UNIVERSITE KASDI MERBAH OUARGLA
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique et des Technologies de l'information

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

Domaine : Informatique et Technologie de l'Information
Spécialité : Informatique Industriel

Présenté par : Berriha Saliha
Miloudi Takoia

Thème :

Etude comparative de deux simulateurs pour les réseaux ad-hoc sans fil

Soutenu publiquement Le 25/06/2018

Devant le jury:

Mme. Wassila KORICHI	Encadreur	UKM Ouargla
Mr. A.K .benmir	Examinateur	UKM Ouargla
Mr M.Mezati	Examinateur	UKM Ouargla

Année universitaire: 2017/2018

Remerciements

LOUANGE À Dieu, sans lui rien de tout cela n'aurait pu être.

Au terme de la rédaction de ce mémoire, nous tenons à remercier notre encadreur Meme Korichi pour ses précieux conseils et son aide durant toute la période du travail. Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions. On n'oublie pas de remercier nos parents, nos frères et sœurs pour leur soutien moral et physique. Pour les merveilleux moments qu'on a passé ensemble, et à tous nos amis sans exception.

Dédicaces

Je dédie ceMémoire …

À mes chers parents, qui sont la cause de mon existence dans cette vie,pour leur soutient, leur patience et leur amour qui m'ont donné la force pour continuer mes études Aucune dédicace, aucun mot ne pourrait exprimer à leur juste valeur la gratitude et l'amour que je vous porte, Aujourd'hui, Je mets entre vos mains, le fruit de longues années d'études, À ma soeur Fatiha et à mon frére Zaki Je leurs souhaite une bonne continuation dans leurs vie,et touts la famile BERRIHA et MILOUDUI À et à tous mes amis du proche ou du loin sans exception.

Merci

Résumé :

Un réseau mobile ad hoc est un ensemble de nœuds mobiles connectés par des liens radio sans fil. Ces environnements présentent l'avantage d'un déploiement rapide et peu coûteux puisqu'ils ne nécessitent l'installation d'aucune infrastructure. Beaucoup de travaux de recherche se sont penchés dans le domaine des réseaux mobile ad hoc, pour développer de nouveaux protocoles et nouvelle application, qui utilisant les particularités sans fil sont proposés.
Afin de tester l'efficacité d'un nouveau protocole il est indispensable d'étudier son comportement dans un environnement proche de l'environnement réel. Cependant la reproduction très coûteuse d'un tel système a motivé la naissance des simulateurs de réseaux.
La simulation permet non seulement de tester sans aucun coût de nouvelles technologies et de nouveaux protocoles, mais aussi d'anticiper les problèmes qui pourraient se poser dans le futur. Les environnements de simulation nous offrent un milieu d'exécution très proche de la réalité (dans la limite du possible), pour obtenir des résultats permettant d'envisager des corrections utiles et constructives**.** Ce travail entre dans le cadre de l'étude des simulateurs dans les réseaux Ad Hoc, notre étude consiste à : comparer les deux simulateurs, « NS-2 » et « OMNET++ » qui ont été testé par plusieurs scientifiques.

Mots clés : Réseaux ad hoc ; la simulation; simulateur NS2 ; simulateur OMNET++.

Abstract :

An ad hoc mobile network is a set of mobile nodes connected by wireless radio links. These environments have the advantage of a fast and inexpensive deployment since they do not require the installation of any infrastructure. A lot of research work has been done in the field of ad hoc mobile networks, to develop new protocols and new application, which using the wireless features are proposed.

In order to test the effectiveness of a new protocol it is essential to study its behavior in an environment close to the real environment. However, the very expensive reproduction of such a system motivated the birth of network simulators.

Simulation not only makes it possible to test new technologies and protocols without cost, but also to anticipate problems that may arise in the future. Simulation environments offer us an environment very close to reality (as far as possible), to obtain results allowing to consider useful and constructive corrections. This work is part of the study of simulators in Ad Hoc networks, our study consists in: comparing the two simulators, "NS-2" and "OMNET++" which have been tested by several scientists.

Keywords: Ad hoc networks; the simulation; NS2 simulator; OMNET ++ simulator.

Table des matières

Introduction générale : I
Chapitre I : Les Réseaux mobiles ad hoc 2
1. Introduction 2
2- Les environnements mobiles 2
2.1 Définition d'un environnement mobile 2
2.2 Le modèle des environnements mobiles 2
2.3 Les réseaux mobiles avec infrastructure : 3
2.4 Les réseaux mobiles sans infrastructure (Ad hoc) : 4
3- La communication cellulaire 5
4- L'utilisation des ondes radio dans la communication sans fil 6
5-La fiabilité de la communication sans fil 6
6-Quelques éléments de l'infrastructure sans fil 7
7- Les réseaux mobiles ad hoc 7
7.1 Définition 7
7.1.1 Un exemple d'un réseau ad hoc 8
7.1.2 Modélisation 8
7.1.3 Les modes de communication dans les réseaux mobiles ad hoc 9
7.2 Les caractéristiques des réseaux Ad hoc 10
7.3 Les applications des réseaux mobiles Ad Hoc 11
7.4 Les avantages et Les inconvénients des réseaux ad hoc 11
7.4.1 Les avantages: 11
7.4.2 Les inconvénients 12
8- Les protocoles de routage dans MANET 12
8.1 les protocoles réactifs 13
8.2 Les protocoles pro-actifs 13
8.3 Les Avantages et les Inconvénients des protocoles de routage dans MANET [2] 14

8.3.1 Les Avantages 14
8.3.2 Inconvénients 14
9. Conclusion 14
Chapitre II: Présentation des deux simulateurs« NS-2 » et « OMNET++» 17
I. Introduction 17
II. Environnement de Simulation 17
1. Introduction à la simulation 17
.2 Système réel et objectif de simulation 17
.3 Modèles de simulation 18
3.1 Gestion du temps 19
3.2 Simulation par événements discrets 19
3.3 Simulateur 19
4. Avantages et inconvénients de la simulation 20
II NS-2 (Simulator Network 2) 20
1- Présentation de network simulator NS-2 20
2-Le langage TCL (Tool Command Language) 21
2.1 Otcl 21
3-NAM 22
4-Le modèle de réseau sous NS-2 22
5-Caractéristiques d'une entité communicante sous NS-2 23
5.1 - Le Classifier 23
5.2- Le lien 23
5.3 - L'agent 24
6-Le processus de simulation 25
7-Création d'un scénario 26
8-L'avantage et inconvénients du simulateur NS-2 28
8.1 Les Avantages du simulateur NS-2 [10] : 28
8.2 Les inconvénients : 29
III. Le simulateur OMNET++ 29
1-Présentation OMNET++ 29
2.Architecture d'OMNET++ 30
3.Les principaux fichiers d'OMNET++ 31

3.1Fichier(.NED): 31
3.2Fichier(.ini) : 32
3.3Fichier(.msg) 33
4 Composant : 34
5 Framework : 34
5.1 Mobility Framework : 34
5.1.2 Les modules de mobilités : 35
5.1.3 la structure d'un hôte mobile : 35
6 INET Framework : 36
6.1 Catalogue de modèles : 38
7. Structure d'un nœud mobile dans OMNET++ : 39
8 L'avantage et inconvénients du simulateur OMNET++ [16] : 39
8.1 Avantages 39
8.2 Inconvénients 39
IV. Conclusion 40
Chapitre III : Comparaison qualitative entre « NS-2 » & « OMNET++ » 42
I.Introduction 42
1.Placement des noeuds 42
2.Scénario de mobilité 43
3.Terrain physique 43
4.Génération de trafic 43
5. Modules implémentés 44
II. Conclusion 46
Chapitre IV : Comparaison qualitative entre « NS-2 » & « OMNET++ » 48
I. Introduction 48
II. Paramètres de simulation 48
1. Portée de transmission 49
1.1 Description 49
1.2 Résultats de simulation 49
1.2.1 Appliqué le scénario dans le simulateur NS-2 50
1.2.2 Appliqué le scénario dans le simulateur OMNET++ 52
1.3 Etude comparative 54

2. Méthode d’accès au medium 55
2.1 Description 55
2.2 Résultats de simulation : 55
2.2.1 Appliqué le scénario dans le simulateur NS-2 : 56
2.2.2 Appliqué le scénario dans le simulateur OMNET++ : 58
2.3 Etude comparative :..... 59
3.Scalabilité, simulation et autres facteurs..... 62
3.1. Description 62
3.2. Résultats de simulation 62
3.2.1 Appliqué le scénario dans le simulateur NS-2 : 62
3.2.2 Appliqué le scénario dans le simulateur OMNET++ : 63
3.3. Etude comparative :..... 63
Conclusion 65

Liste de figure :

Figure 1- La décomposition des réseaux mobiles [3]. 3
Figure 2-Les réseaux mobiles avec infrastructure.[2] 3
Figure 3-Les réseaux mobiles avec infrastructure[2] 4
Figure 4-Le changement de la topologie des réseaux Ad hoc[2] 5
Figure 5-Le principe de réutilisation de fréquence[3] 5
Figure 6-La modélisation d'un réseau ad hoc[2] 9
Figure 7-Les différents modes de communication [2] 9
Figure 8-Classification des protocoles[3]. 12
Figure 9-L'architecture simulation du NS2 22
Figure 10-Schéma d'un nœud dans NS2 24
Figure 11-Le processus de simulation NS-2 26
Figure 12-Le lancement du simulateur OMNET++ 29
Figure 13-Architecture modulaire du simulateur OMNET++ 30
Figure 14-Fichier Ned en mode text 31
Figure 15-Ficher Ned en mode graphique 32
Figure 16-Exemple d'un fichier *.ini 33
Figure 17-Exécution d'une simulation sous OMNET++[Site4] 33
Figure 18-Archetecture de Mobility et Channel Control 35
Figure 19-Architecture interne d'un nœud dans MF 36
Figure 20-Structure d'un nœud mobile dans OMNET++ [14] 39
Figure 21-Scénario pour la vérification de la portée de transmission 49
Figure 22-scénario1.tcl 50
Figure 23-Exécution fichier. tcl 50
Figure 24-NAM, visualisation représenter la topologie d'un réseau décrit avec NS-2 51
Figure 25-NAM, visualisation représenter la topologie d'un réseau décrit avec NS-2 51
Figure 26-Traces de la simulation 52
Figure 27-Ficher Ned en mode graphique 52
Figure 28-Fichier Ned en mode text 53
Figure 29-Fichier configuration .ini 53
Figure 30-TKenv, visualisation représenter la topologie d'un réseau décrit avec OMNET++ 54
Figure 31-Nombre de paquets reçus par le mobile N1 en fonction de la distance (N0 ,N1) 54
Figure 32-Scénario de test de la méthode d'accès 56
Figure 33-Le script fichier.tcl définition des variables 56
Figure 34-scénario2.tcl 56
Figure 35-fichier.tr 57
Figure 36-Fichier configuration .ini 58
Figure 37-Fichier Ned en mode text 58

