Les nœuds émetteurs disposent d'un rayon de propagation qui couvre leur voisinage.

Ces caractéristiques offrent à ces réseaux plus de mobilité et un déploiement très rapide.

Par conséquent, le rafraîchissement des tables de routage pénalise la bande passante par les paquets de contrôles fréquents.

Les performances d'un protocole de routage se déterminent en fonction du taux de rafraîchissement, des temps de réponse ainsi que le niveau de saturation du ''médium'' ou milieu de transmission. Pour améliorer ces performances, les équipes de recherches simulent différents algorithmes de routage en variant leurs paramètres de configuration et analysent le comportement de chacun dans plusieurs scénarios.

# 1. Description du projet

Les protocoles de routage ad hoc existants ne semblent pas très différents. Des études poussées peuvent nous révéler le contraire. Pour accentuer les petits écarts, la simulation présente un outil essentiel, rapide et peu coûteux.

Le but de notre travail est de distinguer les protocoles de routage à partir de leur algorithme afin de désigner le plus performant. La simulation, notre outil de comparaison, nous offre un grand nombre d'avantages. Elle permet de faire vivre un scénario réel d'un réseau informatique en virtuel, c'est à dire dans la mémoire d'un ordinateur. Le temps d'une simulation réelle est divisé par milliers pour avoir le même scénario en virtuel. Ce grand avantage nous laisse la possibilité de multiplier nos tests et faire un grand nombre de simulations. Nous évaluons les résultats par des fonctions mathématiques qui nous rapprochent beaucoup des résultats réels. Si chacune des simulations s'intéresse à une propriété ou à un paramètre de l'algorithme de routage, nous serons en mesure de définir le bon algorithme et voire même ouvrir des voies pour l'amélioration des autres.

Afin d'avoir un grand angle de vision pour la comparaison des protocoles, nous avons pensé à simuler avec différents paramètres. Un paramètre peut bien être à la faveur d'un algorithme mais pas pour un autre.
Les paramètres à varier dans notre étude sont classés en deux principales catégories, physiques et logiques.
Les paramètres physiques jouent principalement sur la topologie de l'environnement au sein du réseau, contrairement aux paramètres logiques, qui configurent la bande passante et le trafic de données.
Cette étude s'intéresse aux protocoles de routage ad hoc les plus connus, le réactif DSDV et les réactifs AODV et DSR. Le simulateur de réseaux NS (Network Simulator)[1] [1], [2], [3] sera utilisé comme outil de simulation durant tout notre travail.

## 2. Objectifs du projet

Notre objectif est de simuler des protocoles de routage ad hoc avec le simulateur de réseaux NS pour analyser leurs performances dans une multitude de scénarios, où des topologies et des flux sont à définir. Une fois les simulations terminées et leurs traces[2] générées, nous les analyserons suivant différents critères et générerons des courbes qui présentent les synthèses finales.

## 3. Organisation du rapport

Le premier chapitre comprend une présentation des réseaux informatiques, sans fils et ad hoc, leurs caractéristiques, leurs protocoles de routage et leurs propriétés. Le second chapitre est consacré au cadre expérimental où nous exposons le simulateur de réseaux. Nous définissons les paramètres qui nous ont servis pour les simulations au troisième chapitre. Le quatrième présente l'étude et calcules de simulations avec toutes les variables utilisées. Enfin, les analyses des résultats couvrent le chapitre cinq qui est prend fin par des conclusions et perspectives.
Deux annexes sont introduites à la fin du document afin de mieux schématiser l'environnement de simulation.

[1] N.S : Simulateur de réseaux, une collaboration de UC Barcley, LBL, USC/ISI et Xerix PARC.
[2] Trace : Fichier journal obtenu en sortie (résultat de la simulation).

# Chapitre 1.
# Les réseaux sans fil

Le développement technologique au cours de ces dernières décennies a révolutionné un certain nombre de domaines, notamment celui des réseaux informatiques et plus spécialement les réseaux sans fil. Les performances ne cessent d'augmenter et les prix de chuter. Cette avancée les a rendu abordable par le grand public. Ce marché de l'équipement sans fil est actuellement en plein essor. Le constructeur de la puce Bluetooth avait annoncé le chiffre de 100 millions d'équipements électroniques Bluetooth en 2002.

Cette révolution des réseaux informatiques devrait se poursuivre au vu des coûts des investissements dans le câblage de plus en plus élevés.
Le sans fil a touché beaucoup de domaines mis à part les réseaux informatiques, par exemple, la téléphonie mobile où beaucoup de couches de transports ont été développées. Nous citerons quelques exemples comme (GSM[3], GPRS[4], UMTS[5], EDGE[6]). Dans notre travail nous allons nous intéresser principalement aux réseaux sans fil informatiques.
Voici quelques définitions des principaux types de réseaux sans fil existant actuellement.

## 1.1 Réseaux sans fil téléphonique (Couches de transport)

- **GSM (Global System for Mobile Communications)**

Norme mondiale la plus répandue de téléphonie cellulaire numérique qui exploite les fréquences 900 et 1800 Mhz dans plusieurs pays du monde et la 1900 Mhz en Amérique du Nord.

- **GPRS (General Packet Radio Service)**

Amélioration du GSM pour supporter les transferts de données.

- **UMTS (Universal Mobile Telecomunications System)**

La 3éme génération introduite comme la révolution des applications mobiles où elle assure aussi le transfert de sons, images et vidéos en haut débit (de 384 Kbps à 2 Mbps)

- **EDGE (Enhanced Data rate for GSM Evolution)**

Proposé par l'opérateur téléphonique Bouygues Télécom alternative à l'UMTS avec un débit maximum de 384 Kbps.

[3] GSM : Global System for Mobile Communications
[4] GPRS : General Packet Radio Service
[5] UMTS : Universal Mobile Telecommunications System
[6] EDGE : Enhanced Data rate for GSM Evolution

Les couches citées ci-dessus sont utilisées par quelques applications telles que WAP[7], iMode[8].

- **WAP (Wireless Application Protocol)**

Introduit en 1997 dans le but de permettre aux téléphones portables de se connecter au Web par les collaborateurs NOKIA, ERICSSON, MOTOROLA et UNWIRED PLANET. Cette initiative devait permettre la consultation d'e.mails, l'accès aux réseaux locaux d'entreprises et beaucoup d'autres services. Mais les aspects techniques étaient limités et loin de l'image marketing annoncée pour ce protocole.

- **iMode (Internet Mode)**

Technologie développée par l'opérateur japonais NTT-DoCoMo et introduite en France par l'opérateur téléphonique Bouygues Télécom. iMode utilise la couche basse GPRS, et réalise de ce fait une économie considérable par rapport au WAP. Il garde la connexion juste au moment du transfert des paquets contrairement au WAP qui facture le temps de la consultation mais pas le volume des données transférées.

## 1.2 Réseaux sans fil informatiques

Les autres réseaux qui nous intéressent particulièrement sont : Bluetooth[9], HomeRF[10], HiperLAN[11] et WiFi[12].

- **Bluetooth (Internet Mode)**

Cette technologie gère les connexions sans fil de type ondes radio utilisant les fréquences 2,4 Ghz d'un débit de 1Mbps et d'une portée de 10m à 30m. Cette technologie concurrence fortement l'infrarouge IrDA (InfraRed Data Association).
Le nom Bluetooth a été inspiré en 1994 par ses créateurs (ERICSSON, IBM, INTEL, NOKIA et TOSHIBA) du nom de Harald Blaatand (910-986), littéralement « Harald à la dent bleue » qui unifia le Danemark et la Norvège, dans une Europe divisée.
Le Bluetooth Special Interest Group (SIG) qui compte notamment Microsoft, IBM et Nokia comme membres vient d'adopter en novembre 2004, les spécifications de la norme Bluetooth 2.0+ Enhanced Data Rate. Offrant des débits théoriques de l'ordre du 3 Mbps, la future norme a fait son apparition début 2005 dans les magasins.
L'organisme a révélé quelques unes des principales nouveautés de son nouveau standard. Outre l'augmentation du débit qui va donc passer de 1 Mbps à 3 voire 10

[7] WAP : Wireless Application Protocol
[8] iMode : Internet mode
[9] Bluetooth :
[10] HomeRF :
[11] HiperLAN :
[12] WiFi :

Mbps dans certains cas, la consommation devrait diminuer de l'ordre de 50% environ. Le taux d'erreurs sur bits, donnée caractérisant la fiabilité des communications, sera également en baisse. Ce lot d'innovation s'accompagne d'une compatibilité garantie avec les versions antérieures du Bluetooth.

- **HomeRF**

Norme qui devrait servir à banaliser la mise en place des réseaux locaux domestiques sans fil. Elle est inspirée des deux normes DECT[13] (pour la transmission de la voix) et WLAN[14] (pour la transmission de données TCP/IP)

HomeRF assure un traitement allant jusqu'à 127 nœuds et 6 liaisons voix.

Dans la pratique, la portée d'une base est de 50 mètres, ce qui est censé couvrir une maison moyenne avec son jardin. À l'opposé, un réseau Bluetooth définit un PAN (Personal Area Network), réseau radio de très courte portée, d'environ 10 mètres. Le débit brut radio d'une base HomeRF est de 1,6 Mbps et de 1 Mbps utile en IP, 800 Kbps en tenant compte des pertes et des interférences (une évolution vers un débit de 10 Mbps et 2 Mbps utiles est prévue, mais les matériels sont encore en phase de développement).

En matière de réseaux domestiques sans fil, HomeRF doit affronter la concurrence de Bluetooth. La norme est en effet promue par plus de 1 000 industriels contre 90 pour HomeRF. En outre, le groupe de travail 802. 15 de l'IEEE vient de se former pour standardiser une norme de réseau personnel PAN (Personal Area Network) sans fil reposant sur la technique Bluetooth. HomeRF a cependant pris de l'avance, puisque les premiers matériels sont déjà disponibles sur le marché. " *Bluetooth possède un avantage marketing certain. Mais pour relier tous les appareils d'une maison, il est insuffisant en termes de portée. Bluetooth serait plutôt concurrent des liaisons infrarouges. De plus, les premiers modèles d'appareils à la norme tardent à être commercialisés. Il se pourrait que la première technologie sur le marché réussisse tout de même à l'emporter* ", analyse le responsable des produits sans fil chez France Télécom R&D.

- **HiperLAN**

Spécifié par l'ETSI (European Telecommunications Standards Institute)

Mobilité supportée : piétonne (3m/s max i.e. 10km/h)

Hiperlan émet dans la bande des 5 GHz et permet d'atteindre un débit de 54 Mbps.

HiperLAN possède des avantages techniques, par exemple l'inclusion d'une classe de service lui permettant de gérer la voix et l'émission multimédia en continu. Elle intègre

[13] DECT : Digital Enhanced Cordless Technology
[14] WLAN : Wireless LAN appelé aussi WiFi

également une technique empêchant les interférences avec d'autres équipements radio.

- **WiFi**

La norme IEEE 802.11 (ISO/IEC 8802-11) est un standard international décrivant les caractéristiques d'un réseau local sans fil (WLAN). Le nom WiFi[15] (contraction de Wireless Fidelity, parfois notée Wi-Fi) correspond initialement au nom donné à la certification délivrée par la Wi-Fi Alliance, anciennement WECA (Wireless Ethernet Compatibility Alliance), l'organisme chargé de maintenir l'interopérabilité entre les matériels répondant à la norme 802.11. Par abus de langage (et pour des raisons de marketing) le nom de la norme se confond aujourd'hui avec le nom de la certification. Ainsi un réseau Wifi est en réalité un réseau répondant à la norme 802.11. Le tableau ci-dessous regroupe toutes les nomes 802.11 de la plus ancienne à la plus récente.

[15] WiFi : Wireless Fidelity

| Nom de la norme | Nom | Description |
|---|---|---|
| **802.11a** | Wifi5 | La norme 802.11a (baptisé *WiFi 5*) permet d'obtenir un haut débit (54 Mbps théoriques, 30 Mbps réels). La norme 802.11a spécifie 8 canaux radio dans la bande de fréquence des 5 GHz. |
| **802.11b** | Wifi | La norme 802.11b est la norme la plus répandue actuellement. Elle propose un débit théorique de 11 Mbps (6 Mbps rééls) avec une portée pouvant aller jusqu'à 300 mètres dans un environnement dégagé. La plage de fréquence utilisée est la bande des 2.4 GHz, avec 3 canaux radio disponibles. |
| **802.11c** | Pontage 802.11 vers 802.1d | La norme 802.11c n'a pas d'intérêt pour le grand public. Il s'agit uniquement d'une modification de la norme 802.1d afin de pouvoir établir un pont avec les trames 802.11 (niveau *liaison de données*). |
| **802.11d** | International | La norme 802.11d est un supplément à la norme 802.11 dont le but est de permettre une utilisation internationale des réseaux locaux 802.11. Elle consiste à permettre aux différents équipements d'échanger des informations sur les plages de fréquence et les puissances autorisées dans le pays d'origine du matériel. |
| **802.11e** | Amélioration<br><br>de la qualité de service | La norme 802.11e vise à donner des possibilités en matière de qualité de service au niveau de la couche *liaison de données*. Ainsi cette norme a pour but de définir les besoins des différents paquets en terme de bande passante et de délai de transmission de telle manière à permettre notamment une meilleure transmission de la voix et de la vidéo. |
| **802.11f** | Itinérance (roaming) | La norme 802.11f est une recommandation à l'intention des vendeurs de point d'accès pour une meilleure interopérabilité des produits. Elle propose le protocole *Inter-Access point roaming protocol* permettant à un utilisateur itinérant de changer de point d'accès de façon transparente lors d'un déplacement, quelles que soient les marques des points d'accès présentes dans l'infrastructure réseau. Cette possibilité est appelée *itinérance* (ou *roaming en anglais*) |
| **802.11g** | | La norme 802.11g offre un haut débit (54 Mbps théoriques, 30 Mbps réels) sur la bande de fréquence des 2.4 GHz. La norme 802.11g a une compatibilité ascendante avec la norme 802.11b, ce qui signifie que des matériels conformes à la norme 802.11g pourront fonctionner en 802.11b |
| **802.11h** | | La norme *802.11h* vise à rapprocher la norme 802.11 du standard Européen (HiperLAN 2, d'où le *h* de 802.11h) et être en conformité avec la règlementation européenne en matière de fréquence et d'économie d'énergie. |
| **802.11i** | | La norme *802.11i* a pour but d'améliorer la sécurité des transmissions (gestion et distribution des clés, chiffrement et authentification). Cette norme s'appuie sur l'*AES* (*Advanced Encryption Standard*) et propose un chiffrement des communications pour les transmissions utilisant les technologies 802.11a, 802.11b et 802.11g. |
| **802.11IR** | | La norme *802.11j* a été élaborée de telle manière à utiliser des signaux infra-rouges. Cette norme est désormais dépassée techniquement. |
| **802.11j** | | La norme *802.11j* est à la règlementation japonaise ce que le 802.11h est à la règlementation européenne. |

*Tableau 1. Différentes normes 802.11*

## 1.3 Les réseaux sans fil ad hoc

### Introduction

Un réseau sans fil ad hoc est un réseau sans fil d'entités mobiles liées entre elles par des liaisons à base d'ondes radio sans infrastructure fixe ni administration centralisée. Chaque nœud joue le rôle d'hôte ou de routeur à un instant donné. L'interconnexion de tous les nœuds mobiles forme une topologie temporaire dynamique qui se déploie aisément.

Par ailleurs, la mobilité des nœuds leur pose différentes contraintes sur les ressources ainsi que leurs capacités. Un nœud mobile peut bouger librement sans prévenir son entourage qui doit détecter l'absence de ce dernier et surtout maintenir une vue cohérente de l'environnement à chaque instant.

Les protocoles de routage ad hoc doivent acheminer les paquets à leurs destinations indépendamment des changements de topologie. Pour répondre à ces exigences, le protocole doit converger rapidement vers la stabilité pour assurer la résolution des ruptures de liaisons qui sont dues à la mobilité et interrompent le trafic.

La représentation des réseaux se fait classiquement sous forme de petits cercles pour les nœuds et des segments de droite pour les liaisons entre ces nœuds. Pour les réseaux ad hoc, la représentation peut être différente du fait qu'il n'y a plus de liaisons câblées. Alors le segment est remplacé par un cercle concentré sur le nœud pour représenter la zone de connexion du nœud ou en d'autres termes son rayon de propagation.

La figure 1.1 représente un simple réseau ad hoc.

Sur cette figure, le nœud A ne peut communiquer qu'avec le nœud B et le nœud C ne peut communiquer qu'avec B à cause de leur limitation du rayon de propagation. Par contre B peut communiquer avec les deux, ce qui lui attribuera la fonction de routeur qui achemine le trafic entre les deux hôtes.

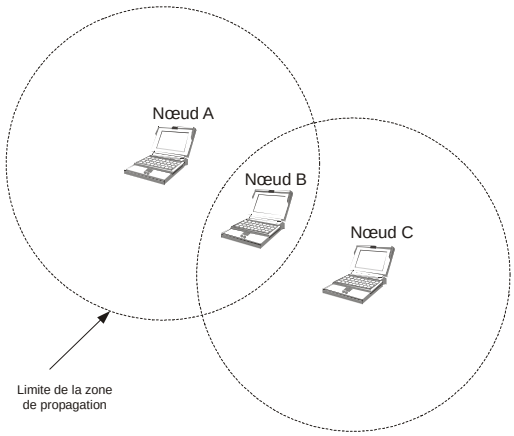



*FIG 1.1 - Simple réseau ad hoc.*

Dans ce type précis de réseaux ad hoc que nous étudions, les nœuds sont dotés d'une antenne radio leurs permettant d'émettre leurs signaux et de réceptionner ceux émis par leurs voisins. Pour chaque nœud, est définie une zone de propagation quand il est entrain d'émettre. Tout nœud se trouvant sur cette zone est appelé un voisin et il ne peut émettre au même moment, parce que la fréquence utilisée est unique pour tous les nœuds. Les interférences empêchent les émetteurs d'émettre directement sur le support. Chaque émetteur doit tout d'abord écouter sur le ‘’médium’’ puis émet s'il est libre sinon attendre un moment, ce protocole évite les collisions provoquées par deux émissions simultanées exemple de ce protocole CSMA/CA (Carrier Sense, Multiple Access/Collision Avoidance). Sinon un autre protocole qui détecte les collisions est aussi valable il s'appelle CSMA/CD (Carrier Sense Multiple Access / Collision Detect). La norme IEEE 802.11[16] [7] utilise ces fonctions soit pour éviter les collisions comme pour le CSMA/CA, soit pour détecter les collisions et programmer une rediffusion après pour le CSMA/CD.

### 1.3.1 Caractéristiques des réseaux ad hoc

Les réseaux mobiles ad hoc ont plusieurs caractéristiques ; nous citons certaines d'entre elles :

- Topologie dynamique : la topologie du réseau est à chaque instant définie par les positions des nœuds qui se déplacent arbitrairement formant ainsi un graphe d'interconnexion composé de liaisons unidirectionnelles et bidirectionnelles.
- Liaisons à débit variable et à bande passante limitée : la capacité radio est inférieure à celle des réseaux câblés surtout si nous prenons en considération les interférences, le bruit et les accès multiples au médium qui réduisent considérablement la bande passante.
- Énergie limitée : Le nomadisme des nœuds ne leur permet pas de se déplacer avec d'énormes batteries, de ce fait une bonne gestion d'énergie doit être mise en place pour utiliser le minimum d'énergie et des modes de mise en veille sont à prévoir.
- Sécurité limitée : Compte tenue de la souplesse de déploiement des réseaux ad hoc, n'importe quel nœud peut faire partie du réseau juste en se plaçant dans une zone de propagation, où il pourra écouter tout ce qui passe par le médium physique; ce qui réduit la sécurité et s'oppose à la confidentialité. Le groupe de travail MANET (Mobile Ad-Hoc Network [8]) de IETF (The Internet Engineering Task Force [9]) aborde ce problème et prévoit des méthodes de

[16] IEEE 802.11 : Norme de transmission de données par ondes radio pour les réseaux locaux.

chiffrement, d'authentification inter-routeur, soit par simple clé partagée ou même jusqu'à une infrastructure de clés Publiques/Privées [1].

## 1.4 Le routage classique

Le routage s'occupe de l'acheminement des paquets vers les destinations désirées. Il offre des services aux différentes applications qui désirent envoyer des données à d'autres applications se trouvant dans d'autres réseaux distants. Il existe plusieurs protocoles de routage mais appartenant à deux classes majeures, les protocoles de routage statiques et dynamiques.

Dans le routage statique, les entrées de la table de routage sont créées par défaut avec la configuration de l'interface ou par des commandes. Ce type de protocoles est généralement utilisé pour les réseaux de petite taille surtout s'ils comportent un seul point de connexion.

Les protocoles de routage ad hoc sont basés sur des algorithmes distribués. Les routeurs communiquent entre eux pour s'échanger l'information des réseaux auxquels ils appartiennent et avec lesquels ils peuvent communiquer, ainsi chaque changement dans le réseau est signalé et tous les routeurs auront mis à jour leurs tables de routage [2].

Suivant le fonctionnement et la stratégie des protocoles de routage, nous distinguons les principaux protocoles :

**1.4.1 Les Protocoles à État de Liaisons** Leur stratégie est d'envoyer des paquets du type (LSP Link State Packets) entre les routeurs contenant l'adresse de tous les voisins avec le coût des liaisons qui les joignent. Ainsi, chaque routeur sera capable de calculer indépendamment des autres routeurs le chemin le plus court vers n'importe quelle destination, exemple OSPF (Open Shortest Path First [4]).

**1.4.2 Les protocoles à Vecteur de Distance** Chaque routeur transmet périodiquement un vecteur comportant pour chaque destination connue du routeur : son adresse, distance depuis le routeur qui transmet le vecteur jusqu'à cette destination. Le vecteur de distance n'est qu'un résumé de la table de routage. Chaque routeur reçoit ces vecteurs de distance provenant de ses voisins immédiats et se base sur cette information pour construire et mettre à jour sa table de routage, exemple de protocoles, RIP (Routing Information Protocol [2]).

**1.4.3 Source routing** Ce genre de routage propose pour chaque paquet émis un chemin complet dès sa mise sur réseau jusqu'à sa destination, ce qui économise le temps de calcul du meilleur chemin et surtout de routage au niveau des routeurs intermédiaires et évite aussi les boucles. Cette stratégie peut causer des surcharges au niveau des routeurs. La figure 1.2 présente un simple exemple pour ce type de routage.

Les paquets émis par la source comportent le chemin qu'ils doivent parcourir (N1-N2-N5-N8, sur l'exemple de la figure 1.2, ainsi les routeurs intermédiaires n'auront pas à consulter leurs tables de routage.

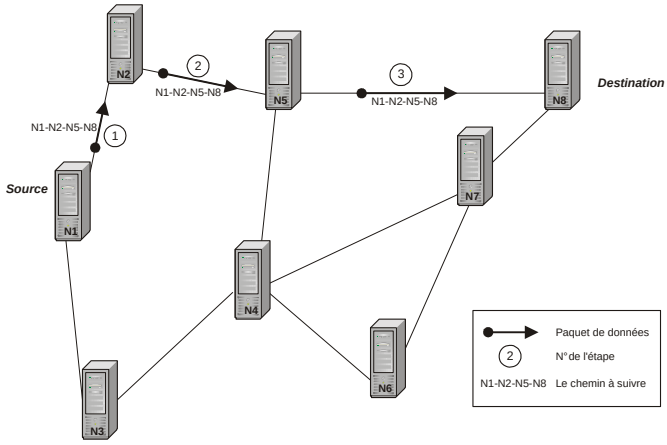



*FIG 1.2 - Routage par « Source Routing ».*

**1.4.4 Flooding** Le flooding (l'inondation) est une technique utilisée dans le routage classique pour le multicast (Routage d'une source vers plusieurs récepteurs). Elle présente beaucoup d'inconvénients comme l'usage inutile de la bande passante. Son principe est très simple, le nœud qui désire envoyer des paquets, les transmet à ses voisins (de même zone de propagation) qui, à leur tour les retransmettent à leurs voisins et ainsi de suite jusqu'à ce que tous les nœuds du réseau aient le paquet. L'estampille est quand un paquet passe plusieurs fois par un même serveur, pour cela l'usage d'un numéro de série unique pour chaque paquet est possible, il sera incrémenté par la source à chaque nouveau paquet.

Sur la figure 1.3, nous voulons que l'émetteur envoie les paquets sur toutes ces connexions ainsi que tous les autres nœuds intermédiaires (flèches en ligne continue sur la figure 1.3).Quelques-uns peuvent recevoir plusieurs fois le même paquet, alors ils l'ignorent (flèches en pointillés sur la figure 1.3). Ainsi de suite jusqu'à atteindre le ou les récepteurs.

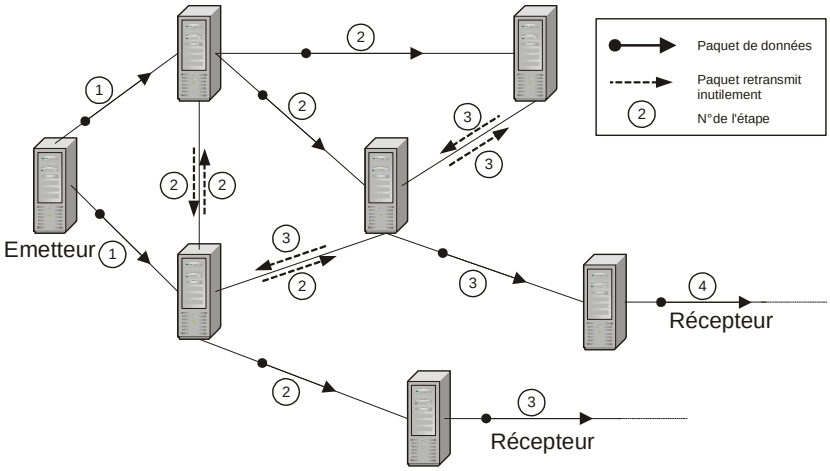



*FIG 1.3 - Routage par « Flooding ».*

## 1.5 Protocoles de routage dans les réseaux ad hoc

Les protocoles de routage classiques ne peuvent pas s'appliquer aux réseaux ad hoc compte tenu de leur mobilité et support de transmission spécifique. Il faut adapter ces protocoles pour ce genre de réseaux qui changent de topologie fréquemment et aléatoirement. Les protocoles de routage ad hoc sont divisés en trois catégories, deux principales et la troisième est issue de leur combinaison, ces catégories sont : Les protocoles réactifs, les protocoles proactifs et les protocoles hybrides. La différence entre les deux grandes catégories est que les réactifs doivent initialiser le chemin entre la source et la destination avant d'envoyer les données par une demande de chemin. Par contre les proactifs ont dans leur table de routage tous les chemins vers tous les nœuds du réseau, il ne leur manque que d'utiliser ce chemin pour envoyer directement les données à la destination. Les protocoles hybrides sont le compromis entre les protocoles réactifs et les protocoles proactifs. Ils se comportent en proactifs juste dans leur voisinage. Par contre, pour trouver des chemins plus longs ils procèdent en réactifs.

### 1.5.1 Protocoles réactifs

Ce type d'algorithmes ne nécessite pas le maintien permanent de tables de routage, ni une connaissance préalable de la topologie du réseau au moment de l'envoi. Par contre une phase précédant l'envoi consiste à rechercher le chemin vers la destination voulue. Cette phase est initialisée par la diffusion d'un paquet *« découverte de route »* depuis la source sur tout le réseau et redirigé par chaque nœud intermédiaire

jusqu'à atteindre la destination, où ce paquet sera retransmis vers la source sous forme d'une réponse *« réponse de route »* traçant ainsi un chemin vers la destination, Des algorithmes ont été proposés, des améliorations et des évaluations sont en cours. Nous citons à titre d'exemple AODV, DSR et TORA.

#### 1.5.1.1 AODV (ad hoc On-Demande Distance-Vector [11])

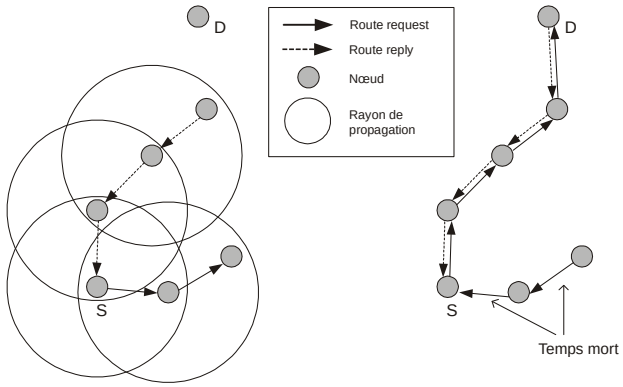
Ce protocole est basé sur l'algorithme de routage Distance-Vector (DV). Il admet aussi le routage Multicast, génère des routes fraîches sans cycle parce qu'il utilise le numéro de séquence sur les paquets. Quand une source veut transmettre des paquets, elle tente tout d'abord de trouver le chemin vers sa destination en diffusant un paquet de demandes de chemin Route Request notée *[Rreq]* à tous ses voisins. A leurs tours, les voisins retransmettent ce même paquet mais en enregistrant l'adresse du prédécesseur et en ajoutant sa propre adresse pour le successeur, pour garder la trace du chemin parcouru, et ainsi de suite jusqu'à ce qu'il soit arrivé à destination qui après son acceptation de la demande, transmet sa réponse Route Reply notée *[Rrep]* à la source directement en suivant le chemin inverse ou bien en diffusion *[Broadcast]*. A la réception de *[Rrep]* que la source attendait, elle enregistre localement le chemin vers sa destination qui reste valide pour un temps précis, sauf dans le cas d'une rupture de liaison signalée par l'un des nœuds de ce chemin. Dans ce cas la source relance un autre *[Rreq]* après une mise à jour de sa table.
Une topologie mixte entre réseaux sans fils (ad hoc) et réseaux filaires est simple à mettre en œuvre. AODV provoque moins de message de contrôle en créant des routes juste à la demande et il s'oppose aussi au maintien d'une liste complète des routes, de ce fait il est appelé routage sur demande. Les nœuds qui ne sont pas sur un chemin choisi ne maintiennent pas l'information de routage et ne participent pas aux échanges de tables de routages. Chaque nœud maintient son numéro de Séquence avec l'ID de la diffusion (qui est incrémenté à chaque *[Rreq]* initialisé par un nœud). Pour garantir des routes sans cycle et récentes. La source inclus dans le *[Rreq]* le numéro de séquence le plus récent pour la destination voulue, et tout nœud intermédiaire peut répondre à ce *[Rreq]* s'il possède pour cette destination un numéro de séquence plus grand ou égal à celui contenu dans le *[Rreq]*. Durant les retransmissions des *[Rreq],* chaque nœud enregistre l'adresse de son voisin prédécesseur et ce pour garantir le chemin inverse. Dans le cas d'une réception du même *[Rreq]*, les nœuds le rejettent. Durant le retour (chemin inverse) chaque nœud met à jour sa table de routage vers la destination qui avait émis le *[Rrep]* (voir Figure 1.4). Chaque entrée de la table de routage est effacée si celle-ci n'était pas

utilisée ou mise à jour après un temps déterminé.