Figure 38-Ficher Ned en mode graphique 59
Figure 39-TKenv, visualisation représenter la topologie d'un réseau décrit avec OMNET++..... 59
Figure 40- scénario3.tcl définie les variable pour les paramètres..... 62
Figure 41-fichier.ini config paramater..... 63
Figure 42-Taux paquet perdu en fonction de la mobilité..... 64
Figure 43-Taux paquets perdus en fonction du débit 65

Introduction générale :

L'internet est le réseau des réseaux informatiques communicants entre eux grâce à un protocole de communication commun. Toutefois, la croissance exponentielle d'Internet a induit la naissance des nouvelles technologies de réseaux informatiques notamment le réseau sans fil(en anglais : Wireless network) Ce dernier est un réseau informatique qui connecte différents postes ou systèmes entre eux par onde radio sans qu'aucune connexion « filaire » ils se basent sur une infrastructure où le maintien de la connectivité est ménagé par des équipements spécifiques citant des routeurs wifi et des points d'accès. Les réseaux sans fils peuvent être classés en deux catégories : les réseaux avec infrastructure qui utilisent généralement le modèle de la communication cellulaire, et les réseaux sans infrastructure ou les réseaux ad hoc

Dans les réseaux avec infrastructure, les communications s'effectuent via une station de base fixe. Cette approche est utilisée dans les réseaux sans fil traditionnels comme les réseaux GSM (Global System for Mobile communications), et les réseaux locaux sans fil. Cependant, les communications dans un réseau Ad Hoc s'effectuent en absence de toute infrastructure de communication fixe. Si les équipements communicants se trouvent dans la zone de transmission les uns des autres, la communication s'effectue en mode point-à-point ; Si encore les nœuds communicants sont éloignés, plusieurs nœuds intermédiaires participent à l'acheminement des données vers leurs destinations. On parle, dans ce cas, du routage "multi sauts". Les réseaux Ad Hoc multi-sauts suscitent beaucoup d'intérêt grâce à leur déploiement rapide et économique et à leur nature décentralisée.

Pour pouvoir tester et évaluer des travaux dans cet environnement, Il existe différentes techniques. Parmi elles, on peut citer la modélisation analytique, les mesures obtenues à partir d'expériences réelles, et la simulation, il faut que les modèles sur lesquels se base le simulateur soient le plus proches possible de la réalité.

Cependant, choisir un simulateur parmi les autres est un problème, sur quelle base et comment un chercheur choisir ce simulateur n'est non pas l'autre ?

Dans ce travail nous allons fait une étude comparative entre les deux simulateurs de réseaux les plus utilisé par la communauté de chercheurs travaillant sur les environnements mobiles sans fil, qu'ils sont certainement « NS-2 » et « OMNET++ »

Ce document est composé de quatre chapitres. Le premier chapitre présente les environnements mobiles et en particulier les environnements mobiles ad hoc. Il met en évidence les caractéristiques des communications sans fil et leurs conséquences sur le fonctionnement de ces réseaux. Les problèmes que posent ces environnements pour l'implantation de nouveaux services, sont alors clairement définis. Le deuxième chapitre nous décrire les caractéristiques et les capacités des deux simulateurs « NS-2 » et « OMNET++ ».

Le troisième chapitre consiste à comparer les deux simulateurs d'une manière préliminaire ou qualitative des deux simulateurs. Dans le dernier chapitre, nous procurions aux simulations pour pouvoir compares les deux simulateurs en termes de résultats et de temps de simulation.

Chapitre I:

Les Réseaux mobiles ad hoc

Chapitre I : Les Réseaux mobiles ad hoc

1. Introduction

L'évolution récente de la technologie dans le domaine de la communication sans fil pousse aujourd'hui les chercheurs à faire des efforts pour mieux assurer la fonction des réseaux, à savoir l'accès rapide à l'information indépendamment du lieu et du temps. Dans le passé, les réseaux sans fil se basaient sur des infrastructures planifiées. Avec la croissance importante qu'ont connue les réseaux sans fil, des besoins en termes d'auto-organisation, d'indépendance et de réduction de couts se sont réalisés. Les réseaux mobiles ad hoc répondent à ces problèmes.

Contrairement aux réseaux sans fil basés sur la communication cellulaire, les réseaux ad hoc n'ont aucune administration centralisée, ce sont les hôtes mobiles eux-mêmes qui forment une infrastructure du réseau. Aucune limitation n'est faite sur la taille des réseaux ad hoc, le réseau peut contenir des centaines ou des milliers de nouds mobiles.

Dans ce chapitre, nous allons présenter les deux types de réseaux mobiles selon leurs architectures : réseaux avec infrastructure et réseaux sans infrastructure (ad hoc). Ensuite, nous énumérons les différentes caractéristiques des réseaux ad hoc et nous citerons quelques applications.

2- Les environnements mobiles

2.1 Définition d'un environnement mobile

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

2.2 Le modèle des environnements mobiles

La classification des réseaux sans-fil peut être menée selon le mode opératoire du réseau et on distingue ainsi deux modes : Le mode avec infrastructure qui utilise généralement le modèle de la communication cellulaire, et le mode sans infrastructure ou mode ad hoc.

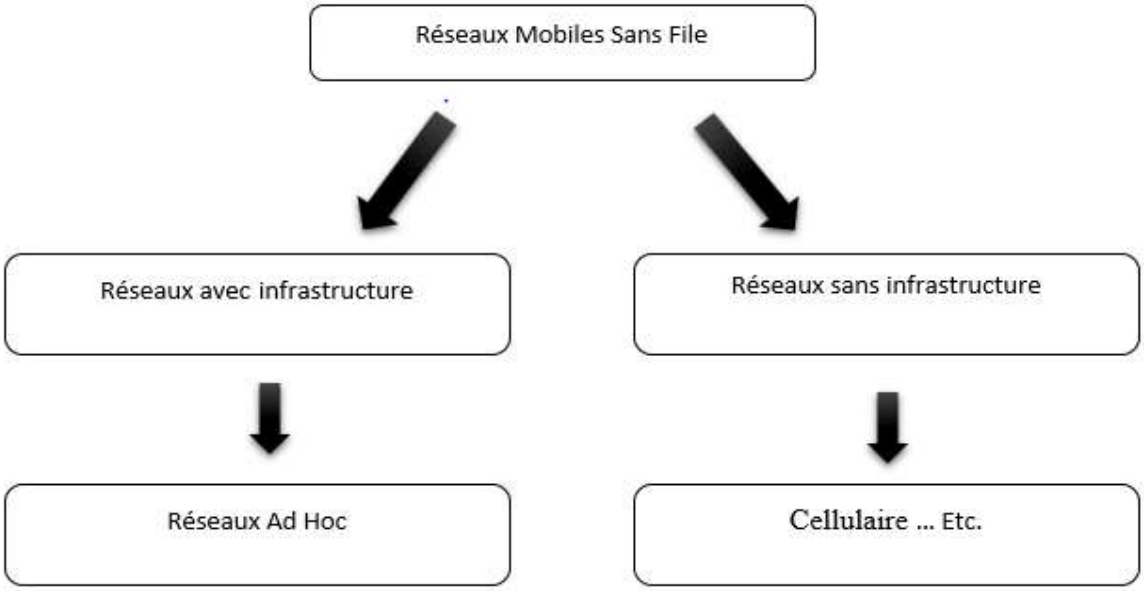

Figure 1- La décomposition des réseaux mobiles [3].

2.3 Les réseaux mobiles avec infrastructure :

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

- Les " sites fixes " d'un réseau de communication filaire classique (Wired network).
- Les "sites mobiles" (Wireless network).

En mode avec infrastructure, également appelé le mode BSS (Basic Service Set) Certains sites fixes, appelés stations de base (SB) sont munis d'une interface de communication sans fil pour la communication directe avec les sites mobiles ou unité mobile (UM) localisés dans une zone géographique limitée, appelée cellule comme le montre la figure (2).

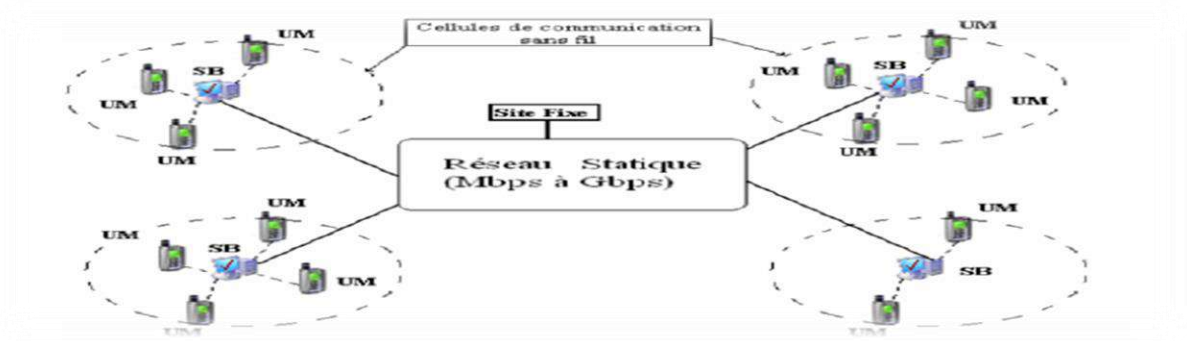

Figure 2-Les réseaux mobiles avec infrastructure.[2]

A chaque station de base correspond une cellule à partir de laquelle des unités mobiles peuvent émettre et recevoir des messages. Alors que les sites fixes sont interconnectés entre eux à travers un réseau de communication filaire, généralement fiable et d'un débit élevé. Les liaisons sans fil ont une bande passante limitée qui réduit sévèrement le volume des informations échangées. Dans ce modèle, une unité mobile ne peut être, à un instant donné, directement connectée qu'à une seule station de base. Elle peut communiquer avec les autres sites à travers la station à laquelle elle est directement rattachée [4]

2.4 Les réseaux mobiles sans infrastructure (Ad hoc) :

Les réseaux sont composés d'un ensemble relativement des nouds mobiles qui se déplacent librement dans une zone géographique sans aucune infrastructure fixe préexistante. Tous les sites du réseau sont mobiles et communiquent d'une manière directe en utilisant leurs interfaces de communication sans fil. L'absence de l'infrastructure ou du réseau filaire composé des stations de bases (sans contrôle central), oblige les unités mobiles de se comporter comme des routeurs qui participent à la découverte et la maintenance des chemins pour les autres hôtes du réseau **[2].**

Comme le montre la figure suivante :