AODV ne supporte que l'utilisation des liens symétriques. Dans le cas de la mobilité des nœuds :

- Si la source bouge elle n'a qu'à réémettre un nouveau
- Si un nœud appartenant au chemin change de position, son nœud ascendant notifie par un message de rupture de lien (*[Rrep]* positionné à l'infini) chaque nœud actif du chemin ascendant, et ainsi de suite jusqu'à ce que la source soit informée de cette rupture de lien et elle va décider de lancer à nouveau un *[Rreq]* si elle tient toujours à envoyer à cette destination.

Sur la figure 1.4 (a), la réponse à la *[Rreq]* est envoyée par un nœud intermédiaire en suivant le chemin inverse. Par contre sur la figure 1.4 (b), le *[Rrep]* est transmis par la destination D de nœud à l'autre jusqu'à la source S.




(a) Formation par chemin inverse (b) Formation par chemin transmis

*FIG 1.4 - Formation de chemin sous AODV*

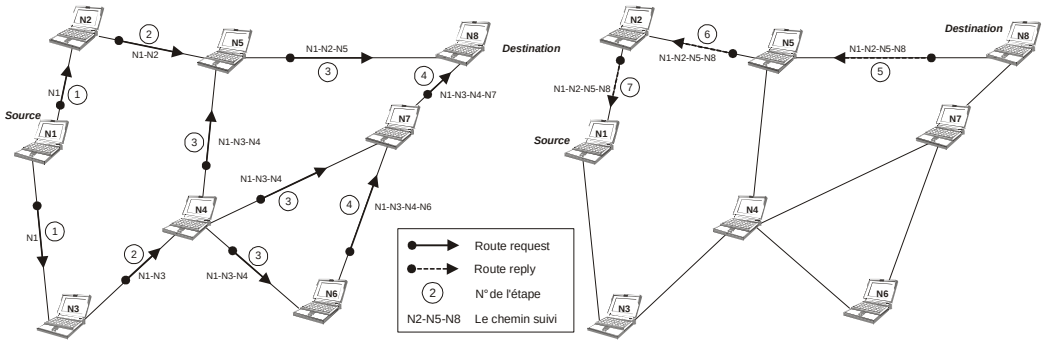
#### 1.5.1.2 DSR (Dynamic Source Routing [12])

Ce protocole est réactif du fait que les nœuds doivent découvrir la route avant chaque transmission. L'avantage de cet algorithme est l'allégement du réseau des paquets d'erreurs et indication de rupture de liens, où chaque couche MAC[17] (Medium Access Control [10]) doit informer son protocole de la rupture de ses liens et une mise à jour est effectuée. Alors le principe de cet algorithme est la découverte du chemin puis son maintien.

[17] La couche MAC : Responsable pour contrôler l'accès au réseau.

- La découverte s'effectue par l'envoi d'un *[Rreq]* et l'attente du *[Rrep]* qui pourrai être envoyé soit par la destination soit par un nœud existant sur le chemin et possédant dans son cache la suite du chemin vers la destination.
- Le maintien c'est en fait un suivi des transmissions des paquets dans l'attente d'un message d'erreur envoyé par un nœud qui précède le nœud qui n'est plus sur son rayon de propagation (n'est plus un voisin) alors qu'il l'était avant et sur le même chemin vers la destination. Dès réception du message d'erreur contenant l'adresse du nœud sortant, la source met à jour ses caches en tronquant tous les chemins contenant le nœud en cause. Dès qu'un nœud désire envoyer un paquet, il doit d'abord consulter sa table des routes si un chemin vers sa destination qui n'a pas encore expiré existe il l'utilise, sinon il lance une requête de recherche de chemin *[Rreq]* avec l'adresse de la destination, son adresse et un numéro d'identification unique. Chaque nœud intermédiaire vérifie sur son cache s'il possède une entrée vers cette destination, sinon il ajoute son adresse à la liste des adresses contenues dans le paquet et l'envoie vers ses liens sortants (Figure 1.5). Pour limiter le nombre de demandes répétées chaque lien vérifie si cette demande n'a pas été traitée avant et si son adresse ne figure pas sur la liste des nœuds transités par le paquet *[Rreq].* Si la réponse *[Rrep]* doit être envoyée par le nœud destination il ne fera que copier la liste des nœuds transités qui est dans le *[Rreq]* dans *[Rrep]* et l'envoyer à la source. Sinon si un nœud intermédiaire est chargé de le faire, alors il ajoute à la liste des nœuds visités l'entrée de sa table des routes qui coïncide avec la destination (si elle est toujours valide). Le paquet résultant *[Rrep]* sera transmis à la source soit en suivant le chemin inverse (si l'algorithme supporte les routes symétriques) ou bien par une initialisation de route vers la source faute d'une entrée de table pour cette source. Si une erreur de liaison est entendue, tout nœud doit enlever le nœud en question de toute sa table et tronquer toutes les routes le contenant en ce point précis. Pour router les messages d'erreurs, des acquittements sont utilisés pour vérifier le bon fonctionnement de ces notifications où l'acquittement passif est utilisé aussi quand les mobiles sont capables d'écouter le prochain routage des nœuds de la zone de propagation.

Sur la figure 1.5 (a), la source diffuse un *[Rreq]* qui se propage dans tout le réseau en gardant trace du chemin parcouru. La réponse *[Rrep]* de la destination que nous pouvons voir sur la figure 1.5 (b), est envoyée à la source suivant le chemin emprunté par *[Rreq].*




(a) Construction de l'enregistrement du chemin durant la découverte de la route.

(b) Propagation de la réponse de la route avec l'enregistrement du chemin.

*FIG 1.5 – Création d'une route dans DSR [5]*

#### 1.5.1.3 TORA (Temporally Ordered Routing Algorithm [13])

TORA produit seulement le mécanisme de routage et repose sur le protocole IMEP (Internet MANET Encapsulation Protocole [15]), le protocole élaboré pour supporter les opérations de nombreux algorithmes de routage ou d'autres protocoles de couches supérieures dans les réseaux mobiles ad hoc.

TORA repose sur 3 fonctions :

- Création de route.
- Maintien de route.
- Effacement de route.

A tout nœud lui est associée une hauteur. Les messages transitent d'un nœud de grande hauteur à un autre de plus petite hauteur.

- La création de route : elle est effectuée à la demande de la source *[QRY]*, le nœud qui possède (dans son cache) la route vers la destination on le nœud destination lui-même envoi la réponse UPD à la source contenant sa hauteur et avec cette méthode la source aura plusieurs routes.
- Maintien de route : il est effectué grâce aux messages de contrôle du voisinage (voir Figure 1.6).
- L'effacement de route : il est effectué à l'issue de la réception du message d'effacement Clear packet *[CLR]*.

TORA repose sur le protocole IMEP pour réduire le nombre de messages de contrôle. La *« hauteur »* assignée à chaque nœud aide beaucoup pour la création du graphe dirigé acyclique *DAG*[18] où la destination représente la racine.

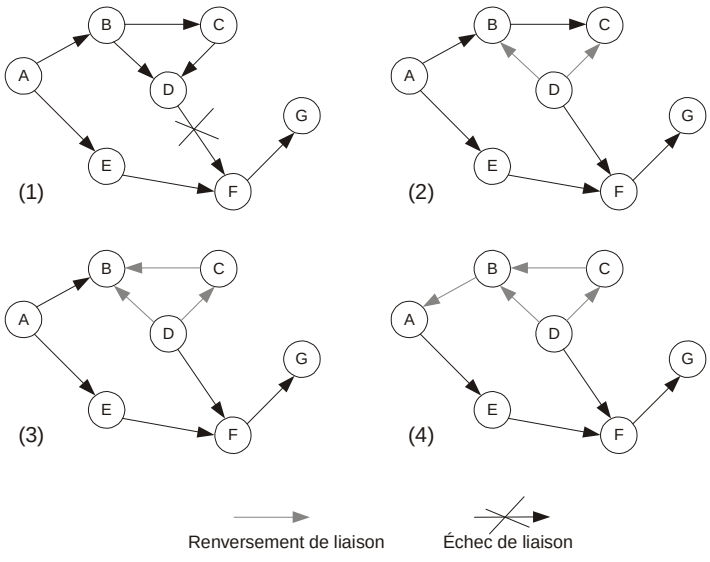
[18] DAG : Graphe Dirigé Acyclique (Directed Acyclic Graph).

Avec la mobilité des nœuds, le *DAG* est brisé quelque part et une maintenance est nécessaire. La synchronisation est un facteur important pour TORA parce que la *« hauteur »* dépend du temps logique de rupture de liens TORA présume que tous les nœuds sont synchronisés depuis une source unique (comme le Système de Position Globale GPS). La métrique de TORA est un quintuple qui comporte :

- Période logique d'une rupture de lien.
- Un ID unique qui définit la nouvelle référence des niveaux.
- Un bit indicateur de réflexion.
- Un paramètre d'ordre de propagation.
- Un ID unique du nœud.

Les trois premiers éléments représentent la référence de niveau, où une nouvelle référence de niveau est définie chaque fois qu'un nœud perd son dernier lien descendant due à une rupture de lien. L'effacement de route produit essentiellement un broadcast du message *[CLR]* sur tout le réseau pour effacer toute route non valide. Dans TORA, il peut arriver que des oscillations se produisent surtout quand différentes configurations de nœuds coordonnés détectent concurremment des partitions. Comme TORA utilise la coordination inter nodale, son problème d'instabilité se rapproche de celui du *« compte à l'infini »* de l'algorithme de routage Vecteur de Distance, par contre sur TORA les oscillations sont temporaires et les routes convergeant tôt ou tard.

Sur la figure 1.6, quand le nœud D détecte une erreur de liaison avec le F, il diffuse un message d'erreur *[CLR]* sur tout le réseau. Ses voisins font de même après réception du message. Une fois le message d'erreur reçu par la source A, elle utilise un autre chemin pour envoyer ses données à sa destination parce que TORA génère plusieurs routes.




(a) A : Source G : Destination

*FIG 1.6 - Maintenance de route sous TORA.*

### 1.5.2 Protocoles Proactifs

Ce type d'algorithme est différent du précédant du fait qu'il est principalement basé sur la garde des tables de routage au niveau de chaque nœud du réseau. Chaque émetteur consulte sa table pour trouver une entrée vers la destination voulue. Plusieurs algorithmes ont été proposés et même implémentés et testés. Exemple DSDV, CGSR, WRP et OLSR.

#### 1.5.2.1 DSDV (Destination Sequenced Distance-Vector Routing [14])

Ce protocole est basé sur le mécanisme de routage classique de Bellman-Ford. Chaque nœud du réseau maintient une table de routage vers toute destination possible. La mise à jour des tables de routage est faite périodiquement Pour alléger la charge du le réseau, il existe deux méthodes de mise à jour :

- *Full dump* : Des paquets transportant toute information disponible du routage et peut demander plusieurs NPDUs, leur transmission est occasionnelle.
- *Incremental* : Des paquets portent juste l'information de changement sur les tables de routage depuis la dernière mise à jour. Les nœuds maintiennent aussi une autre table où ils enregistrent les paquets envoyés.

#### 1.5.2.2 CGSR (Clusterhead Gateway Switch Routing [17])

Ce protocole décompose la topologie en un groupe de clusters où dans chaque cluster un algorithme distribué est chargé d'élire un leader qui prend la responsabilité de tout le cluster (voir Figure 1.7). Plusieurs schémas de routage sont acceptés. L'allocation de la bande passante est aussi possible. CGSR utilise DSDV comme protocole de routage fondamental, mais ici l'approche est hiérarchique où pour le routage il faut passer par les têtes de cluster. La gestion de ce système de têtes de cluster peut réduire les temps d'acheminements des paquets utiles. Une passerelle *Gateway* est le nœud chargé de router les paquets entre deux têtes de cluster. Le routage s'effectue dans l'ordre : Source - Tête de cluster - Gateway - ..... - Tête de cluster - Gateway - Destination. Chaque nœud garde une table de membres de clusters et une autre table de routage. Elles sont mises à jour en utilisant DSDV. Avec ces deux tables chaque nœud sera en mesure de déterminer la tête de cluster le plus proche de sa destination. Sur la figure 1.7, nous remarquons que le nœud 1 pour envoyer au nœud 8, il doit s'adresser à la tète de cluster auquel il appartient puis c'est elle qui prend en charge le reste du routage en passant par les *Gateways* et même d'autres têtes de clusters.

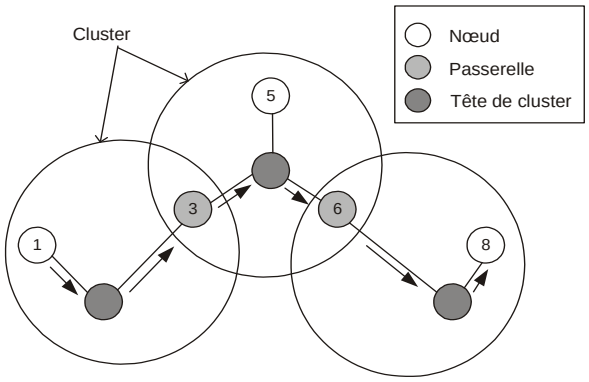



*FIG 1.7 - Routage du nœud 1 au nœud 8 par CGSR.*

#### 1.5.2.3 WRP (Wireless Routing Protocol)

Chaque nœud dispose de 4 tables
- Table de destination
- Table de routage
- Table des coûts des liens
- Table de la liste des messages à retransmettre (MRL)[19]

Chaque entrée de la table MRL contient :
- Numéro de séquence du message de mise à jour.
- Un compteur de retransmission.
- Un vecteur de flag des demandes d'acquittement avec une entrée par voisin.
- Une liste de mise à jour qui sera envoyée dans les messages de mise à jour

Les enregistrements de la MRL mis à jour depuis la réception d'un message de mise à jour doivent être retransmis aux voisins qui doivent accuser réception. Les messages sont envoyés seulement entre les nœuds voisins et contenant une liste de mise à jour (destination, distance à la destination, prédécesseur de la destination), aussi bien une liste de réponses indiquant quels nœuds devraient reconnaître la mise à jour. Les nœuds envoient des mises à jours après traitement de mise à jour reçue ou bien après avoir détecter des changements dans les liens voisins. En cas de rupture de liens, un message de mise à jour est envoyé aux voisins qui à leurs tours mettent à jour leurs tables des distances et cherchent une nouvelle route par d'autres nœuds. Chaque changement est signalé par un message de mise à jour aux voisins. Sans les messages de mise à jours, les nœuds s'envoient mutuellement des paquets *« hello »* pour confirmer l'existence et la validité des liaisons sinon nous pourrions déduire qu'une liaison vient d'être perdu. Dans le cas d'une réception d'un nouveau *« hello »*

[19] MRL : Message Retransmission List (Liste des Messages à Retransmettre).

envoyé par un nouveau nœud, nous l'insérons dans la table de routage et une copie de cette table sera envoyée au nouveau nœud. L'exception majeure de cet algorithme est qu'il évite le problème du *« compte à l'infini »* en obligeant chaque nœud d'effectuer des tests consistants sur les informations disponibles depuis les voisins. Ainsi il n'y aura plus de boucle et un routage plus rapide est garanti.

#### 1.5.2.4 OLSR (Optimized Link State Routing [16])

Protocole pro-actif présente une optimisation de *« link state »* dont:

- La réduction de l'impact de l'inondation sur le réseau, par la réduction du nombre de nœuds participants juste aux Multi-Points Relais *[MPR]*, ce qui économise la bande passante.
- La minimisation de la taille des messages de contrôle, qui ne contiendront que l'information du voisinage de l'expéditeur mais pas de tous le réseaux.
- En plus des routes sans cycle qui sont garanties, OLSR offre des routes symétriques de plus court chemin.

Des paquets *[TC] « Topology Control »* sont périodiquement diffusés dans le réseau, ne transitent que par les *[MPR]* et ne contiennent que la liste des relais multipoints *[MPR]*.

Un système d'élection des *[MPR]* est mis en place et chaque nœud élu reçoit l'information dans un message *« hello ».*

OLSR supporte l'adressage IP, où à chaque nœud est associé une adresse IP régulière, de plus il ne demande aucun changement sur le format des paquets IP. Le protocole n'intervient que sur la gestion de la table de routage.

### 1.5.3 Protocoles hybrides

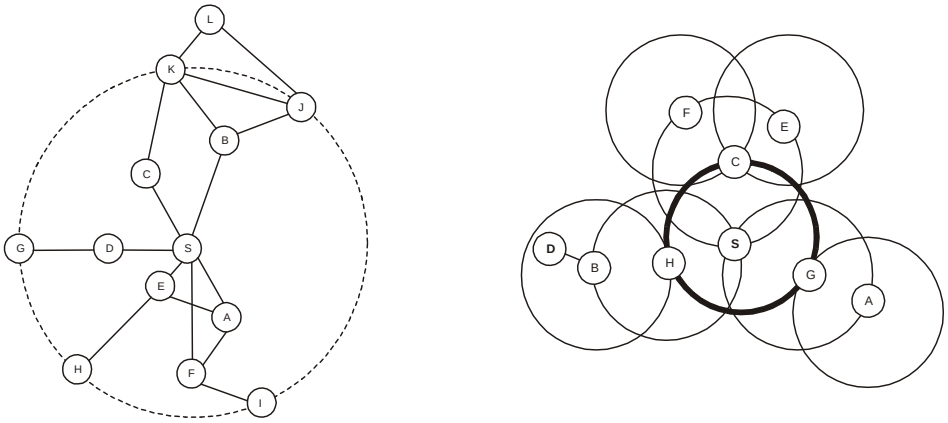
Ce type de protocoles est un compromis entre les réactifs et les proactifs et ce sont des protocoles qui d'un côté utilisent une procédure de détermination de route sur demande mais de l'autre un coût de recherche limité.

#### ZRP (Zone Routing Protocol [18])

Le protocole ZRP est un modèle hybride entre un schéma 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 (voir Figure 1.8 (a)). 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 destinations potentielles peut également l'être, ce qui requiert des

échanges de données important et fréquents. Donc la quantité de données de mise à jour du trafic peut être conséquente. Cela est en contradiction avec le fait que toutes les mises à jour dans un réseau interconnecté ad hoc circulent dans l'air et donc sont coûteuses en ressources. Le protocole ZRP limite la procédure proactive uniquement aux nœud voisins et d'autre part, la recherche à travers le réseau (voir Figure 1.8 (b)), est effectuée de manière efficace dans le réseau, contrairement à une recherche générale sur tout le réseau.

Sur la figure 1.8 (a), Le nœud source S est centré dans un grand cercle en pointillés qui délimite sa zone de routage à rayon 2, dans laquelle il se comporte comme les protocoles proactifs. Par contre, sur la figure 1.8 (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.




(a) Zone de routage à rayon 2 (b) Opération interzone

*FIG 1.8 - Le routage dans ZRP.*

## 1.6 Propriétés ciblées par les protocoles de routage des réseaux ad hoc

Les protocoles ad hoc vérifient des propriétés que d'autres n'ont pas, mais il reste encore d'autres dignes d'intérêt. Nous citons certaines propriétés pour les deux cas :

### 1.6.1 Distribution des opérations

Il faut que chacun des nœuds agisse tout seul suite à un événement et il ne doit dépendre d'aucun autre nœud. Tous les nœuds sont au même niveau et il n'existe aucune hiérarchie ni de structure centralisée pour la supervision, seuls les trois états sont accessibles récepteur, émetteur ou routeur.

### 1.6.2 Routes sans cycle

Le protocole doit générer des routes sans cycle *« loop-free »* ce qui nous évite les pertes sur la bande passante ainsi que la consommation de ressources CPU ou d'énergie.

### 1.6.3 Opération à la demande

La réaction à la demande permet d'économiser de l'énergie. Parce qu'avec les messages périodiques, non seulement la bande passante est mal exploitée mais en plus l'énergie est utilisée inutilement dans la plupart des cas.

### 1.6.4 Liens unidirectionnels

L'usage de la technologie radio provoque des liaisons unidirectionnelles surtout si des obstacles physiques se trouvent dans l'environnement, où nous trouvons un nœud qui peut atteindre un autre par contre la réciproque ne peut se faire. L'acceptation des liens unidirectionnels améliore considérablement les performances du protocole.

### 1.6.5 La sécurité

Les réseaux ad hoc sont très sensibles aux attaques. Des mesures de sécurité sont appelées à être mis en place et l'usage de l'authentification, du tatouage de l'information et la cryptographie semblent nécessaires. Il y a même des discussions sur IP-sec qui introduit les tunnels pour le transport des paquets.

### 1.6.6 Conservation d'énergie

Étant donné que l'énergie emmagasinée dans les batteries des nœuds mobiles est réduite et très limitée, il serait nécessaire que les protocoles proposés supportent le mode veille au niveau de chacun des nœuds.

### 1.6.7 Multi-routes

Si chaque nœud peut stocker plusieurs routes vers la même destination, les changements fréquents de la topologie influeront peu sur le trafic où nous trouvons moins de congestions et peu de paquets de contrôles issues des découvertes de routes.

### 1.6.8 Le support de la qualité de service

Plusieurs modèles de qualité de service sont envisageables pour les protocoles ad hoc et beaucoup d'applications verront le succès, comme le support du trafic temps réel et la réservation de la bande passante pour la vidéo conférence à titre d'exemple.

## 1.7 Comparaison

Sur le tableau 1.2, une comparaison entre les différents protocoles de routage ad hoc accentue les différences et met en valeur les propriétés des chacun des protocoles [6]. Nous remarquons par ailleurs qu'aucun des protocoles ne supporte la conservation d'énergie ou la qualité de service, ce qui pousse les recherches vers ces points précis.

| | DSDV | AODV | DSR | CGSR | WRP | OLSR | ZRP | TORA (IMEP) |
|---|---|---|---|---|---|---|---|---|
| Routes sans cycle | Oui | Oui | Oui | Oui | Oui, Pas instantané | Oui | Oui | Non, cycle à temps |
| Multi-routes | Non | Non | Oui | Oui | Non | Non | Non | Oui |
| Distribution | Oui | Oui | Oui | Oui | Oui | Oui | Oui | Oui |
| Réactif | Non | Oui | Oui | Non | Non | Non | Partiel | Oui |
| Liens unidirectionnels | Non | Non | Oui | Non | Non | Non | Non | Non |
| Support de la Qos | Non | Non | Non | Oui | Non | Non | Non | Non |
| Multicast | Non | Oui | Non | Oui | Non | Oui | Non | Non |
| Sécurité | Non | Oui | Non | Non | Non | Non | Non | Non |
| Conservation d'énergie | Non | Non | Non | Non | Non | Oui | Non | Non |
| Diffusions périodiques | Oui | Oui | Non | Oui | Oui | Oui | Oui | Oui (IMEP) |

*TAB 2 - Tableau comparatif des différents protocoles de routage ad hoc [6]*

# Chapitre 2.

# Cadre Expérimental

## Etude de simulation

### 2.1 Introduction

Les documentations qui traitent sur les réseaux sans fil Ad hoc nous permettent de connaître le type et caractéristiques fonctionnelles de chacun des protocoles ainsi que son algorithme de routage. Toutes les comparaisons sont faites sur un plan purement théorique. Elles ne nous permettent pas de classer ces protocoles en fonction des critères réels.

Depuis toujours la nature humaine rêve de connaître le futur et dans notre cas, de connaître les différents comportements des protocoles avant même de les implémenter et les commercialiser. L'étude de simulation offre cette vision du futur qui est quasiment réelle.

Une simulation sur les réseaux sans fil qui ne prend pas beaucoup de temps, nous rapproche de l'utilisation réelle du protocole de routage. Ces deux grands avantages nous aident à sélectionner les meilleurs protocoles qui ont un bon comportement dans différents scénarios. Les développeurs de protocoles aussi font des études de simulation pour améliorer les capacités de leur algorithme de routage.

Dans le cadre de notre projet nous avons opté aussi pour le cadre expérimental de la simulation des réseaux ad hoc pour mieux comparer différents protocoles de routage, tirer des conclusions et peut être même faire un pas vers l'amélioration.

#### 2.1.1 Les simulateurs de réseaux

Dans le monde de la simulation des réseaux informatiques, beaucoup de contributions ont participé à l'enrichissement de ce domaine. La majorité des travaux effectués sont d'ordre pédagogique et sont élaborés par des laboratoires informatiques à travers le monde.

Beaucoup d'axes de recherches se basent sur ces simulateurs. En fonction des particularités de chacun, les chercheurs choisissent le simulateur le plus approprié.

Les simulateurs les plus répandus sont OMNET ++, NS-2 [20] [22], SensorSIM, GlomoSim [21], QualNet, Jist / SWANS, JSim, OPNET[21] [23] Modeler [26].

##### 2.1.1.1 Omnet ++[22]

Il est utilisé sur la plate-forme Microsoft Windows (avec Cygwin), Unix. Sa licence est gratuite pour les universitaires et pour toute utilisation non lucrative.

[20] NS : The Netword Simulator
[21] OPNET : Open NETwork
[22] OMNET ++ : http://www.omnetpp.org/

Il ne semble pas particulièrement prévu pour le sans-fil. Il n'existe pas de modèles spécifiques aux capteurs indiqués sur le site Web. Cependant Omnet++ semble séduire de plus en plus la communauté scientifique et un nombre croissant de modèles est disponible.

#### 2.1.1.2 NS-2[23]

Il est utilisé sur la plate-forme Unix (Linux, solaris, Mac OS X incertain), Microsoft Windows. Sa licence est gratuite.

NS-2 est très utilisé pour les réseaux ad hoc et les réseaux filaires. Toutefois les modèles de couche physique sont simplistes. Le développement des protocoles s'effectue en C++ et en OTcl (évolution objet de TCL). Les scénarios sont décrits en OTcl. La prise en main est peu aisée ; en effet OTCL est peu connu et la programmation en C++ nécessite de comprendre l'interface entre les deux langages.

L'analyse des résultats est en général peu aisée ; le résultat de la simulation étant essentiellement composé d'un fichier retraçant l'ensemble des envois, réceptions et suppressions de paquets. Un certain nombre de scripts ont été développés (ou sont en cours de développement) pour faciliter cette analyse. Du fait de sa popularité, de nombreux protocoles sont a priori disponibles pour NS-2. Quelques protocoles spécifiques aux réseaux de capteurs sont disponibles.

#### 2.1.1.3 SensorSIM[24]

Il est utilisé sur la plate-forme Unix (Linux, solaris, Mac OS X incertain), Microsoft Windows avec une licence gratuite.

Il s'agit d'un projet d'UCLA[25] visant à créer un simulateur spécifique aux réseaux de capteurs sur la base de NS-2. Cependant, le projet vient de démarrer et l'échéancier n'est pas clair. Les sources ont d'ailleurs été retirées de la page du projet du fait de l'absence de support.

#### 2.1.1.4 GlomoSim[26]

Utilisé sur la plate-forme Unix avec une licence gratuite pour les universitaires.

Peu de modèles semblent disponibles. Le moteur de GlomoSim est basé sur la bibliothèque Parsec [27] . Le simulateur peut donc être parallélisé. Toutefois, l'apprentissage de cette API peut se révéler difficile.

---

[23] NS-2 : http://www.isi.edu/nsnam/ns/

[24] SensorSIM : http://nesl.ee.ucla.edu/projects/sensorsim/

[25] UCLA : Université de Californie à Los Angeles

[26] GlomoSim : http://pcl.cs.ucla.edu/projects/glomosim/

[27] Parsec : Le langage de programmation de GlomoSim

#### 2.1.1.5 QualNet[28]

Il est utilisé sur la plate-forme Microsoft Windows, Linux, Solaris. Simulateur payant. Des réductions sont appliquées pour la recherche.

QualNet est la version commerciale de GlomoSim. Une documentation plus fournie que GlomoSim et un support technique sont fournis. Une interface graphique est aussi intégrée au logiciel.

Le projet européen Bison a choisi, après étude comparative de l'ensemble des simulateurs, d'utiliser ce produit pour sa facilité d'utilisation et son caractère inter opérable.

#### 2.1.1.6 Jist / SWANS[29]

Il est développé sous Java. Sa licence est gratuite. Jist est le moteur de simulation, SWANS est le "simulateur", c'est-à-dire une interface de Jist.

Jist permet d'utiliser, comme générateur de trafic, n'importe quelle application Java. Il souffre cependant du manque de modèles lié à sa jeunesse. Jist adopte un mode de programmation similaire à J-Sim (essentiellement Java). Les protocoles sont conçus comme des composants indépendants interconnectés par des interfaces. Concernant l'aspect passage à l'échelle, il présente de meilleures performances que J-Sim. Le calcul de la propagation est "optimisé".