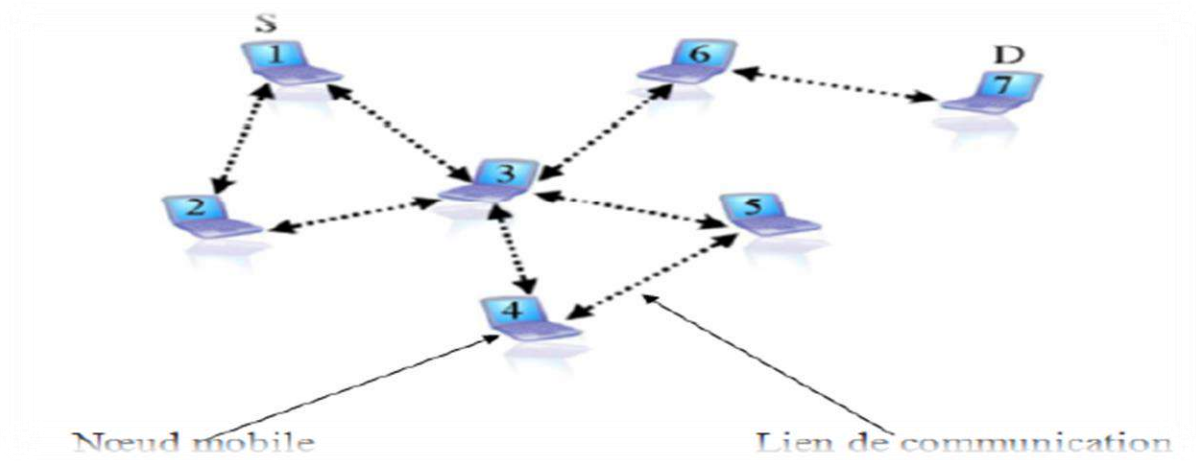

Figure 3-Les réseaux mobiles avec infrastructure[2]

La topologie du réseau peut changer à tout moment, elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités soit très fréquente comme le montre la figure suivante **[2]** :

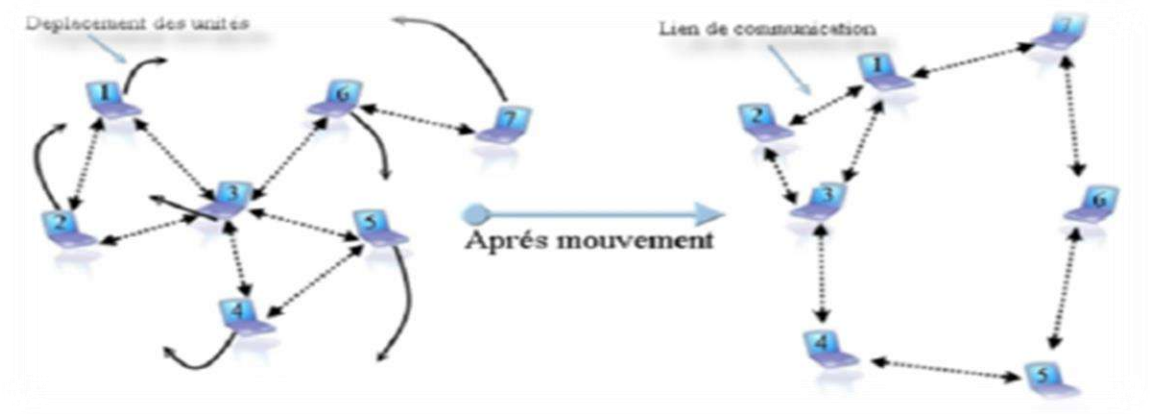

Figure 4-Le changement de la topologie des réseaux Ad hoc[2]

3- La communication cellulaire

La configuration standard d'un système de communication cellulaire est un maillage de cellules hexagonales : Initialement, une région peut être couverte uniquement par une seule cellule. Quand la compétition devient importante pour l'allocation des canaux, la cellule est généralement divisée en sept cellules plus petites.

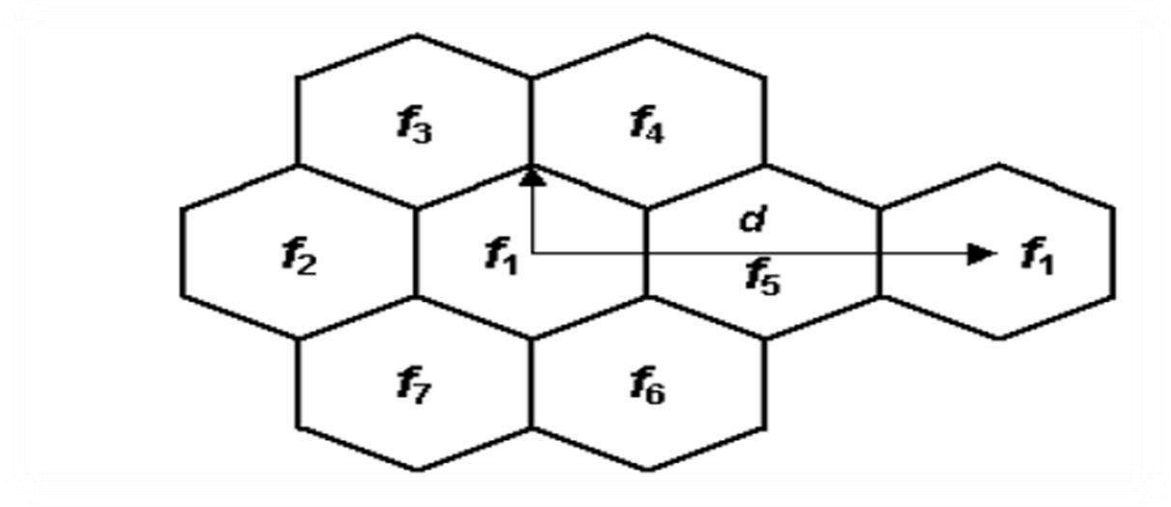

Figure 5-Le principe de réutilisation de fréquence[3]

Les cellules adjacentes dans le maillage doivent utiliser des fréquences différentes, contrairement à celles qui sont situées sur les côtés opposés du maillage et qui peuvent utiliser la même fréquence sans risque d'interférence **[3].**

4- L'utilisation des ondes radio dans la communication sans fil

La transmission radio utilisée dans la communication sans fil des unités mobiles est basée sur le principe que l'accélération d'un électron crée un champ électromagnétique qui à son tour accélère d'autres électrons et ainsi de suite. Il est alors possible de provoquer le déplacement électromagnétique. Plus le nombre d'électrons déplacés est important, plus le signal est fort et plus sera grande sa portée, avec une vitesse proche de celle de la lumière. Un déplacement coordonné d'électrons peut alors servir pour le transfert d'information et constitue la base de la communication sans fil. L'approche standard de la transmission radio est le déplacement des électrons à une fréquence donnée.

Des techniques de modulation et de multiplexage permettent d'adapter les signaux transmis à la bande passante du support de communication et de rentabiliser son utilisation. Deux signaux sur la même fréquence interfèrent et s'altèrent mutuellement. Pour y remédier le spectre de fréquence est divisé en plusieurs parties (bandes de fréquence), chaque partie étant dédiée à une utilisation spécifique. La taille limitée du spectre de fréquence impose donc le regroupement d'usagers dans des bandes étroites comme dans le cas de la radio cellulaire. Par exemple, la bande de 25 Mhz à 890 Mhz est réservée aux émissions de télévision et la bande supérieure à 890 Mhz pour la téléphonie cellulaire et la transmission par satellite. Au lieu d'allouer à chaque appel la totalité de la fréquence, la technologie cellulaire limite la puissance du signal au minimum nécessaire ; ce qui réduit les limites des interférences à une région de taille réduite autour de la station d'émission. Deux stations d'émission/réception situées dans des régions suffisamment éloignées l'une de l'autre peuvent utiliser la même fréquence sans risque d'interférence **[1].**

5-La fiabilité de la communication sans fil

La communication sans fil est moins fiable que la communication dans les réseaux filaires. La propagation du signal subit des perturbations (erreurs de transfert, microcoupure, timeout, etc.) dues à l'environnement, qui altèrent l'information transférée. Il s'ensuit alors un accroissement du délai de transit de messages à cause de l'augmentation du nombre de retransmissions. La connexion peut aussi être rompue ou altérée par la mobilité des sites. Un usager peut sortir de la zone de réception ou entrer dans une zone de haute interférence. Le nombre d'unités mobiles dans une

même cellule (dans le cas des réseaux cellulaires), par exemple lors d'un rassemblement populaire, peut entraîner une surcharge du réseau.

L'une des limites de la communication sans fil vient de la relative faiblesse de la bande passante des technologies utilisées. On distingue les réseaux utilisant l'infrarouge avec un débit de 1Mbps, la communication radio avec 2Mbps et le téléphone cellulaire avec 9 à 14 Kbps. La bande passante est évidemment partagée entre les utilisateurs d'une même cellule. Pour augmenter la capacité de service d'un réseau, deux techniques sont utilisées : la technique de recouvrement des cellules sur différentes longueurs d'ondes et celle qui réduit la portée du signal pour avoir plus de cellules mais de rayon moindre couvrant une région donnée **[1].**

6-Quelques éléments de l'infrastructure sans fil

Les réseaux informationnels de demain dits PCN (Personale Communication Network) intégreront une large variété de services (voix, données, multimédia, etc.). L'architecture générale de ces réseaux, bien qu'encore en débat, sera construite autour des infrastructures déjà existantes telles que :

- ✓ Les réseaux téléphoniques cellulaires (à l'avenir micro cellulaires) reliés au réseau téléphonique public.
- ✓ Les réseaux locaux traditionnels tels Ethernet, étendus à la communication sans fil, et reliés à des réseaux plus étendus de type LAN, WAN, Internet, etc.
- ✓ Les architectures orientées vers des services spécialisés fournis par diffusion sur des portions d'ondes radio en modulation de fréquence ou par des satellites à des usagers munis de terminaux spéciaux.

La même unité mobile peut, en principe, interagir avec les trois types d'infrastructures à différents moments, par exemple, en se déplaçant de l'intérieur d'un bâtiment où elle interagit avec un réseau local pourvu d'une interface de communication sans fil à l'extérieur du bâtiment où elle interagit avec le réseau téléphonique cellulaire **[1].**

7- Les réseaux mobiles ad hoc

7.1 Définition

MANET, acronyme de Mobile Ad hoc Networks, est le nom d'un groupe de travail de l'IETF, crée 1998, chargé de normaliser des protocoles de routage basés sur la technologie IP pour les réseaux ad hoc. Depuis la naissance de ce groupe de travail, le terme MANET est utilisé comme nom commun pour distinguer un réseau mobil ad hoc. Un réseau MANET est formé de manière dynamique par un système autonome de nouds mobiles connectés via des liens sans _l sans nécessiter d'infrastructures externe ni d'administration centralisée.

La topologie du réseau peut changer rapidement et de façon imprévisible. Un tel réseau peut opérer dans un monde autonome et il y'a aucune limitation de taille dans un réseau ad hoc, il peut contenir des dizaines ou des milliers de nouds. Les MANET sont utiles dans beaucoup d'applications car ils n'ont pas besoin de tout support d'infrastructure **[2].**

7.1.1 Un exemple d'un réseau ad hoc

Un groupe d'unités portables reliées par des cartes HIPERLAN. Les réseaux appelés GSM ne représentent pas des réseaux ad hoc, car la communication entre les unités passe obligatoirement par des stations de base du réseau filaire **[3].**

7.1.2 Modélisation

Un réseau ad hoc peut être modéliser par un graphe $G_t = (V_t,E_t)$. Où :

- V_t représente l'ensemble des nœuds (i.e. les unités ou les hôtes mobiles) du réseau et
- E_t modélise l'ensemble les connections qui existent entre ces nœuds.

Si $e = (u.v) \in E_t$, cela veut dire que les nœuds u et v sont en mesure de communiquer directement à l'instant t. La figure suivante représente un réseau ad hoc de 10 unités mobiles sous forme d'un graphe **[3]** :

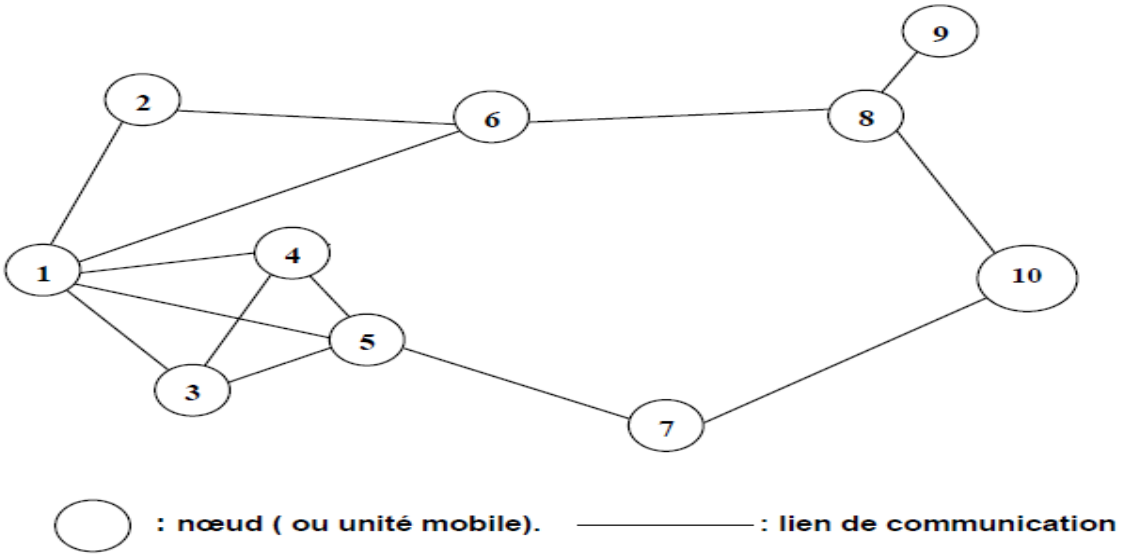

Figure 6-La modélisation d'un réseau ad hoc[2]

La topologie du réseau peut changer à tout moment, elle est donc dynamique et imprévisible ce qui fait que la déconnexion des unités soit très fréquente.

7.1.3 Les modes de communication dans les réseaux mobiles ad hoc

La communication dans les réseaux mobiles Ad Hoc utilise plusieurs modes dont : la communication

" point à point " ou " Unicast ", la communication " multipoint " ou " Multicast ", et la diffusion " Broadcasté "**[2].** Ces trois modes de communication peuvent être schématisés par la figure suivante :

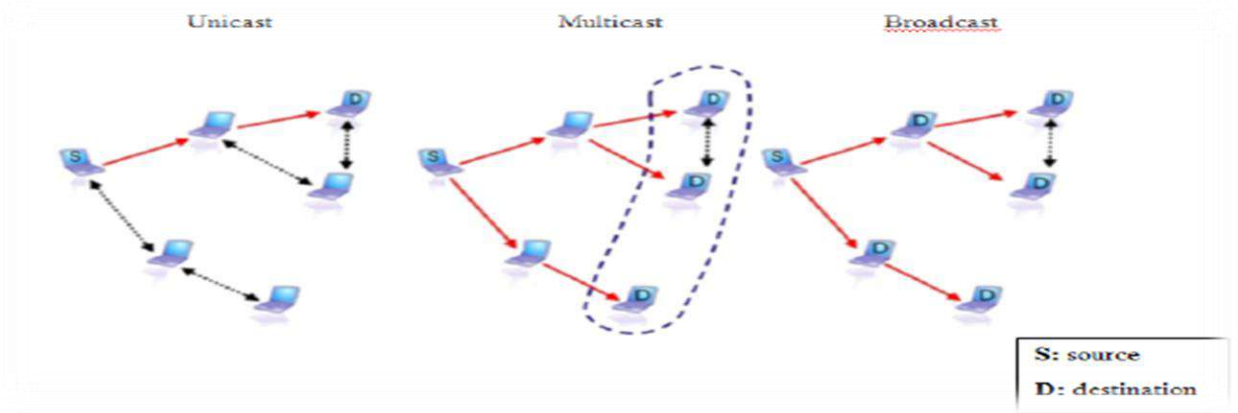

Figure 7-Les différents modes de communication [2]

7.2 Les caractéristiques des réseaux Ad hoc

Les réseaux sans fil ad hoc se caractérisent principalement par **[2]** :

- Bande passante limitée : Une des caractéristiques primordiales des réseaux basés sur la communication sans fil est l'utilisation d'un médium de communication partagé (ondes radio). Ce partage fait que la bande passante réservée à un hôte soit modeste.

- Contraintes d'énergie : Les hôtes mobiles sont alimentés par des sources d'énergie autonomes comme les batteries ou les autres sources consommables. Le paramètre d'énergie doit être pris en considération dans tout contrôle fait par le système.

- Sécurité physique limitée : Les réseaux mobiles Ad Hoc sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justice par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.

- Erreur de transmission : Les erreurs de transmission radio sont plus fréquentes dans les réseaux ad hoc que dans les réseaux filaires.

- Topologie dynamique : Les unités mobiles du réseau se déplacent d'une façon libre et arbitraire. Par conséquent, la topologie du réseau peut changer à des instants imprévisibles, d'une manière rapide et aléatoire

.

- Interférences : Les liens radios ne sont pas isolés, deux transmissions simultanées sur une même fréquence ou, utilisant des fréquences proches peuvent interférer.

- Absence d'infrastructure : Les réseaux ad hoc se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructures préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d’une manière continue.

7.3 Les applications des réseaux mobiles Ad Hoc

Les réseaux mobiles ad hoc sont rapides et faciles à déployer, ils sont particulièrement intéressants pour les applications militaires ou l'installation d'infrastructure fixe **[2].** Ils peuvent être aussi utilisées dans :

- Les opérations de recherches et de recours : En cas de tremblement de terre, de feux ou d'inondation, dans le but de remplacer rapidement l'infrastructure détruite.

- Les gares et aéroports : Pour la communication et la collaboration entre les membres du personnel.

- Les applications industrielles : Des scénarios plus complexes dans le domaine industriel peuvent former un MANET, pour s'adapter aux différents environnements. Un exemple d'une telle application est la formation d'un MANET pour la surveillance médicale.

- Applications commerciales : Pour un paiement électronique distant ou pour l'accès mobile à l'Internet, ou service de guide en fonction de la position de l'utilisateur.

- Home Network : Partage d'applications et de communications des équipements mobiles.

7.4 Les avantages et Les inconvénients des réseaux ad hoc

7.4.1 Les avantages:

Le mode de fonctionnement des réseaux ad hoc présente de nombreux avantages **[2]** :

- L'avantage principal d'un tel réseau et son déploiement rapide (en effet, Il est possible d'utiliser un réseau dans des situations d'urgence, par exemple, pour organiser les secours lors d'une catastrophe naturelle), sa robustesse, sa flexibilité et le support de la mobilité.
- La possibilité de créer un réseau local mobile : en effet, Un ensemble de personnes se déplaçant peuvent créer un réseau local mobile. On peut, par exemple, imaginer l'utilisation d'un réseau ad hoc pour relier entre eux au cours d'un trajet, les camions d'un convoi routier.
- Facilité d'installations temporaires : telles que les stands de foire, les expositions ou salles de conférence.
- Installation plus économique du réseau dans les endroits difficiles à câbler.

- Adapté aux environnements dynamiques nécessitant des transformations fréquentes grâce au cout minimum du câblage.

7.4.2 Les inconvénients

Le mode de fonctionnement des réseaux ad hoc présente de nombreux inconvénients **[2]** :

- Bruits et interférences : Accroissent le nombre d'erreurs sur la transmission et amoindrissent d'autant les performances d'un lien radio.
- Perte de paquets.
- Débits plus faibles que filaire, les nouds sont souvent peu puissants, problème de consommation d'énergie.
- Faible en termes de sécurité.
- Liens asymétriques : ces liens n'accueillent que des communications à sens unique entre un Émetteur et un récepteur.
- Mobilité des nouds : modification de la topologie du réseau, et la transformation du tracé des routes lors des échanges des paquets.

8- Les protocoles de routage dans MANET

Plusieurs protocoles de routage ont été proposés pour le contexte ad hoc, qu'ils s'agissent d'adaptation des protocoles de routage Internet au contexte ad hoc ou de nouveaux protocoles spécifiques aux réseaux MANET.

Le principal but de toutes stratégies de routage est de mettre en œuvre une bonne gestion d'acheminement qui soit robuste et efficace. D'une manière générale, toutes stratégies de routage reposent sur des méthodes et des mécanismes que l'on peut regrouper en trois grandes classes : les protocoles de routage proactifs, les protocoles de routage réactifs **[3].**

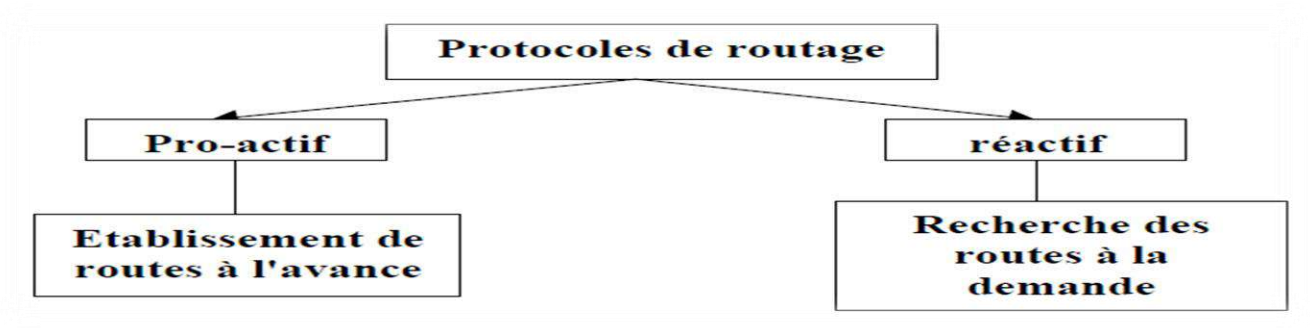

Figure 8-Classification des protocoles[3]**.**

8.1 les protocoles réactifs

Les protocoles de routage réactifs (ou protocoles de routage à la demande) représentent les protocoles les plus récents proposés dans le but d'assurer le service du routage dans les réseaux sans fil ad hoc. La majorité des solutions proposées pour résoudre le problème de routage dans les réseaux ad hoc, et qui sont évaluées par le groupe de travail MANET de l'IETF, appartiennent à cette classe de protocoles de routage.