#### 2.1.1.7 JSim[30]

Il est développé sous Java. Sa licence est gratuite J-Sim est utilisé à l'INT[31] et permet de simuler des réseaux de l'ordre de 1000 noeuds. Le passage à l'échelle peut toutefois être amélioré. J-Sim souffre peut être de sa jeunesse, quelques corrections étant nécessaires. Le simulateur utilise quasi-indifféremment deux langages : Java et TCL. L'architecture et le code sont suffisamment bien structurés pour permettre une prise en main relativement aisée. L'analyse des résultats est aisée. J-Sim permet d'utiliser, comme générateur de trafic, n'importe quelle application Java. J-Sim semble gérer correctement l'aspect consommation d'énergie.

#### 2.1.1.8 Opnet Modeler[32]

Il utilise la plate-forme Microsoft Windows (NT, 2000, XP) et Solaris. Sa licence est payante. Il est possible de l'obtenir gratuitement en s'inscrivant au programme Opnet pour les universités. Le développement s'effectue en C++, au travers de l'interface du

---

[28] QualNet : http://www.scalable-networks.com/products/qualnet.php
[29] Jist / SWANS : http://jist.ece.cornell.edu/
[30] JSim : http://chief.cs.uga.edu/~jam/jsim/
[31] INT : Institut National des Télécommunications
[32] OPNET Modeler : http://www.opnet.com/

logiciel. Les scénarios se spécifient essentiellement à travers la même interface. Toutefois, un mode batch est disponible et permet de réaliser des batteries de simulations. Les modèles (protocoles) fournis avec le simulateur sont validés et précis. Toutefois, la majorité des modèles sont écrits par des développeurs indépendants (modèles contribués) et ne sont pas testés par Opnet. C'est un simulateur à évènements discrets qui va de la préparation de la simulation jusqu'à l'analyse des données résultantes [20]. Il est orienté réseaux filaires pour l'analyse de l'ordonnancement ainsi que la qualité de service. Il traite aussi bien les réseaux de communication et les systèmes distribués. Des chercheurs de l'INPL[33], ont fait usage de OPNET sur des réseaux à commutation de paquets pour étudier le respect des contraintes temps réel. Un autre groupe de recherches, nommé IETF[34], a travaillé sur la simulation du protocole OLSR[35] avec OPNET où les chercheurs ont trouvé qu'avec un mécanisme hiérarchique dans le protocole OLSR, ils arrivent à réduire considérablement le trafic de contrôle sur des liens à grand débit [25].

Pour des raisons de disponibilité du simulateur au sein du laboratoire ainsi que dans la continuité des projets antérieurs, nous avons opté pour NS-2 qui nous offre plus de souplesse au niveau du paramétrage des protocoles sans fils.

Dans ce chapitre nous exposons notre étude expérimentale. Des protocoles de routage ad hoc ont été l'objet de plusieurs simulations, où différents scénarios ont été implémentés avec des paramètres variés.

### 2.2 Simulateur NS

Le simulateur réseau NS (Network Simulator) est un simulateur à événements discrets orienté objet, basée sur le simulateur réseau REAL[36]. Au départ, la version 1.0 de NS a été développée au Laboratoire National de Lawrence Berkeley[37] (LBNL) par le groupe de recherche réseau. Son développement fait maintenant partie du projet VINT sous lequel la version 2.0 est sortie.

Le projet VINT [27] (Virtual InterNetwork Testbed) est dirigé par l'Université de Californie du Sud et est financé par le DARPA[38] en collaboration avec Xerox PARC[39] et LBNL. Le but de ce projet est la construction d'un simulateur réseau qui offre des outils et des méthodes innovatrices dans un environnement proche de la réalité.

[33] INPL : Institut National Polytechnique de Lorraine (Nancy - France)
[34] IETF : The Internet Engineering Task Force
[35] OLSR : Optimized Link State Routing
[36] Simulateur réseau REAL : http://www.cs.cornell.edu/skeshav/real/overview.html
[37] LBLN : http://www-nrg.ee.lbl.gov/
[38] DARPA : http://www.darpa.mil
[39] Xerox PARC : http://www.parc.xerox.com/parc-go.html

Ce simulateur essaie de répondre aux questions de mise à l'échelle (simulation de grandes topologies) et d'interaction entre protocoles dans des services intégrés à l'Internet (problèmes d'hétérogénéité). L'extension de NS nous a permit de simuler les mobiles sans fil à base d'ondes radio.

La philosophie générale de NS est assez simple. Comme entrée, pour le simulateur de réseaux (ns), sont introduits des fichiers de script en Otcl [40] qui décrivent l'environnement avec tous ses nœuds, leurs déplacements et leurs trafics de données. (Voir l'annexe A). Après que le simulateur traite ces fichiers, il génère en sortie un fichier que nous pouvons visualiser avec NAM[41] pour voir le comportement des nœuds et des paquets émis (voir l'annexe B), ainsi qu'un autre fichier traces (journal) qui sera analysé par la suite pour générer des courbes.

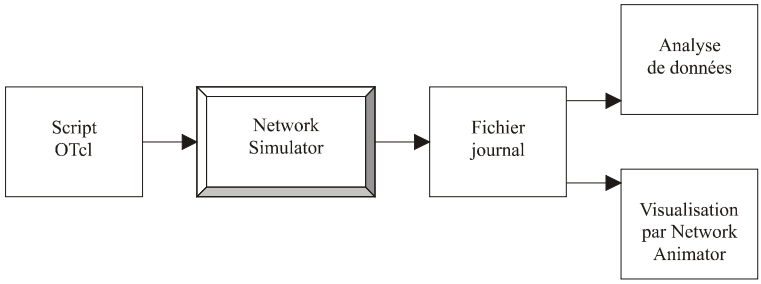



*FIG 2.1 - La simulation sous ns.*

[40] Otcl : Object Tool Command Language
[41] NAM : Network Animator

# Chapitre 3.
# Étude de cadrage

Les programmes implémentés pour la simulation qu'ils soient compilés ou sous forme de script ont été implémentés d'une façon qui facilite la modification des paramètres de la simulation, où nous pouvons augmenter le nombre de nœuds, changer la manière du déplacement des nœuds, augmenter ou diminuer la charge du réseau ainsi que rajouter du trafic TCP, Le temps de la simulation lui aussi est paramétrable.

### 3.1 Générateur de scripts

Pour lancer un grand nombre de simulations. Il est nécessaire de disposer d'un grand nombre de scripts de description de topologie et de description de trafic de données du réseau à simuler. Chaque deux fichiers présentent l'entrée du simulateur NS qui lui permet de schématiser le réseau à simuler. Un programme paramétrable a été développé afin de répondre à ce besoin. Il génère une diversité de scripts de topologie et de trafic de données.

### 3.2 Paramètres de la simulation

Dans cette partie, nous allons présenter les paramètres de nos simulations, le modèle de topologie, la propagation du signal radio, le trafic des données, la mobilité des nœuds et le modèle de l'énergie utilisé par chacun des nœuds.

#### 3.2.1 Le modèle de topologie

Afin de simuler sur un environnement inspiré de la réalité, un laboratoire simple d'architecture a été imaginé pour faire l'objet de notre cadre expérimental.
La topologie utilisée est un terrain de 85x24m où les nœuds mobiles peuvent se déplacer dans un rectangle de 75x8m centré sur ce terrain.

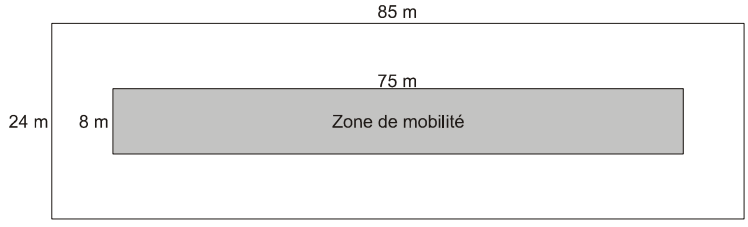



*FIG 3.1 - Terrain de simulation.*

Ce terrain est une modélisation de couloir dans un immeuble qui sépare deux alignements de 15 bureaux de 5 mètres chacun (voir Figure 3.2)

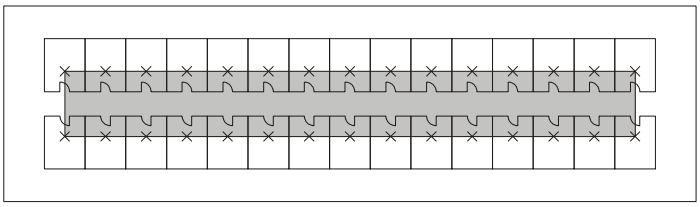


*FIG 3.2 - Architecture utilisée.*

Cette architecture nous permet de simuler un véritable immeuble où les nœuds peuvent se déplacer d'un bureau à un autre tout en passant par un grand couloir. Cette zone de mobilité des nœuds est cadrée par le rectangle gris.

### 3.2.2 Le modèle de propagation

Il existe dans NS le modèle de propagation radio « Friss-space » avec une atténuation 1/r2 pour les petites distances. Nous avons pris l'autre modèle appelé « TwoRayGround » qui emploie une atténuation de 1/r4 pour les grandes distances. Le type d'antenne radio NS choisie est « OmniAntenna » qui représente les antennes omnidirectionnelles où l'émission est en *360°*. Tous les nœuds que nous avons pris ont la même configuration et les mêmes antennes d'où les mêmes rayons de propagation.

Le protocole de l'accès au médium MAC[42] (Medium Access Control [10]) que nous avons utilisé est le IEEE 802.11[43] développé par CMU[44]. Le support de transmission a été configuré pour fonctionner comme l'interface radio du Lucent WaveLAN DSSS (914 Mhz) implémenté dans NS, avec une modification au niveau de la puissance de transmission pour donner un rayon de connexion radio approximatif à 35 mètres.

### 3.2.3 Le modèle de trafic

Le modèle de trafic utilisé ne prend en charge que les trafics du type TCP avec différentes tailles de paquets pour plusieurs combinaisons entre nombre de trafics TCP (Agents TCP) et débits utilisés. La figure 3.3 présente l'architecture d'une connexion TCP sous NS.

[42] La couche MAC : Responsable pour contrôler l'accès au réseau
[43] IEEE 802.11 : Standard pour les réseaux locaux sans fil sans infrastructure.
[44] CMU : Carnegie Mellon University


*FIG 3.3 - Modèle d'une connexion TCP dans ns.*

L'attachement des agents TCP et Sink aux nœuds, effectué dans la phase préparation, est totalement aléatoire. Le programme de génération de scripts nous permet d'avoir des connexions TCP aléatoires pour les différentes simulations

### 3.2.4 Le modèle de mobilité

Le modèle de mobilité utilisé donne le droit aux nœuds de se déplacer entre les bureaux aléatoirement avec une vitesse aléatoire aussi dans l'intervalle [1.5,3[ en mètres par seconde suivant une trajectoire rectiligne. Cet intervalle de vitesse correspond à une vitesse normale de déplacement d'une personne portant un portable.

### 3.2.5 Le modèle d'énergie

Le modèle d'énergie utilisé est celui implémenté dans NS et qui attribue aux nœuds une énergie initiale que nous avons fixé à 100 joules et, les énergies consommées, soit par une transmission ou une réception, sont celles du modèle radio Lucent WaveLAN DSSS, 0.6 et 0.3 joules respectivement.

## 3.3 Les variables de la simulation

Dans nos simulations nous avons varié quelques paramètres de configuration dans les scénarios pour accentuer les écarts entre les différents protocoles. Ainsi nous pouvons voir l'impact de chacun des critères de configuration de la simulation sur les résultats obtenus et surtout sur le comportement de chacun des protocoles. Chacune de ces variables peut favoriser l'un ou l'autre des protocoles alors qu'au même temps une autre variable aggrave ou le défavorise. Pour bien analyser ces protocoles dans différentes conditions nous avons choisi les variables suivantes :

- Le type de protocole.
- Le nombre de nœud.
- Le temps pour la mobilité.
- Le type de déplacement.
- Nombre d'agents TCP.
- Occupation de la bande passante.

### 3.3.1 Les protocoles simulés

Depuis les protocoles implémentés dans NS, nous avons pris ceux qui rentrent dans le cadre des deux types étudiés. Réactifs et Proactifs.
Les protocoles simulés étaient :

- Réactif : AODV (Ad hoc On-Demand Distance Vector)
- Proactif : DSDV (Destination-Sequenced Distance-Vector)
- Réactif : DSR (Dynamic Source Routing)

### 3.3.2 Le nombre de nœuds

Au niveau de nos topologies, nous avons varié la configuration du nombre de nœuds où le nombre maximum était limité à 30 nœuds (un nœud par bureau). Nous avons simulé aussi pour 10 et 20 nœuds. Afin d'étudier le comportement du réseau dans différentes configurations.
Un réseau peu chargé n'aura sûrement pas le même comportement que quand il est composé de beaucoup de nœuds mobiles.

### 3.3.3 La mobilité

Pour assurer la diversité des configurations et étudier l'impact de la mobilité sur la stabilité du réseau nous avons pris en considération trois cas possibles pour la mobilité :

- Simulation où tous les nœuds sont stables.

- Simulation en trois phases : où le temps de la simulation est divisé en trois phases : la première et la dernière ne comportent aucune mobilité. Seulement la deuxième phase admet une mobilité des nœuds afin d'étudier la stabilité du trafic dans le réseau durant la troisième phase.

- Simulation avec mobilité continue : Dans ce cas la mobilité est étendue sur le temps de la simulation

### 3.3.4 Les déplacements

Cette variation traite les modes de déplacement. Un mode virtuel où la topologie des bureaux n'est pas prise en compte et un autre mode réel où les nœuds se déplacent de bureau en bureau via le couloir :

- Un déplacement direct d'un bureau à un autre suivant un seul segment de droite.
- Un déplacement indirect en trois étapes (segments) où le nœud doit passer par le couloir pour changer de bureau, appelé aussi déplacement en 'Z'.

La figure 3.4 présente les deux manières de déplacement possibles des nœuds.

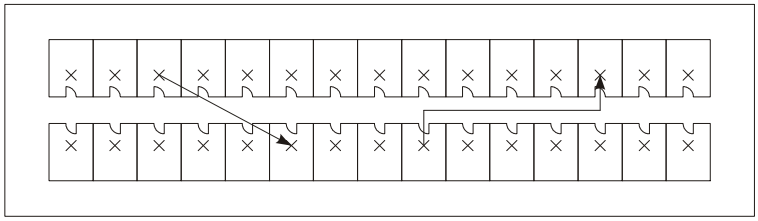


*FIG 3.4 - Types de déplacements des noeuds.*

Avec ces deux variantes du déplacement de nœuds nous pouvons étudier l'effet de la trajectoire sur le routage dynamique ainsi que sur le trafic de données.

### 3.3.5 Nombre de trafic TCP

Le type de trafic que nous avons utilisé est le TCP parce que nous pouvons étudier l'acquittement des paquets de données, ce qui va nous permettre un bon suivi du taux d'erreurs et des bonnes réceptions. Par ailleurs, l'usage du protocole UDP ne nous offre pas cette flexibilité d'analyse du fait de l'absence des acquittements des paquets (perte de paquets sans trace.)