Les protocoles de routage appartenant à cette catégorie, créent et maintiennent les routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte globale de routes est lancée, et cela dans le but d'obtenir une information. Actuellement, les plus connus de ces protocoles est AODV (Ad hoc On Demande Distance Vector) et DSR (Dynamic Source Routing Protocol).

A l'opposé des protocoles proactifs, dans le cas d'un protocole réactif, aucun message de contrôle ne charge le réseau par des routes inutilisées, ce qui permet de ne pas gaspiller les ressources du réseau. Mais la mise en place d'une route par inondation peut être coûteuse et provoquer des délais importants avant l'ouverture de la route et les retards dépassent bien souvent les délais moyens admis par les logiciels, aboutissant ainsi à une impossibilité de se connecter alors que le destinataire est bien là **[2].**

8.2 Les protocoles pro-actifs

Les protocoles de routage proactifs essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles (qui peuvent représenter l'ensemble de tous les nouds du réseau) au niveau de chaque nœud noud du réseau. Les routes sont sauvegardées même si elles ne sont pas utilisées. La sauvegarde permanente des chemins de routage, est assurée par un échange continu des messages et de mise à jour des chemins. Les plus aboutis de ces protocoles est OLSR (Optimized Link State Routing Protocol) et DSDV (Destination Sequenced Distance Vector).

Avec un protocole proactif, les routes sont disponibles immédiatement ; ainsi, l'avantage d'un tel protocole est le gain de temps lors d'une demande de route. Mais, les changements de routes

peuvent être plus fréquents que la demande de route, et le trafic induit par les messages de contrôle et de mise à jour des tables de routage peuvent être important et partiellement inutile, ce qui gaspille la capacité du réseau sans fil. De plus, la taille des tables de routage croit linéairement en fonction du nombre de nœud **[2]**

8.3 Les Avantages et les Inconvénients des protocoles de routage dans MANET [2]

8.3.1 Les Avantages

Parmi les avantages des réseaux ad hoc, on site :

- Proactifs : Une route est toujours disponible entre une source et une destination, rapide ce qui implique le gain du temps.
- Réactifs : La génération de trafic de contrôle ne se fait que lorsqu'il est nécessaire.

8.3.2 Inconvénients

Les inconvénients des protocoles de routages dans MANET sont :

- Proactifs : Le coût, dû au maintien de la topologie, génère une consommation continuelle de la bande passante.
- Réactifs : L'inondation est un mécanisme très coûteux, génération de délai important pour ouvrir une route entre deux nouds.

9. Conclusion

Ce chapitre a été axé sur le concept des environnements mobiles et l'utilisation de la technologie de communication sans fil. L'évolution rapide qu'a connu la technologie sans fil récemment, a permet l'apparition de nouveaux systèmes de communication qui offrent plus d'avantages par rapport aux systèmes classiques. Les nouveaux systèmes n'astreignent plus l'usager à une localisation fixe, mais lui permet une libre mobilité.

La compréhension parfaite de la communication utilisée dans le nouvel environnement, nécessite la compréhension des notions de base de la technologie sans fil comme l'utilisation des ondes radio, la notion de bande passante, la réutilisation des fréquences, la portée d'une unité mobile … etc. Le

but de ce chapitre a été de donner un aperçu général sur cette technologie qui ne cesse pas de croître.

Les environnements mobiles sont caractérisés par de fréquentes déconnexions et des restrictions sur les ressources utilisées, surtout si tous les usagers du système sont mobiles ce qui est le cas pour les réseaux ad hoc. Ces limitations transforment certains problèmes, ayant des solutions évidentes dans l'environnement classique, en des problèmes complexes et difficiles à résoudre. Parmi ces problèmes figure le problème de routage.

Chapitre II :

Présentation des deux simulateurs « NS-2 » et « OMNET++ »

Chapitre II: Présentation des deux simulateurs« NS-2 » et « OMNET++»

I. Introduction

Pour tester les performances d'une solution apportée à un problème de communication dans un réseau, il n'est pas toujours possible d'accéder aux infrastructures nécessaires en raison de leurs coûts élevés. De plus, les expérimentations réelles n'offre souvent pas une grande souplesse. Rappelons que les réseaux ad hoc sont des réseaux qui englobent plusieurs unités mobiles qui se déplacent dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces radio. En effet, il serait très coûteux voire impossible de mettre en place un réseau à des fins de tests de certains critères. Pour remédier à ce problème et afin de tester les performances d'un nouveau protocole, on a recours à la simulation qui met à la disposition de l'utilisateur un environnement d'expérimentation.

II. Environnement de Simulation

1. Introduction à la simulation

La simulation connaît de nos jours un essor considérable. Ceci est dû aussi bien à l'intérêt théorique que présente la modélisation des systèmes simulés, que par les besoins croissants de simuler par ordinateur des réalisations de plus en plus complexe.

On constate en effet que les conditions d'expérimentation sont aujourd'hui toujours plus coûteuses, notamment aux techniques dites virtuelles. Celle-ci consiste à représenter la réalité simulée sur ordinateur, à donner à celle-ci des moyens de perceptions réalisés artificiellement, puis à simuler son fonctionnement. C'est sur ce dernier point que la simulation joue un rôle important en mettant à la disposition de l'utilisateur un environnement d'expérimentation dont on peut faire varier les paramètres.

Les conceptions et les techniques de programmation on fait de leur côté d'important progrès, on particulier dans les domaines de l'abstraction et du parallélisme. C'est ainsi que les langages de programmations d'aujourd'hui offrent des moyens performants, ce qui rend possible la réalisation d'un simulateur dans un environnement de programmation existant **[17].**

2. Système réel et objectif de simulation

La simulation est une technique de modélisation du monde réel. Elle permet de représenter le fonctionnement d'un système composé de différent centre d'activité, de mettre en évidence les caractéristiques de ceci et les interactions entre eux, de décrire la circulation de différents objets

traités par ces processus et en fin d'observer le comportement du système dans son ensemble et dans son évolution dans le temps.

On appelle communément système réel le phénomène que l'on veut observer. On y distingue d'une part les centres d'activités ou taches d'activités, c-à-d les endroits où il se passe quelque chose ou un objet est créé, traité, transformé ou détruit, d'autre part les objets qui circulent dans le système et qui sont manipulés par différentes taches actives.

Nous considérons que le but de l'opération est d'étudier un tel système et pour comprendre les relations entre ces composantes ou pour prédire ces performances dans un environnement donné **[17].**

La simulation par événement discret désigne la modélisation d'un système réel tel qu'il évolue dans le temps, par une représentation dans laquelle les grandeurs caractérisant les systèmes (variables) ne changent qu'en un nombre fini de points isolés dans le temps. Ces points sont les instants où se passent les événements, c'est-à-dire le phénomène capable de modifier l'état du système et nous appelons événement tout changement d'état d'un système réel se produisant à un instant donné, ainsi que les actions qui accompagnent ou caractérisent ce changement.
La simulation par événement discret consiste alors à prendre en compte, dans la modélisation des tâches actives, les seuls instants où un événement se produit et à concentrer l'activité des tâches à simuler sur ces instants là **[Site5]**

3. Modèles de simulation

On distingue plusieurs modèles de simulation selon qui sont statiques ou dynamiques, déterministes ou aléatoires, continues ou discrets.

- Un modèle de simulation est dynamique ou statique selon que les valeurs de ses variables
- Un modèle de simulation est déterministe s'il ne contient aucune variable aléatoire, il est aléatoire dans le cas contraire.

Dans le cas d'un système aléatoire, les résultats de la simulation sont eux même aléatoires et ne donnent qu'une estimation du comportement du système simulé.

Enfin, un modèle de simulation est continu si l'ensemble des instants considéré forme un intervalle compact sur l'axe des temps, il est discret si celui-ci se compose d'un nombre fini ou dénombrable de valeurs isolées.

3.1 Gestion du temps

Une simulation consiste à gérer le temps ainsi que les actions qui sont liés aux différents instants du système réel et à faire fonctionner abstraitement le modèle qui représente ce système.

Dans un modèle discret, le temps est géré par un échéancier et par une horloge centrale ou une horloge du système. L'échéancier est une liste d'événement ordonné chronologiquement selon l'heure à laquelle ils doivent être activés. L'horloge du système donne à chaque instant l'heure du système de simulation.

Elle est réglée au départ sur l'heure d'activation du premier événement. Chaque événement inséré dans l'échéancier en est retiré lorsque son heure d'activation est égale à celle du système de simulation. Des actions associées à cet événement sont alors exécutées, ensuite l'horloge du système est avancée à l'heure d'activation de l'événement suivant.

Dans un modèle continu le temps est discrétisé selon un pas donné et, chaque avance du temps, les valeurs des variables du système sont mises à jour **[17].**

3.2 Simulation par événements discrets

La simulation par événements discrets désigne la modélisation d'un système réel tel qu'il évolue dans le temps, par une représentation dans laquelle les grandeurs caractérisant le système (variable) ne changent qu'en un nombre fini de ou dénombrable de points isolés dans le temps. Ces points sont les instants ou se passent les événements, c-à-d le phénomène capable de modifier l'état de système et nous appelons comme événement tout changement d'état du système réel se produisant à un instant donné, ainsi que les actions qui accompagnent ou caractérisent ce changement.

La simulation par événement discret consiste alors à prendre en compte dans la modélisation des taches actives, les seuls instants ou un événement se produit et à concentrer l'activité des taches simulées sur ces instants-là. Nous considérons comme système de simulation par événements discrets tous les systèmes basés sur cette abstraction **[17].**

3.3 Simulateur

Nous appelons simulateur un programme qui met en œuvre un modèle de simulation par événements discrets. La tache première d'un simulateur est d'assurer que la chronologie des

événements soit respectée. A chaque occurrence d'un événement, les actions qui sont associées à celui-ci sont exécutées **[17].**

4. Avantages et inconvénients de la simulation

Nous trouvons aussi dans la simulation les avantages et les inconvénients **[4]** :

- **Avantage :**
  - ✓ Observations des états du système.
  - ✓ Etudes des points de fonctionnement d'un système.
  - ✓ Etudes de l'impact des variables sur les performances du système.
  - ✓ Etude d'un système sans les contraintes matérielle

- **Inconvénients :**
  - ✓ La conception de modèles peut nécessiter des compétences spéciales.
  - ✓ Résultats pas forcément généralisable.

II NS-2 (Simulator Network 2)

1- Présentation de network simulator NS-2

Le simulateur du réseau NS-2 est un outil logiciel de simulation de réseaux informatiques. Il est principalement bâti avec les idées de la conception par objets, de réutilisation du code et de modularité.