Pour varier du nombre de trafic TCP, nous avons varié le nombre de sources émettrices afin d'étudier la charge du réseau.

Les valeurs utilisées dans nos simulations sont 3, 5, 7, 10 et 15 trafics TCP que nous pouvons appeler aussi agents TCP. Un agent TCP est lié qu'à un seul nœud source.

### 3.3.6 Occupation de la bande passante

Le choix de ce paramètre a été établi dans le but de voir le comportement du médium avec les différentes occupations de la bande passante pour tous les protocoles de routage. Du fait que le médium ne peut être utilisé à un instant « t » que par un seul et un seul émetteur, sa sollicitation est gérée comme les sections critiques.

Nous avons simulé pour des occupations de 10%, 30%, 50% et 90% de la bande passante afin de voir la capacité des protocoles de routage à gérer les goulots d'étranglement.

Pour exprimer ces pourcentages, nous avons varié la taille des paquets IP ainsi que les fréquences d'émissions.

# Chapitre 4.

# Calcul de la simulation

Ce chapitre a pour but de présenter l'étude et calcules des simulations effectuées. Les métriques utilisées ainsi que les variables qui nous ont permis d'établir les courbes représentatives. Cette phase de notre travail nous semblait très importante du fait que le bon choix des métriques et des variables nous donne la possibilité d'établir des schémas clairs et significatifs.

### 4.1 Métriques de simulation

Chacune des métriques que nous avons calculé dans nos simulations, nous aide a mieux départager les protocoles suivant leur propriétés dans différents scénarios. Une mobilité excessive ou un trafic intense peuvent affaiblir l'un des algorithmes mais pas un autre dans les mêmes conditions. De ce fait, nous avons développé quelques métriques qui nous aident à trouver les écarts entre les algorithmes simulés. Ci-dessous nos six métriques.

#### 4.1.1 Les paquets de control

Détermine le nombre de paquets émis par un nœuds dans le but de gérer le réseau (identification, recherche de route, maintien de la table de routage, maintenance de liens rompu … etc).
Comme chaque protocole a son propre algorithme de routage. Nous espérons avec cette métrique, trouver lequel d'entre eux utilise le minimum de paquets de control pour un meilleur acheminement de paquets et une meilleure visibilité de la topologie du réseau à l'importe quel instant.

#### 4.1.2 Les paquets utiles

Détermine le nombre de paquets de données utiles émises par un nœud source ou bien routeur pour un nœud destinataire. Le trafic visible dans la simulation, n'est pas utile en entier. Les messages de contrôle occupent une grande partie. Un bon protocole fini par acheminer le maximum de données utiles avec un minium de control dans n'importe quelle topologie même si la mobilité est importante.

#### 4.1.3 Les paquets perdus

Détermine le nombre de paquets de control ou de données perdus physiquement dans le réseau. Ces pertes sont issues de trafic important ou de temps d'attente assez élevé. Les nœuds routeurs perdent des paquets de données utiles ou même de control quand les liaisons sont perdues à cause du déplacement des nœuds

récepteurs où ils sortent du rayon de propagation de leurs émetteurs. Un bon algorithme génère plusieurs chemins pour éviter les pertes dans des cas pareils.

### 4.1.4 Le trafic émis

Détermine le volume des paquets de données effectives émises par un nœud source au profit d'un nœud destinataire. Un nœud source peut ne pas avoir suffisamment de temps pour envoyer toutes les données qu'il souhaite envoyer. Les principales raisons sont soit le medium est trop occupé par les nœuds voisins, soit le nœud même est trop chargé par le routage des données d'autres nœuds sources. Un bon algorithme essaie de tendre vers un bon équilibre et trouver la meilleure combinaison de chemins qui optimise les flux sortants et routés.

### 4.1.5 Le trafic routé

Détermine le volume des paquets de données effectives routées par un nœud routeur. Un nœud peut être source, routeur ou destination.  Alors l'idéal serait, que chacune des sources envoie directement à sa destination sans déléguer sa charge à des nœuds intermédiaires (routeurs). Cette situation est loin d'être la réalité, mais en choisissant des chemins plus courts, le routage sera diminué en conséquence.

### 4.1.6 Le temps d'attente forcé du médium

Il détermine la durée où un nœud source ou routeur attend la disponibilité du support de transmission pour émettre ses paquets. Ces temps d'attente peuvent augmenter en fonction de l'augmentation de la charge du réseau, de l'augmentation du nombre de nœuds ou bien de l'augmentation de la mobilité. Toutes ces situations provoquent beaucoup de trafic qu'il soit de control ou de données. Le médium n'est utilisé que par un seul nœud à la fois. Donc, tous ces voisins sont à l'écoute pour savoir s'ils sont concernés sinon ils sont en attente de libération du support de transmission.

## 4.2. Variables de la simulation

Pour bien représenter nos métriques. Il nous fallait les schématiser par rapport à des variables qui soient représentatives sur le plan quantitatif et qualitatif. Deux algorithmes de routage ad hoc peuvent avoir les mêmes valeurs au niveau des métriques parce qu'elles sont consolidées. Une fois, ces valeurs éclatées en plusieurs variables, le contraste des limites et des différences de ces protocoles sera plus

accentué. Ci-dessous sont présentées nos variables de simulation.

### 4.2.1 Mobilité

La mobilité est la moyenne de la vitesse de tous les nœuds calculée en Km/h. Quand les nœuds d'un réseau se déplacent tous à basse vitesse, les nœuds auront le temps de rafraîchir leurs tables de routage ou de trouver les meilleurs chemins, qui resteront valides un certain temps. Par contre, quand la vitesse des nœuds est élevée, les tables de routage ne sont pas fraîches. Ainsi, le réseau est déstabilisé et génère beaucoup de contrôle et de routage. Il est important d'étudier le comportement des protocoles et l'impact de la mobilité sur les algorithmes de routage.

Le tableau ci-dessous liste les valeurs discrètes de la mobilité qui ont été calculées sur l'ensemble des résultats de simulations.

| *Mobilité (km/h)* | | | | | |
|---|---|---|---|---|---|
| 0,00 | 0,10 | 0,13 | 0,28 | 0,32 | 0,42 |

*TAB 3 - Valeurs discrètes de la Mobilité*

### 4.2.2 Intensité de flux sortants

C'est le volume moyen en Ko par seconde des paquets émis par nœud. Quand, dans un réseau tous les nœuds veulent émettre en même temps, leurs fenêtres d'émissions ont tendance à se chevaucher dans le temps. Une gestion de priorité s'installe. Un réseau à forte intensité de flux sortants peut devenir très instable à cause des goulots d'étranglement qui apparaissent au niveau des files d'attente d'émissions. Une bonne gestion des flux rend le réseau plus stable et efficace.

Le tableau ci-dessous liste les valeurs discrètes de l'Intensité du flux sortant qui ont été calculées sur l'ensemble des résultats de simulations.

| *l'Intensité du flux sortant (Koctet/seconde/nœud)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| 9,68 | 16,93 | 24,60 | 29,04 | 39,45 | 48,40 | 87,12 | 114,26 |

*TAB 4 - Valeurs discrètes de l'Intensité du flux sortant*

Afin d'accentuer les écarts entre les protocoles simulés. Nous avons imaginé un protocole idéal qui répond positivement à toutes les exigences techniques et fonctionnelles. Le tableau suivant présente les variations idéales des métriques que nous avons utilisés pour ce protocole vers la hausse ↗ ou vers la baisse↘.

| Métriques | sens de variation |
|---|---|
| Les paquets de control | ↘ |
| Les paquets utiles | ↗ |
| Les paquets perdus | ↘ |
| Le trafic émis | ↗ |
| Le trafic routé | ↘ |
| Le temps d'attente forcé du médium | ↘ |

*TAB 5 - Sens de variation des métriques dans le cas idéal.*

L'algorithme idéal imaginé doit en effet : minimiser les paquets de control et envoyer un maximum de paquets utiles avec un minimum de paquets perdus. Tout en assurant un grand ratio trafic émis / trafic routé sans trop attendre pour émettre sur le médium.

# Chapitre 5.

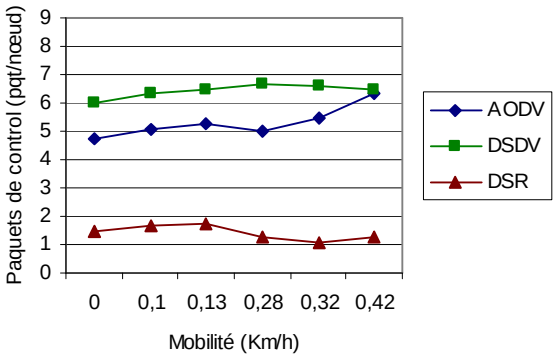
# Résultats de la simulation

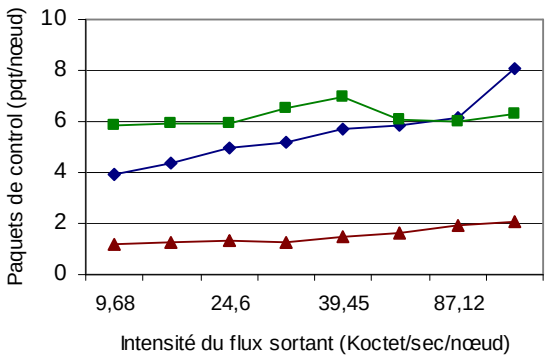
Ce chapitre a pour but de présenter tous les résultats des simulations effectuées dans le cadre de cette étude. Ces résultats sont présentés sous forme de courbes calculées en fonction de six métriques. Ces métriques sont en fonction de deux variables, la Mobilité des nœuds dans le réseau et l'Intensité du flux sortant que nous avons calculé pour ces mêmes fins.

## 5.1 Résultats et discussions

### 5.1.1 Paquets de control (pqt/nœud)

Pour ce scénario nous avons deux courbes. La première illustre le nombre de paquets de control en fonction de la mobilité (Fig. 5.1-A) et la deuxième illustre le nombre de paquets de control en fonction de l'Intensité du flux sortant (Fig. 5.1-B).




Nous remarquons sur la première figure la courbe du protocole proactif DSDV positionnée au dessus des deux autres courbes. DSDV ainsi génère plus de paquets de control du fait qu'il soit proactif (maintient permanent d'une tables de routage fraîche).

Pour le protocole réactif ADOV, le nombre de paquets de control converge vers le protocole proactif en augmentant la mobilité à cause des interruptions fréquentes des trafics de données dues au changement de position des nœuds.

Le réactif DSR se comporte mieux que les précédents même avec une forte mobilité. Nous pouvons quasiment dire qu'il est insensible à la mobilité.

Pour la deuxième figure, DSDV maintient la barre haute mais presque stable en augmentant le flux sortant. AODV qui reste toujours proche de DSDV, augmente en consommation de paquets de control avec l'augmentation du flux sortant au point où il le dépasse vers le max des flux. Réactif par sa nature, AODV vérifie bien la règle du besoin de paquets de control avec l'augmentation des flux sortants.

DSR par contre requiert un faible control. Mais en augmentant le flux, il provoque plus de control.

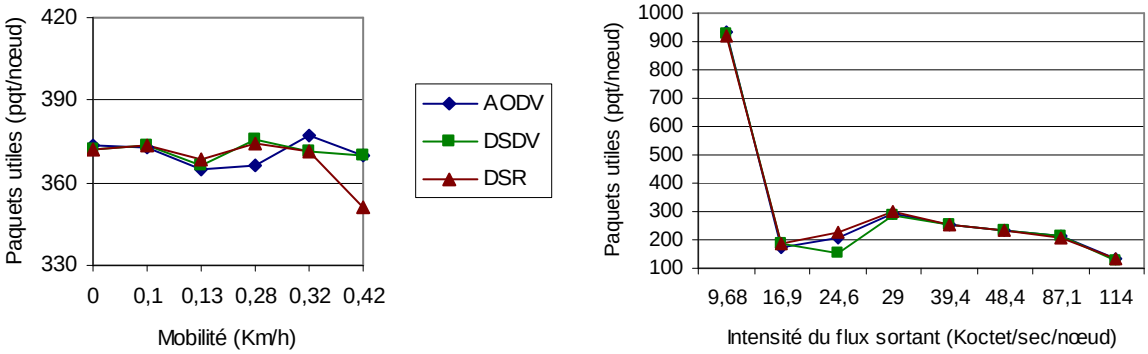
La corrélation des courbes dans la figure A et dans la figure B est bien visible. Les deux protocoles DSDV et AODV ont deux courbes bien proches. Par contre, DSR reste loin des autres avec toujours le minimum de paquets de control.

La logique théorique sollicite une augmentation du trafic de control avec l'augmentation de la mobilité ou bien l'augmentation de l'intensité du flux sortant. Elle est bien exprimée pour les trois protocoles. En théorie, les protocoles proactifs doivent avoir une consommation régulière en matière de paquets de contrôles. Par contre les réactifs utilisent de plus en plus ces paquets avec l'augmentation de la demande. Comme dans notre cas l'augmentation de la mobilité ou le flux. Le proactif DSDV semble être assez stable. Le réactif AODV augmente naturellement son besoin en paquets de control. Par contre, le réactif DSR semble baisser son utilisation en control avec l'augmentation de la mobilité, présentant ainsi une contradiction avec les théories. L'explication que nous pouvons donner à ce phénomène est que les nœuds mobiles ont arrangé le routage des liaisons de données, comme il peut y avoir aussi le cas où le déplacement des nœuds les a regroupé dans un petit rayon de propagation où le control se réduit considérablement.

### 5.1.2 Paquets utiles (pqt/nœud)

Pour ce scénario nous avons deux courbes. La première illustre le nombre de paquets utiles en fonction de la mobilité (Fig. 5.2-A) et la deuxième illustre le nombre de paquets utiles en fonction de l'Intensité du flux sortant (Fig. 5.2-B).

La première figure présente trois courbes très voisines. Avec l'augmentation de la




mobilité, les trois protocoles arrivent donc à acheminer leurs paquets avec presque le même taux.

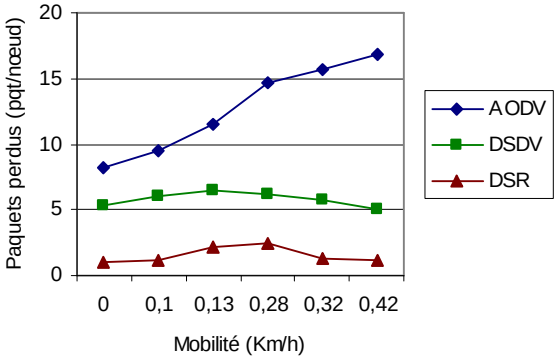
Pour la deuxième figure, tous les protocoles ont le même nombre de paquets utiles, particularité pour la première valeur de flux sortant qui enregistre un maximum de paquet utiles pour tous les protocoles.