NS-2 est écrit en C++ et utilise le langage OTCL (Object Tools Command Langage) dérivé de TCL. A travers OTCL, l'utilisateur décrit les conditions de la simulation : la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés, les communications qui ont lieu. La simulation doit d'abord être saisie sous forme de fichier que NS va utiliser pour produire un fichier contenant les résultats. Mais l'utilisation de l'Otcl permet aussi à l'utilisateur de créer ses propres procédures (par exemple s'il souhaite enregistrer dans un fichier l'évolution d'une variable caractéristique du réseau au cours du temps). Il contient les fonctionnalités nécessaires à l'étude des algorithmes de routage unicast ou multicast, des protocoles de transport, de réservation, des services intégrés, des protocoles d'application. De plus le simulateur possède déjà une palette de systèmes de transmission, d'ordonnanceurs et de politiques de gestion de files d'attente pour effectuer des études de contrôle de congestion **[5]**

Le Simulateur se compose d'une interface de programmation en TCL et d'un noyau écrit en C++ dans lequel la plupart des protocoles réseaux ont été implémentés :

- Traffic parreto, ON/OFF, CBR, FTP, telnet, etc.
- Couche Transport TCP, UDP.
- Couche Réseaux IP, routage dans les réseaux ad-hoc (aodv, dsr,dsdv, tora, amodv), routage dans les réseaux filaire (Link state, Distance vector), les réseaux multicast, IntServ, DiffServ .
- Couche MAC CSMA, CDMA, 802,X, Token ring, MPLS, liens satellite, etc **[site01]**

2-Le langage TCL (Tool Command Language)

TCL (Tool Command Language) Ce langage est à typage dynamique, multi-plateforme et extensible. Il est facile à apprendre **[6].** TCL s'interface trés facilement avec le langage C, de plus il est économe en mémoire vive. est un langage de programmation dont le but est de passer des commandes à des programmes interactifs tels que des éditeurs de texte, des débogueurs et des interpréteurs shell. Il possède une syntaxe simple et il est lui-même programmable : les utilisateurs de Tcl peuvent en effet écrire des procédures pour créer des commandes plus puissantes que celles fournies par l'ensemble préconstruit. La bibliothèque Tcl est constituée d'un analyseur syntaxique du langage Tcl, de routines implémentant les commandes prédéfinies de Tcl, et de procédures permettant à chaque application d'ajouter à Tcl des commandes additionnelles qui lui sont spécifiques. Le programme applicatif génère des commandes Tcl et les passe à l'analyseur syntaxique de Tcl pour l'exécution**[site3]**

2.1 Otcl

OTcl est une extension orientée objet de Tcl. Les commandes Tcl sont appelées pour un objet.OTcl est une extension du langage de commande TCL, qui utilise une programmation structuré (boucles, procédures, notions de variables)**[6].**

Le moteur OTcl transforme les instructions TCL en instructions C++. En OTcl, les classes sont également des objets avec des possibilités d'héritage. La définition d'une classe commence par la directive Class. Les fonctions et les attributs d'une classe s'installent par la suite par les commandes « instvar » et « insproc ». L'utilisation « instproc » définit les méthodes de la classe de manière assez semblable à C++. Lors d'une instanciation en OTcl **[7].**

3-NAM

NAM est un outil de visualisation qui présente deux intérêts principaux : représenter la topologie d'un réseau décrit avec NS-2, et afficher temporellement les résultats d'une trace d'exécution NS-2. Par exemple, il est capable de représenter des paquets TCP ou UDP, la rupture d'un lien entre nœuds, ou encore de représenter les paquets rejetés d'une file d'attente pleine. Ce logiciel est souvent appelé directement depuis les scripts TCL pour NS-2, afin de visualiser directement le résultat de la simulation **[8].**

4-Le modèle de réseau sous NS-2

Un modèle de réseau sous NS est constitué :

- De nœuds de réseau : endroits où est généré le trafic, ou nœuds de routage ;
- De liens de communication entre les réseaux
- D'agents de communication, représentant les protocoles de niveau transport (TCP, UDP) ; ces agents sont attachés aux nœuds et connectés l'un à l'autre, ce qui représente un échange de données (connexion TCP, flux UDP).
- D'applications qui génèrent le trafic de données selon certaines lois (CBR, VBR), et se servent des agents de transport **[5]**.

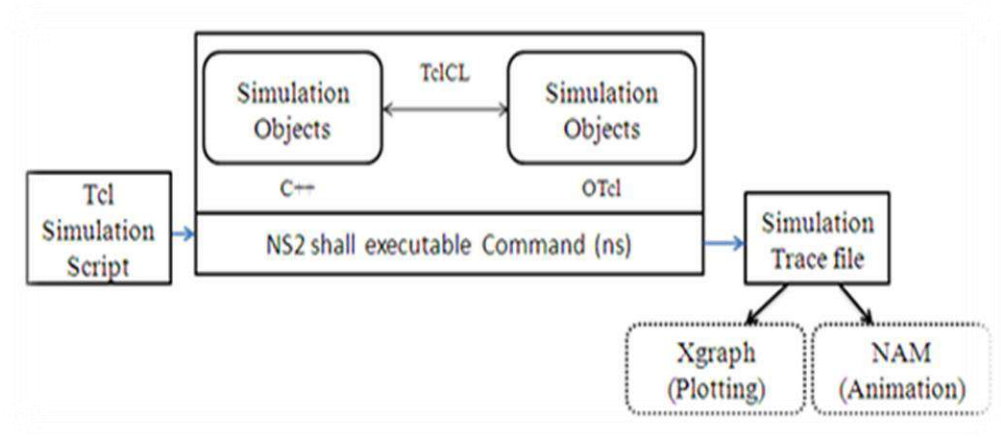

Figure 9-L'architecture simulation du NS2

L'architecture générale du NS2 est présentée sur la (figure 9), il consiste en deux types de Langage de programmation : le C++ et l'OTcl (Object-orientedTool Command Language). Le C++ est utilisé pour programmer les entités internes des systèmes simules, alors que l'OTcl est utilisé pour définir les scenarios des simulations et les paramètres de configuration. Ces deux types de langages sont, ensuite, lies via le TclCL qui permet le passage des codes C++ vers les codes en OTcl et « viceversa » **[9]**

Une fois la simulation terminée, NS produit deux fichiers de traces qui visualisent la dynamique des systèmes simules, et qui peuvent être interpréter en utilisant les outils : NAM (Network AniMator) et Xgraph.**[9]**

5-Caractéristiques d'une entité communicante sous NS-2

Le nœud (entité communicante) constitue l'élément de base de notre modèle. Un nœud dans NS-2 est une classe définie dans OTCL, qui a une adresse et qui contient trois entités : le « classifier », le lien et l'agent **[site2]**

5.1 - Le Classifier

La fonction d'un nœud est d'examiner des champs du paquet reçu, et plus précisément, l'adresse source et l'adresse destination. Selon ces valeurs, le nœud envoie ce paquet sur ses interfaces de sortie (F). En NS-2, ceci est effectué par un objet qui s'appelle « classifier ». Il existe plusieurs types de classifier qui sont utilisés pour différents buts :

- « adresse classifier » : il est utilisé pour traiter les paquets unicast, son rôle est de sélectionner les paquets adressés directement au nœud, et de choisir le lien vers le prochain nœud.
- « port classifier » : son rôle est de sélectionner l'agent auquel le paquet est destiné.
- « multicast classifier » : il est utilisé pour classifier les paquets multicast.

5.2- Le lien

Il est utilisé pour relier les nœuds ***(voir* Figure 10 *)***. Un lien est défini par plusieurs paramètres comme : sa bande passante, le point d'entrée, la durée de vie de chaque paquet, etc. NS2 présente plusieurs types de liens, ainsi on peut distinguer des liens unidirectionnels ou bidirectionnels, des liens filaires et des liens non filaires pour modéliser les réseaux sans fils.

5.3 - L'agent

Les agents représentent des points terminaux, là où des paquets de couche réseau sont construits ou consommés. Ces agents constituent le troisième composant du nœud. Dans NS2, le rôle de l'agent est de fournir l'adresse de destination, les fonctions pour générer les paquets et l'interface à la classe application (voir **Figure 10***)*.

Dans NS2 il existe plusieurs types d'agents, chacun a un rôle spécifique :

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

La figure 10 représente les entités existantes dans un nœud et les liens entre ses entités.
[10]

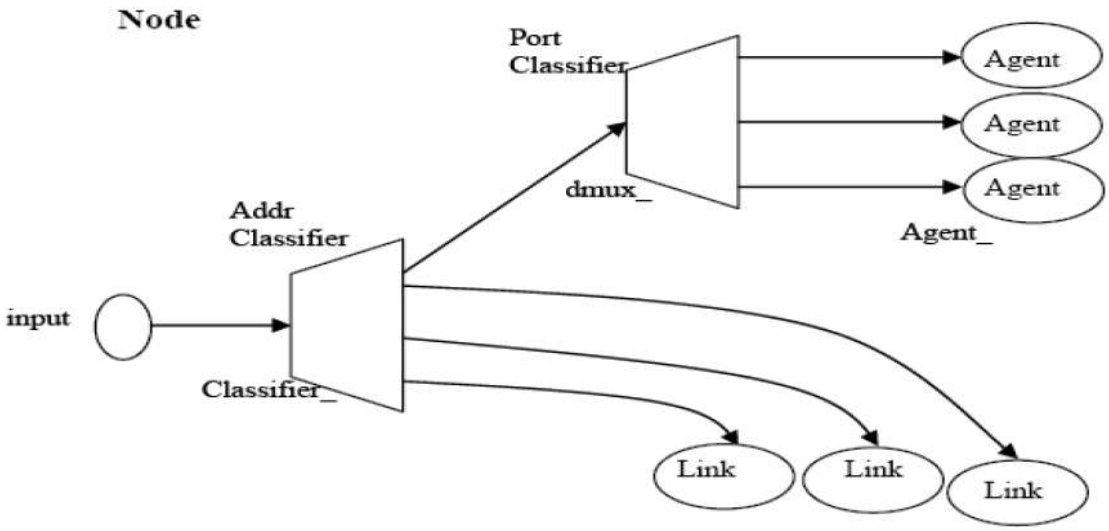

Figure 10-Schéma d'un nœud dans NS2

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 **[10]**

6-Le processus de simulation

Le processus de simulation en utilisant NS-2 est composé de trois phases principales **[10]** :

- **Phase de préparation** : s'occupe de la génération des fichiers d'entrées. A cette étape, on introduit des fichiers de scripts Otcl qui décrivent l'environnement avec tous ses nœuds, leurs déplacements et leur trafic de données. Ces fichiers sont classés en deux catégories :
  - Fichiers de scénario qui décrivent les nœuds, leurs positions ainsi que leurs mouvements.
  - Fichiers de communication qui décrivent le trafic dans le réseau.

- **Phase de simulation** : pour lancer les simulations et générer les traces. Les deux fichiers obtenus de la phase de préparation sont introduits dans un script de lancement OTcl. Le script de simulation consiste à indiquer la topologie du réseau, à activer des traces aux endroits pertinents, à engendrer des événements particuliers à des instants donnés. A la fin de cette étape on obtient deux fichiers (journaux) appelé aussi «fichiers traces ». Le premier fichier sera traité par l'outil de visualisation NAM. Et le deuxième doit être filtré par un script awk afin d'afficher le résultat en utilisant l'outil Xgraph.
- **Phase d'analyse** : pour analyser les traces et générer les courbes. L'outil de visualisation NAM s'occupe du premier fichier trace. Deux éléments intéressants sont proposés à la visualisation : un dessin de la topologie du réseau étudié, et une visualisation dynamique du déroulement du programme dans le temps. Le deuxième fichier de trace sauvegarde tous les échanges de paquets effectués. Afin de dessiner les courbes en utilisant Xgraph, le fichier doit être filtré par un script awk pour ne garder que les informations pertinentes.