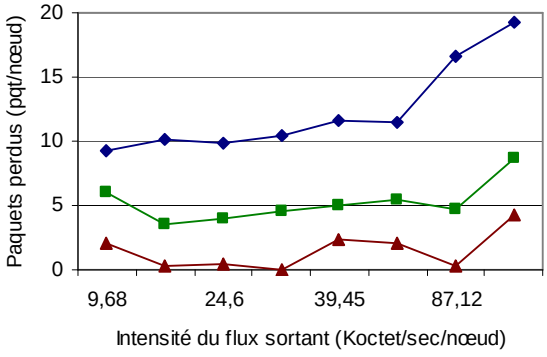
Au niveau des paquets utiles, nous ne pouvons pas dire que cette simulation nous accentue les différences entre les protocoles.

La logique théorique prévoit une réduction des paquets utiles avec l'augmentation de la mobilité ou bien le flux sortant. Quand les nœuds se déplacent beaucoup, les liaisons sont rompues provoquant ainsi la perte des paquets. Nous aurions les mêmes résultats pour le cas de l'augmentation des flux, parce qu'il y aurait beaucoup de goulots d'étranglement qui provoquent la perte des paquets. Aucune différence entre protocole réactif ou proactif n'est à souligner sur le plan théorique pour le taux des paquets utiles acheminés, parce qu'il s'agit de la fonction objective de chacun d'eux. Le but est d'en délivrer le maximum. Nos trois protocoles arrivent à garder un seuil quasi stable avec la variation de la mobilité. Ce seuil devait dans la logique diminuer avec l'augmentation de la mobilité. Soit la mobilité n'était pas assez forte pour qu'elle perturbe le routage ou bien, le maillage du réseau est fait d'une telle façon que les nœuds mobiles ne jouent pas des rôles majeurs dans le routage. L'intensité du flux a vérifié la règle de diminution des paquets utiles. Par contre la courbe des trois protocoles voit une chute importante entre les deux premières valeurs. Il semble que le premier flux (9,68 Koctet/sec/nœud) soit un flux presque idéal pour véhiculer un maximum de paquets. Au delà de ce flux, les protocoles génèrent plus de control, perdent et délivrent moins de paquets.

### 5.1.3 Paquets perdus (pqt/nœud)

Pour ce scénario nous avons deux courbes. La première illustre le nombre de paquets de perdus en fonction de la mobilité (Fig. 5.3-A) et la deuxième illustre le nombre de paquets perdus en fonction de l'Intensité du flux sortant (Fig. 5.3-B).



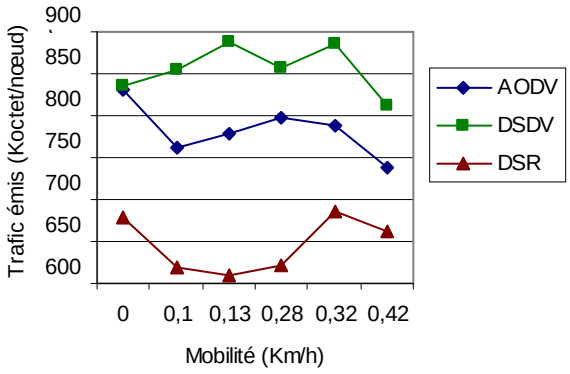

La différence est bien visible sur les deux courbes entre les trois protocoles. Sur la première figure AODV se retrouve le numéro 1 des paquets perdus. DSDV assure moyennement la préservation des paquets. DSR a une perte presque négligeable. Ces deux derniers ont un comportement presque identique malgré l'augmentation de la mobilité. Par contre, AODV perd de plus en plus de paquets avec une forte mobilité à cause des chemins qui ne restent plus valides.

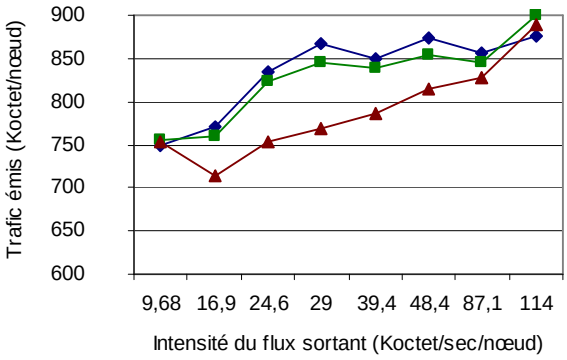
Nous pouvons presque appliquer la même analyse à la deuxième figure, où avec l'augmentation des flux sortants, AODV perd plus de paquets. Sa moyenne de perte est nettement supérieure à celle des deux autres protocoles qui ont une courbe quasi identique à une constante près.

La logique théorique s'attend à une perte croissante avec la mobilité ainsi qu'avec des flux sortants importants. DSDV et DSR semblent garder une moyenne fixe des paquets perdus, qui doit être au détriment du control qui assure une bonne liaison avant d'envoyer les paquets. AODV valide la théorie.

### 5.1.4 Trafic émis (Koctet/nœud)

Pour ce scénario nous avons deux courbes. La première illustre le trafic émis en fonction de la mobilité (Fig. 5.4-A) et la deuxième illustre le trafic émis en fonction de l'Intensité du flux sortant (Fig. 5.4-B).



La première figure nous montre les deux protocoles AODV et DSDV qui ont presque la même tendance de courbe mais DSDV émet plus que AODV du fait qu'il trouve rapidement les chemins et le medium est plus libre. Par contre pour les réactifs AODV et DSR, il leur faut plus de temps pour trouver un chemin. La mobilité ne semble pas beaucoup influencer le trafic émis pour tous les protocoles simulés.

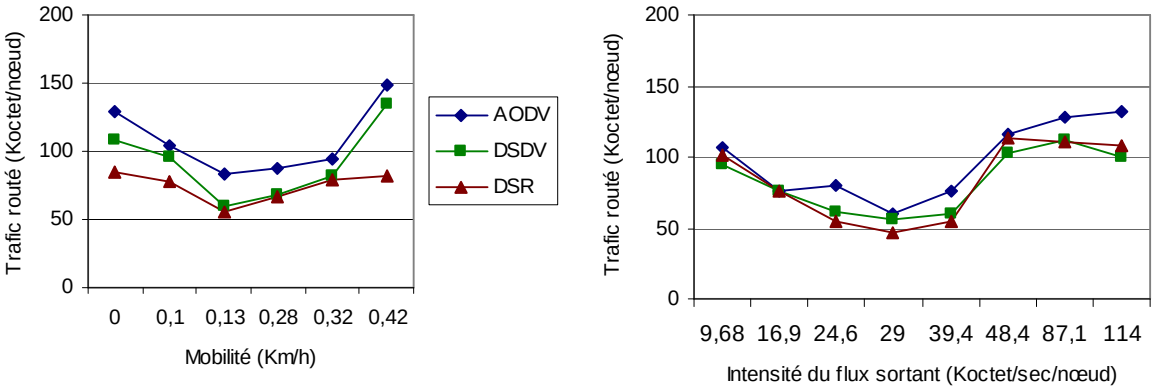
La deuxième figure présente trois courbes assez voisines en hausse avec l'augmentation de l'intensité du flux sortant. DSR émet moins que les deux autres protocoles toujours pour la même précédente raison liée au temps de découverte de chemins.

Nous remarquons bien que la mobilité et l'intensité du flux sortant n'impactent pas les protocoles de la même façon. DSDV se retrouve mieux que AODV pour une grande mobilité et inversement quand il s'agit d'intensité de flux sortant.

En théorie, la hausse de la mobilité ralentit le trafic de données et réduit le volume de paquets émis. Par contre, une grande intensité de flux sortant, bien qu'elle engendre des saturations au niveau des files d'attente, implique une grande taille des paquets qui étend les volumes des données émises. Cette théorie est bien exprimée dans la deuxième courbe. Contenant la mobilité, une aberration pour le protocole réactif DSR figure dans la première courbe, où la mobilité à 0,32 Km/h représente le maximum de trafic émis alors que toutes les mobilités moins importantes donnent des volumes de données émises beaucoup plus faibles. L'explication possible pour ce cas de figure est qu'à cette valeur de mobilité, les nœuds dynamiques n'ont pas joué un rôle avantageux dans le routage ni dans les émissions/réceptions. Ceci pourrait être justifié par une mobilité située aux extrémités du réseau, là où le trafic de données ne présente pas une densité notable.

### 5.1.5 Trafic routé (Koctet/nœud)

Pour ce scénario nous avons deux courbes. La première illustre le trafic routé en fonction de la mobilité (Fig. 5.5-A) et la deuxième illustre le trafic routé en fonction de l'Intensité du flux sortant (Fig. 5.5-B).



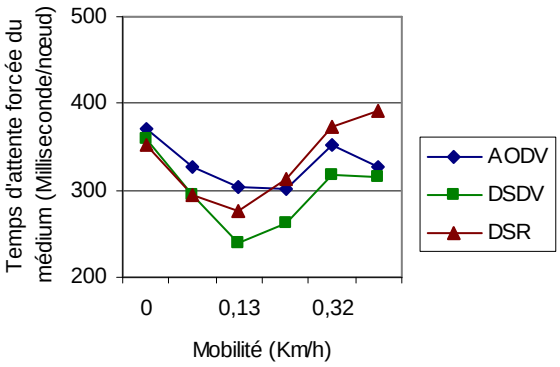

Sur la première figure, AODV utilise plus de routage que les deux autres puisqu'il est réactif. Donc ses chemins ne sont pas optimaux. Le proactif DSDV se situe entre les deux réactifs, gardant ainsi une bonne moyenne. DSR utilise le minimum de routage grâce à son algorithme qui lui procure des chemins courts à la source.

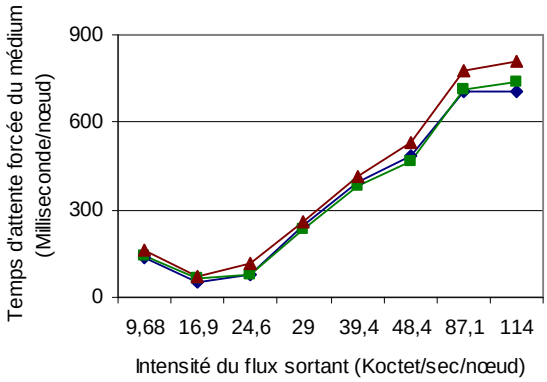
Au niveau de la deuxième figure, les courbes très voisines gardent quasiment les mêmes positions. Les deux figures attribuent une idée commune aux protocoles.

La théorie des algorithmes ad hoc veut que le routage augmente avec la mobilité ainsi qu'avec de larges flux sortants, du fait que la mobilité favorise la disparition des chemins valides. Quand les flux sortants montent, les nœuds routeurs se saturent et les chemins se dilatent. L'algorithme qui calcule le plus court chemin doit bien se comporter dans des scénarios pareils. Les trois protocoles semblent avoir les mêmes performances pour ce calcul. Jusqu'à la mobilité 0,13 Km/h ils semblent bien gérer le routage minimum. Juste après ce seuil, le routage commence à prendre plus d'ampleur. Ce même scénario est visible aussi pour les trois protocoles sur la deuxième courbe. Jusqu'à 29 Koctet/sec/nœud, le routage se décline. Après cette valeur, le trafic routé prend du volume à cause de la mauvaise gestion des chemins courts ou bien de la saturation des nœuds routeurs qui n'arrivent pas à subvenir aux besoins des nœuds qui les sollicitent.

### 5.1.6 Temps d'attente forcé du médium (Milliseconde/nœud)

Pour ce scénario nous avons deux courbes. La première illustre le temps d'attente forcé du médium en fonction de la mobilité (Fig. 5.6-A) et la deuxième illustre le temps d'attente forcé du médium en fonction de l'intensité du flux sortant (Fig. 5.6-B).




Dans la première figure, les trois protocoles augmentent leur temps d'attente du médium avec l'augmentation de la mobilité après la mobilité 0,13 Km/h. DSR semble le plus sensible à ce phénomène, où il se retrouve au maximum du temps d'attente pour une forte mobilité.

Pour la deuxième figure, tous les protocoles se comportent de la même façon avec des courbes quasi identiques, où leurs temps d'attente du médium augmentent avec l'augmentation de l'intensité des flux de données. La simulation confirme la contrainte physique du support de transmission qui doit être utilisé que par un seul émetteur à la fois.

Sur un plan théorique, la mobilité amplifie le temps d'attente du médium. L'intensité du flux sortant aussi agit de la même façon sur le temps d'attente d'un nœud pour émettre des paquets de données via les ondes radios. La deuxième courbe confirme bien cette règle pour les trois protocoles avec une toute petite baisse entre 9,68 et 16,9 Ko/sec/nœud. Dans cet intervalle, les protocoles parviennent à réduire les temps d'attente puisque les flux sont bien gérés et optimisés sur le plan longueur des chemins. Au delà de cette valeur optimale, les temps d'attente augmentent avec l'intensité des flux sortant du fait que le médium est considérablement employé. Même scénario pour la première courbe de mobilité mais d'une façon plus flagrante. La baisse des temps d'attente vont jusqu'au 0,13 Km/h pour remonter aussitôt après, sauf pour le réactif AODV qui garde cette valeur jusqu'à 0,28 Km/h. Cette mobilité exprime le minimum du temps d'attente pour la libération du médium. Le rapport entre les flux sortants, la longueur des chemins et leur degré de fiabilité doit être très intéressant et optimal.

## 5.2 Conclusion et perspectives

### Conclusion

La majorité des protocoles de routage ad hoc ont des propriétés communes et d'autres spécifiques pour chacun. Une configuration de ces propriétés peut définir si le protocole est performant ou non. Une étude consacrée aux propriétés spécifiques nous révèle les vrais écarts entre ces protocoles. Ces derniers peuvent diverger en fonction des scénarios appliqués.

Dans notre étude nous avons pu déceler quelques écarts entre les trois protocoles simulés AODV, DSDV et DSR ainsi que d'autres entre les types de protocoles proactifs et réactifs. Ces écarts sont loin d'être exhaustifs par rapport à la réalité des ces protocoles divers et variés.

Le protocole réactif AODV se comporte globalement d'une manière assez satisfaisante par rapport aux deux autres. Il craint les fortes mobilités ainsi que les flux de données importants où il enregistre une perte de paquets considérable. En le comparant à DSR qui est de la même catégorie nous avons constaté qu'ils ont presque les mêmes tendances sur un ordre général. Le proactif DSDV se distingue par rapport aux autres par sa particularité de garder à jour une table de routage. Nous avons noté qu'il commence à émettre réellement très vite après la demande. Ces routes sont toujours à disposition même dans les fortes mobilités. Par rapport aux taux d'acheminement des paquets il assure un très bon seuil. La quasi-totalité des paquets arrive à destination.

Le meilleur comportement que nous avons enregistré est celui de DSR. Cet algorithme proactif requiert un minimum de paquets de control pour acheminer un maximum de données avec une perte presque négligeable. La diversité des scénarios appliqués à cet algorithme, ne semble pas beaucoup l'impacter. Il arrive à garder presque toutes ses valeurs pour une grande mobilité ou dans un réseau dense. Son attente de la libération du médium augmente avec la mobilité du fait que les routes disparaissent rapidement. Chacun des nœuds émetteurs initie une demande de nouvelle route.