Le schéma des étapes de simulation est illustré dans la figure suivante **[10]** :

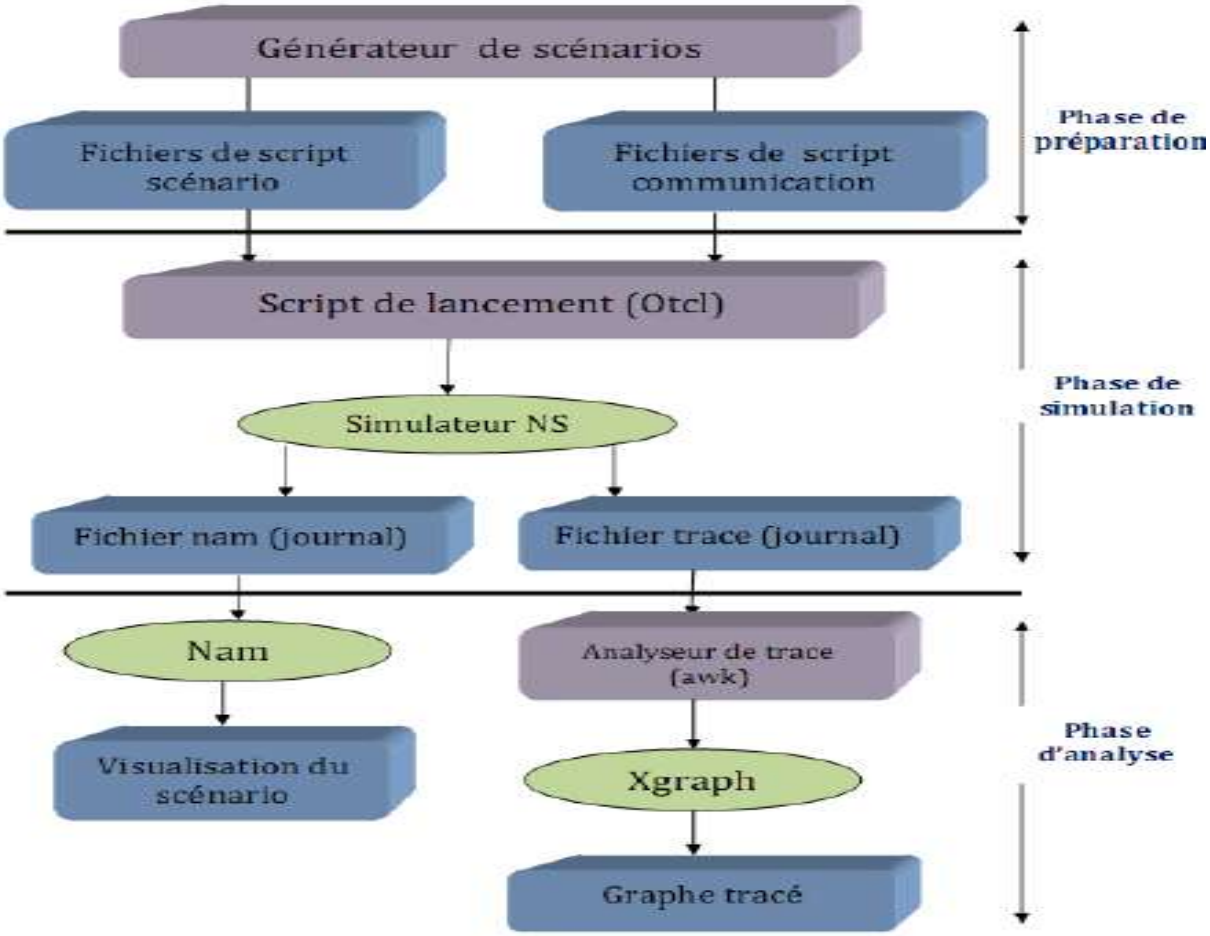

Figure 11-Le processus de simulation NS-2

7-Création d'un scénario

Pour décrire un réseau et son trafic, il faut définir dans l'ordre [11] :

1. la topologie du réseau : les nœuds et les arcs
2. la couche transport (UDP, TCP, ...) entre des pairs de nœuds
3. la couche application qui va fournir les données
4. des temporisateurs précisant les instants auxquels les transferts vont démarrer

- Dans un fichier otcl, on commence par créer un simulateur

```
set ns              [new Simulator]
```

- Pour conserver l'historique de la simulation (notamment pour la visualiser avec NAM), on peut indiquer les deux fichiers vers lesquels seront dirigés les deux historiques :

```
set tracefd       [open simulation_tras.tr w]
set namtrace      [open simulation_tras.nam w]
```

- Topologie : Ensuite, on crée les nœuds du réseau, puis les liens qui les relient avec leur débit et leur latence, et la politique de rejet des paquets en trop (ici DropTailmais il y en a d'autres). On peut aussi préciser le nombre limite de paquets dans la file d'attente.

```
set n1 [$ns node]
set n2 [$ns node]
$ns duplex-link $n1 $n2 1Mb 50ms DropTail
$ns queue-limit $n1 $n2 100
```

- Les nœuds de réglage de position et de mouvement

```
# Provide initial location of mobilenodes
$node_(0) set X_ 50.0
$node_(0) set Y_ 200.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 100.0
$node_(1) set Y_ 200.0
$node_(1) set Z_ 0.0

# Generation of movements node_(1) . vitess Node(1) 1m/s
$ns at 1 "$node_(1) setdest 400.0 200.0 1"
```

Au moment de 1,0 secondes, le nœud commence à se déplacer à partir de son emplacement actuel vers Noeud1 (50, 200, 0.0) à la vitesse de 1m / s. et Nœud 2 (100, 200, 0.0) fixé.

- Transport : Il faut créer les agents de transport, les attacher aux nœuds et les connecter entre eux. Parmi les agents disponibles, on trouve Agent/Null, Agent/UDP, Agent/TCP et Agent/TCPSink. Le code est par example:

```
set udp [new Agent/UDP] # crée une source UDP
set trappe [new Agent/Null] # crée un puits d'information
$ns attach-agent $n1 $udp # attache la source à un noeud
$ns attach-agent $n2 $trappe # attache la destination à un autre
$ns connect $udp $trappe # associe la paire origine/destination
```

- Il faut enfin attacher l'application à l'agent de transport :

```
set app [new Application/Traffic/CBR]
$app attach-agent $udp
```

- Pour démarrer et arrêter une source src, on a :

```
$src start et $src stop
```

- Temporisateurs : Il est possible de programmer des évènements qui seront exécutés à des dates précises de la simulation. Par exemple:

```
$ns at 1.0 "$truc start" # démarre l'application $truc à t=1.0
$ns at 5.0 "$truc stop" # l'arrête à t=5.0
$ns at 10.0 "finish" # invoque la procédure finish à t=10.0
```

- Typiquement, la fin de la simulation se fera par l'invocation d'une procédure chargée du traitement final des statistiques. Exemple :

```
proc finish { } {
global ns f nf stat
puts "C'est fini"
close $f # fermeture du fichier de trace (s'il y a)
close $nf # fermeture du fichier de trace animation
puts "Débit = $stat(debit)" # impression d'une statistique
```

8-L'avantage et inconvénients du simulateur NS-2

8.1 Les Avantages du simulateur NS-2 [10] :

- Un logiciel de simulation multicouche.
- Un outil complètement libre pour plusieurs plateformes.
- Possibilité d'ajouter des composants à la demande.
- Développement orienté objet.

- Du fait de sa popularité, de nombreux protocoles sont à priori disponibles pour NS2.

8.2 Les inconvénients :

- la modélisation dans NS-2 reste une tâche complexe : il n'y a pas d'interface graphique .
- Une forte technicité est requise pour utiliser ce simulateur.

III. Le simulateur OMNET++.

1-Présentation OMNET++

OMNET++(Objective Modular Network Testbed in C++) est un environnement de simulation à événements discrets. Il est basé sur une architecture orientée composant, dont les modules sont écrits en C++. OMNET a été principalement conçu pour simuler la communication dans les réseaux mais grâce à son architecture flexible et générique, il a ensuite été utilisé pour beaucoup d'autres applications. Malgré que OMNET n'est pas un simulateur de réseau proprement dit, il est en train de gagner une large popularité comme étant une plateforme de simulation de réseaux auprès de la communauté scientifique ainsi que du domaine industriel [13]

L'avantage de OMNET ++ est sa facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés.

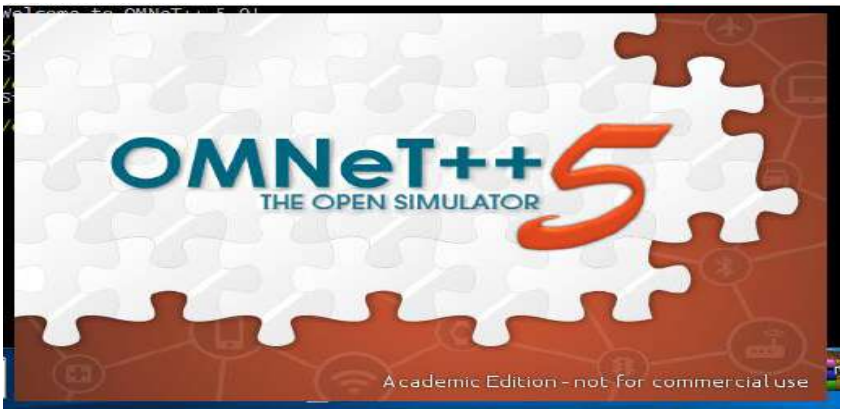

Figure 12-Le lancement du simulateur OMNET++

2.Architecture d'OMNET++

Les modèles OMNET++ constituent en un ensemble de modules hiérarchiquement emboités tel qu'il est montré dans la (Figure13)**:**

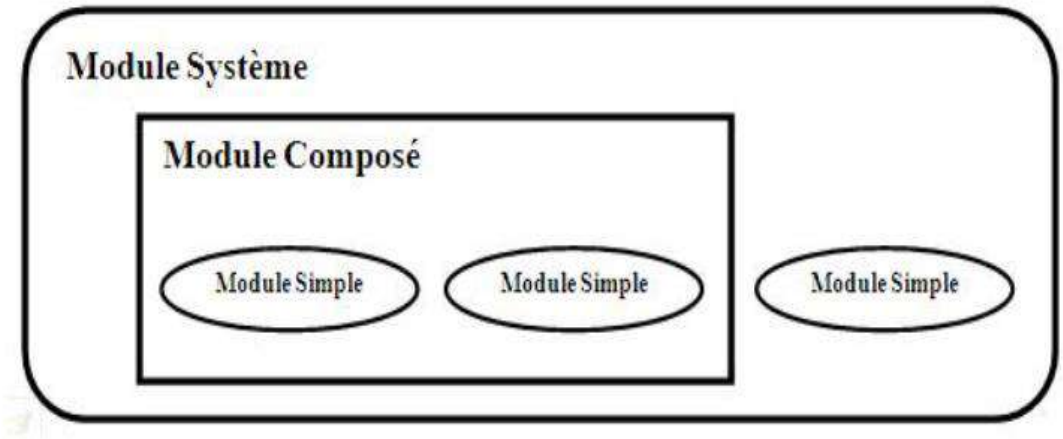

Figure 13-Architecture modulaire du simulateur OMNET++

L'architecture d'OMNET++ est hiérarchique composé de modules. Un module peut être soit module simple ou bien un module composé. Les feuilles de cette architecture sont les modules simples qui représentent les classes C++. Pour chaque module simple correspond un fichier .cc et un fichier.h. Un module composé est composé de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports de chaque module sont spécifiés dans un fichier.ned.