Sur un ordre général, DSR emporte la première place par rapport à tous les résultats que nous avons pu enregistrer durant nos simulations, même si les deux autres le dépassent dans quelques propriétés précises.

**Perspectives**

Durant les deux phases de notre étude théorique et pratique, nous nous sommes vite rendus compte que le domaine de recherche dans les réseaux sans fils est tout jeune et qu'il a un long chemin à parcourir. Plusieurs groupes scientifiques s'intéressent à ce type de réseaux qui promet beaucoup pour l'avenir. Plus spécialement, les protocoles de routage attirent de plus en plus les simulateurs afin de combiner les meilleurs atouts de chacun d'eux en un seul. Après notre étude théorique nous avons sélectionné des propriétés afin de les simuler et trouver les écarts entre les algorithmes de routage.

Ce travail de préparation des simulations nécessite beaucoup de recherche et de développements. Notre modeste contribution a besoin d'être élargie avec des simulations sur d'autres paramètres. De nouvelles métriques de comparaisons sont à envisager pour mieux accentuer les différences. Les études de simulations sur un seul protocole auquel on apporte des changements sur son algorithme de routage pourraient nous révéler de nouvelles particularités. Ainsi, nous allons permettre aux algorithmes de progresser et les réseaux sans fils de voir de nouveaux terrains d'applications. La téléphonie sur IP, la visiophonie et les applications multimédias sont tous à l'attente de tels types de réseaux sans fils mais avec des bandes passantes beaucoup plus larges et des temps de réponses très courts. La majorité des applications actuelles exigent le maintien d'une qualité de service minimale.

## Bibliographie


[1] S.Corson, University of Maryland, J. Macker, Naval Research Laboratory."Mobile Ad hoc Networking (MANET)". RFC 2501, January 1999.

[2] G.Malkin, Xylogics, Inc., F.Baker, Cisco Systems."RIP Version 2 MIB Extension". RFC 1724, November 1994.

[3] Kevin Fall,Kannan Varadhan. "The ns Manual". UC Berkeley, LBL, USC/ISI, and Xerox PARC, avril 2001

[4] J. Moy,Cascade Communications Corp. "OSPF Version 2. RFC 2178", July 1997.

[5] Elizabeth M. Royer, Chai-Keong Toh. "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks". IEEE Personal Communications, April 1999.

[6] T.Larsson, N.Hedman. "Routing protocols in wireless ad hoc networks: a simulation study". Décembre 1998

[7] Daniel L. Lough, T. Keith Blankenship, Kevin J. Krizman."A Short Tutorial on Wireless LANs and IEEE 802.11".Summer 1997.

[8] http://www.ietf.org/

[9] http://www.ee.surrey.ac.uk/Personal/G.Aggelou/MANET_PUBLICATIONS.htm

[10] Ajay Chandra V. Gummalla, John O. Limb. "Wireless Medium Access Control Protocols". IEEE Communications Surveys, Second Quarter 2000.

[11] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das. s"Ad hoc On-Demand Distance Vector (AODV) Routing". Internet Draft, 2 March 2001.

[12] David B. Johnson, David A. Maltz, Yih-Chun Hu, Jorjeta G. Jetcheva. "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks". Internet Draft, 2 March 2001.

[13] V. Park, S. Corson. "Temporally-Ordered Routing Algorithm (TORA) Version 1". Internet Draft, 20 July 2001.

[14] Charles Perkins, Pravin Bhagwat. "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers ". 1994

[15] M. S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, A. Qayyum."An Internet MANET Encapsulation Protocol (IMEP) Specification". Internet Draft, 7 August 1999.

[16] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, Anis Laouiti, Laurent Viennot, Thomas Clausen."Optimized Link State Routing Protocol". Internet Draft, 2 March 2001.

[17] C.-C. Chiang."Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel". Proceedings of IEEE SICON'97, Apr.1997.

[18] Zygmunt J. Haas, Marc R. Pearlman. "The Zone Routing Protocol (ZRP) for Ad Hoc Networks". Internet Draft, November 1997.

[19] Anis KOUBAA. "Gestion de la Qualité de Service temporelle selon la contrainte (m,k)-firm dans les réseaux à commutation de paquets". 2004.

[20] Simulateur de réseaux - http://www.reseaucerta.org/outils/simulateur/

[21] GlomoSim - http://pcl.cs.ucla.edu/projects/glomosim/

[22] NS: The Network Simulator - http://www.isi.edu/nsnam/ns/

[23] OPNET: Open NETwork - http://www.opnet.com/

[24] Une approche efficace de détection des intrusions pour le protocole OLSR des MANET - http://www.crc.ca/fr/html/manetsensor/home/publications/abstracts

[25] Ying Ge, Louise Lamont, Luis Villasenor. "A Scalable Proactive Routing Protocol for Heterogeneous Ad Hoc Networks". WiMob 2005 Wireless and Mobile Computing, Montréal, Canada, août 2005.

[26] Claude Chaudet. "Simulateurs". http://www.infres.enst.fr/~chaudet/index.php/CaptAdhoc/Simulateurs

[27] VINT : Virtual InterNetwork Testbed - http://www.isi.edu/nsnam/vint/

[28] Fred L. Drake. “Python Tutorial”. Release 2.1.1, July 20, 2001

[29] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, Haobo Yu. “Network Visualization with the VINT Network Animator Nam”. Tech. Rép. 1999.

## Annexe A

### Description de la simulation

Le processus de simulation est composé de trois phases essentielles :

- Phase de préparation : s'occupe de la génération des fichiers d'entrées
- Phase de simulation : lance les simulations et génère les traces
- Phase d'analyse : Analyse les traces et génère des courbes

La figure A.1 présente le processus général d'une simulation sous ns.

Dans la première phase de préparation, les fichiers d'entrées de la simulation sont générés par un programme que nous avons implémenté en C. Ces fichiers sont classés en deux catégories :

1. Fichiers de scénario qui décrivent les nœuds, leurs positions ainsi que leurs mouvements
2. Fichiers de communication qui décrivent le trafic dans le réseau

Une fois la simulation lancée en deuxième phase, elle prend comme entrée les deux fichiers scripts (Otcl) et génère en sortie un fichier journal appelé aussi la trace qui décrit les opérations et les transactions effectuées à travers le temps au sein du réseau.

A ce moment, la phase d'analyse peut débuter en prenant la trace comme entrée pour le programme d'analyse que nous avons implémenté en langage Python [28].

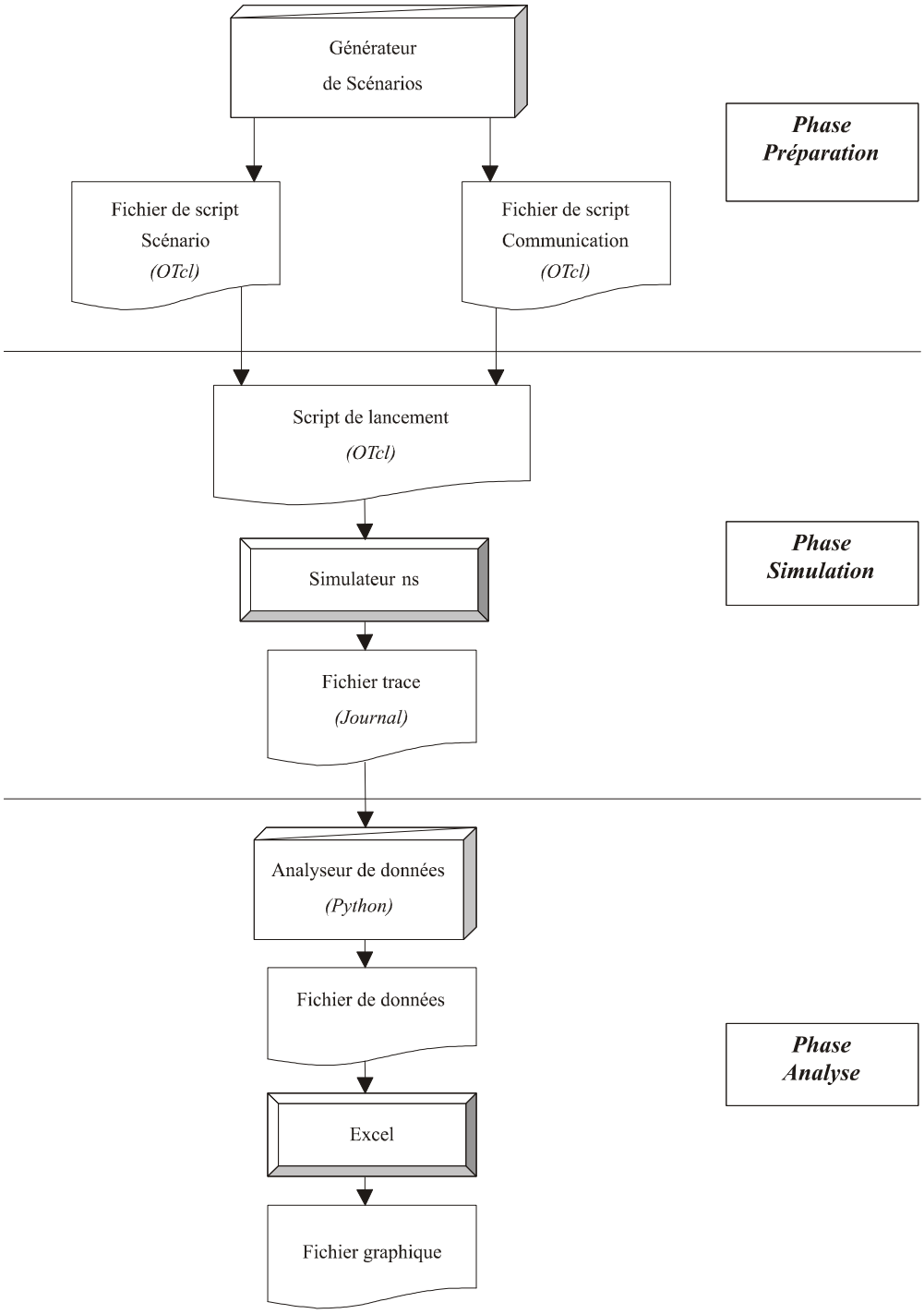



*FIG A.1 - Processus général de simulation.*

## Annexe B

## The Network Animator

La conception de protocole demande une compréhension de plusieurs détails, dont le suivi des états d'un grand nombre de nœuds, une analyse de l'échange de messages et doit caractériser les interactions dynamiques pour des trafics concurrents. Habituellement, des traces de paquets sont utilisées pour accomplir ces tâches. Cependant, ces traces ont deux inconvénients majeurs. Elles présentent un nombre important de détails, ce qui peut compliquer la compréhension des données, et elles sont statiques, ce qui cache une dimension importante du comportement des protocoles. Les outils de visualisation adressent ce problème en permettant à l'utilisateur de prendre en considération plusieurs informations très rapidement, d'identifier visuellement les modèles de communication et de mieux comprendre les interactions et les causalités. NAM [29] est un outil d'animation basé sur Tcl/TK[45] pour l'observation des traces de paquet. Il peut être installé sur un système de type unix ou sur Windows 95/98/NT ayant Microsoft Visual C++ installé. Les données utilisées par NAM peuvent provenir d'un simulateur ou de tests sur des réseaux réels. Il supporte l'affichage de la topologie, l'animation des échanges de paquets et des outils d'inspection de données divers. NAM a été crée par le laboratoire LBL et s'est considérablement développé durant les dernières années.

NAM interprète un fichier de trace contenant des évènements réseau indexés par le temps de différentes manières. Ces évènements sont principalement les arrivées, départs et suppressions de paquets et ruptures de lien. Pour les simulations de réseau sans fil, la localisation et les mouvements des nœuds s’ajoutent aux évènements interprétés.

NAM est exécuté avec comme paramètre le fichier enregistré. Lorsqu'on exécute NAM, une fenêtre de travail NAM est créée. Il est possible de faire tourner plusieurs animations avec une seule instance, ce qui permet de mieux comparer certain protocole.

La figure B.1 représente une fenêtre d'animation NAM. On peut entre autre régler le pas de la simulation (de 8µs à 800ms), zoomer sur des zones de la simulation, et manipuler la lecture : on peut mettre pause à tout moment ce qui donne un « arrêt sur Image », revenir, avancer sur les étapes de la simulation ce qui permet d'examiner des occurrences particulières. Cette animation comporte 20 nœuds avec un trafic TCP entre les nœuds 13 et 17.

[45] TCL/TK : Tool Command Language, TK: Toolkit (La bibliothèque graphique de TCL).

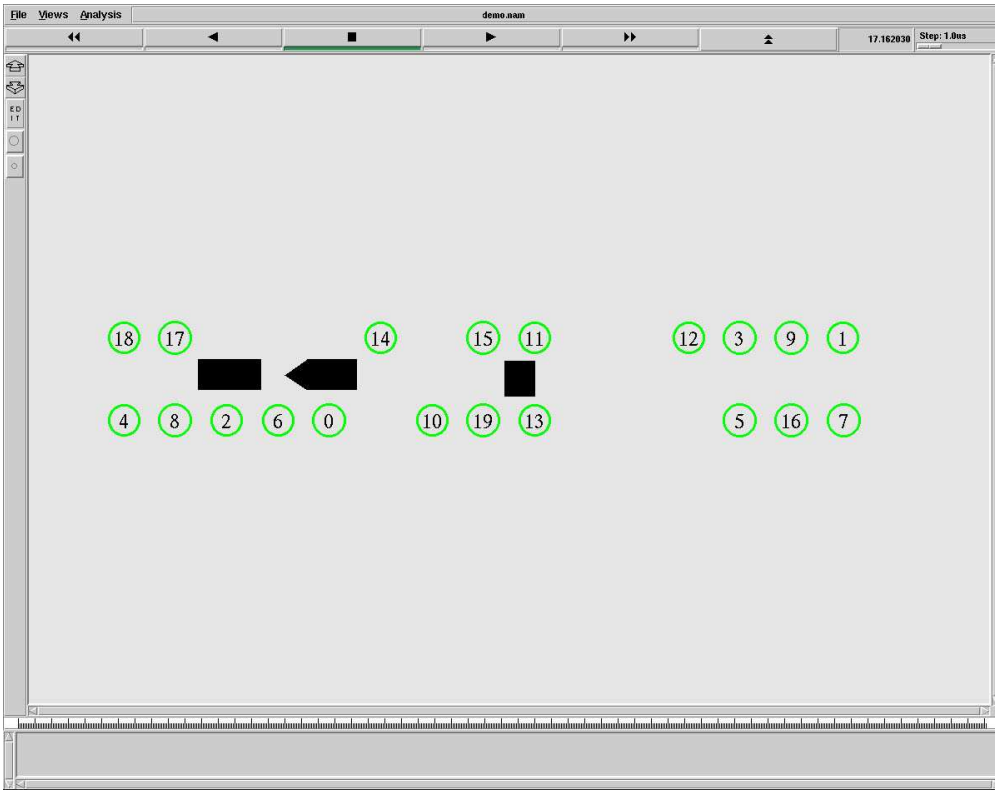



*FIG B.1 - Une capture d'écran du Network Animator.*