La communication entre les différents modules se fait à travers les échanges de messages. Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service. Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module. La conception d'un réseau se fait dans un fichier .ned et les différents paramètres de chaque module sont spécifiés dans un fichier de configuration (.ini). OMNET++ génère à la fin de chaque simulation deux nouveaux fichiers omnet.vec et omnet.sca qui permettent de tracer les courbes et calculer des statistiques **[14]**

- La modélisation des protocoles de communications
- La modélisation des réseaux filaires et sans fils

- La modélisation des systèmes répartis
- L' architecture HardWare

3.Les principaux fichiers d'OMNET++

Les différents fichiers sont **[13]** :

3.1Fichier(.NED):

Utilise le langage NED(NEtwork Descriptor) de description de réseaux. Il peut être utilisé en deux modes : Mode Graphique ou Mode Texte qui permettent de décrire les paramètres et les ports du module. Les erreurs cosmisent sont indiquées en temps réel par un point rouge situé à la gauche du code Un exemple de fichier Ned en mode "source" & "Graphique" sont présentés dans la (Figure14) et (Figure 15)

```
network WirelessA
{
    parameters:
        string hostType = default("WirelessHost");
        string mediumType = default("IdealRadioMedium");

        @display("bgb=650,500;bgg=100,1,grey95");
        @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);

        @figure[rcvdPkText](type=indicatorText; pos=420,20; anchor=w; font=,20; textFormat="packets received: %g"; initialValue=
        @statistic[rcvdPk](source=hostB_rcvdPk; record=figure(count); targetFigure=rcvdPkText);
        @signal[hostB_rcvdPk];
        @delegatesignal[rcvdPk](source=hostB.udpApp[0].rcvdPk; target=hostB_rcvdPk);

    submodules:
        physicalEnvironment: PhysicalEnvironment {
            @display("p=580,425");
        }
        visualizer: IntegratedCanvasVisualizer {
            @display("p=580,125");
        }
        configurator: IPv4NetworkConfigurator {
            @display("p=580,200");
        }
        radioMedium: <mediumType> like IRadioMedium {
```

Figure 14-Fichier Ned en mode text

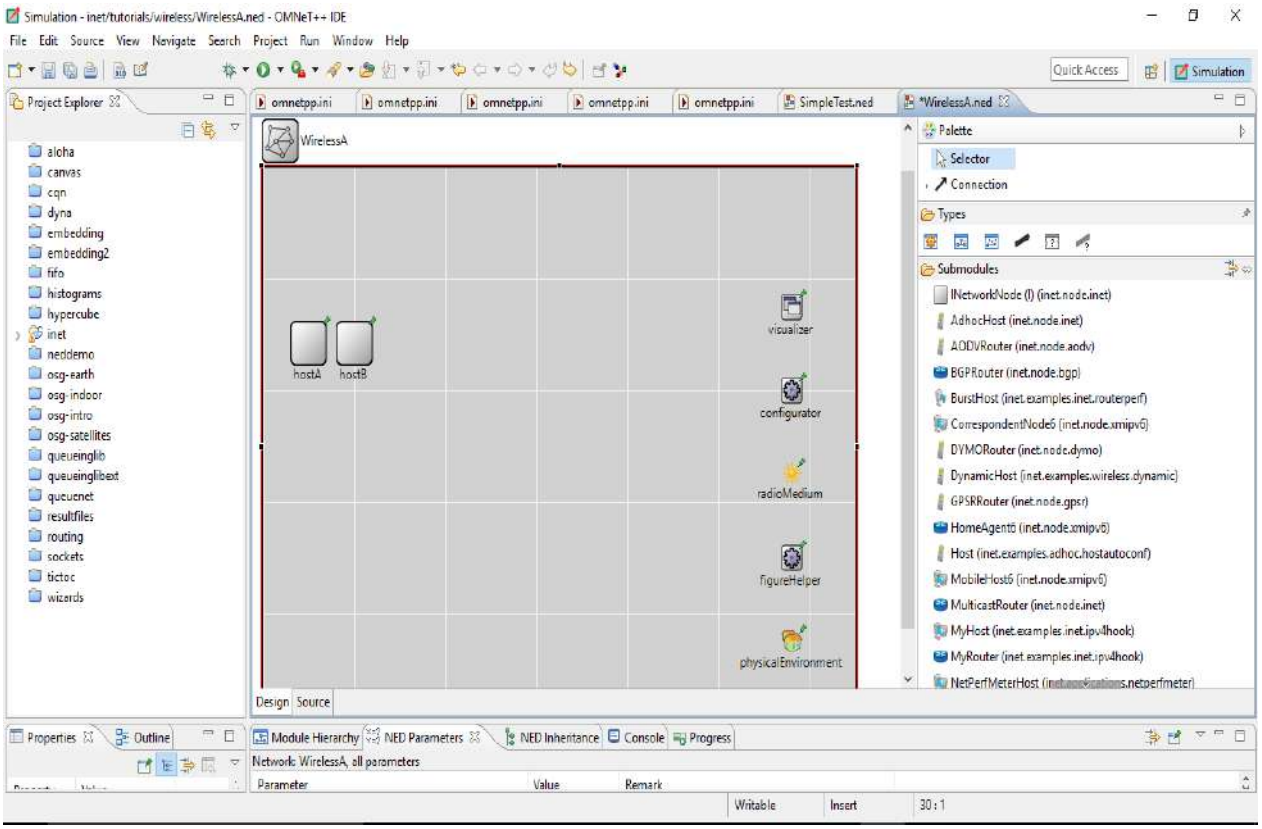

Figure 15-Ficher Ned en mode graphique

3.2Fichier(.ini) :

Est lié étroitement avec le fichier NED. Permet à l'utilisateur d'initialisé les paramètres des différents modules ainsi la topologie du réseau.

Voici un exemple présenté ci-dessous :

```
[Config Wireless01]

description = Two hosts communicating wirelessly
network = WirelessA
sim-time-limit = 300s

*.host*.networkLayer.arpType = "GlobalARP"

*.hostA.numUdpApps = 1
*.hostA.udpApp[0].typename = "UDPBasicApp"
*.hostA.udpApp[0].destAddresses = "hostB"
*.hostA.udpApp[0].destPort = 200
*.hostA.udpApp[0].messageLength = 512B
*.hostA.udpApp[0].sendInterval = 0.005s
####
*.hostA.udpApp[0].packetName = "UDPData"

*.hostB.numUdpApps = 1
*.hostB.udpApp[0].typename = "UDPSink"
*.hostB.udpApp[0].localPort = 200

#movement
*.hostB.mobilityType = "LinearMobility"
*.hostB.mobility.speed = 1mps
*.hostB.mobility.angle = 0deg
```

Figure 16-Exemple d'un fichier *.ini

3.3Fichier(.msg)

Les modules communiquent en échangeant des messages. C'est derniers peuvent être déclarés dans un fichier dont l'extension est(.msg) ou l'on peut ajouter des champs de données. OMNET++ traduira les définitions de messages en classes C++ le diagramme suivant peut donner une idée plus détaillée sur le développement d'exécution d'une simulation sous OMNET++

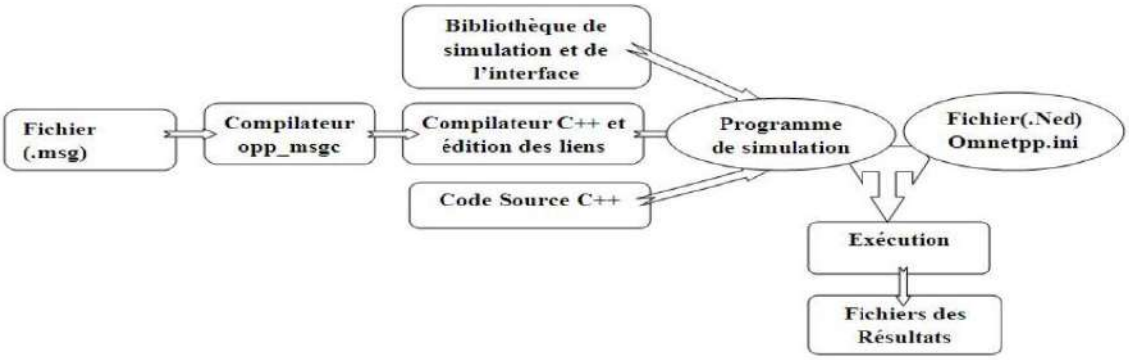

Figure 17-Exécution d'une simulation sous OMNET++[Site4]

4 Composant :

Application	FTP, Telnet, générateur de trafic (IPTrfGen..), Ethernet, Ping App, UDPApp, TCPApp
Transport	TCP, UDP, RTP
Réseau	IPv4, IPv6, ARP, OSPF, LDP, MPLS, ICMP, TED...
Liason	Mgmt, MAC, Radio
Node	Ad Hoc, Wireless, MPLS...

Tableau 1-Structure d'un nœud mobile dans OMNET++ [14]

5 Framework :

5.1 Mobility Framework :

La librairie MF (Mobility Framework) est une extension du simulateur OMNET++. Elle a été développée par une équipe de chercheurs à l'université de Berlin. Cette librairie est destinée à soutenir les réseaux sans fil et mobiles au sein de simulations OMNET++. En effet, elle permet une bonne manipulation des noeuds mobiles et la gestion des connexions dynamiques pour avoir un modèle de mobilité de réseau sans fils qui fournie des résultats le plus proche possible du monde réel. En outre, MF prévoit des modules de base qui peuvent être utilisés pour former de nouveaux modules. Avec ce concept, un programmeur peut facilement développer ses propres protocoles avec l'interface nécessaire [14].

Elle peut être utilisée pour la simulation de :

- Les réseaux sans fil fixes
- Les réseaux sans fil mobiles
- Les réseaux distribués (ad-hoc) et les réseaux centralisés
- Les réseaux de capteurs
- Les multi-réseaux sans fil

Aujourd'hui, il y'a un développement d'une bibliothèque de protocoles normalisés pour la MF (802,11, AODV, ...), dont l'objectif est de disposer d'une bibliothèque riche de ces protocoles afin de permettre facilement Plug-and-Play des simulations de différents types de protocoles largement utilisés

5.1.2 Les modules de mobilités :

le coeur de Mobility FrameWork est composé de **[13]**

Mobility Controller(Gestion de la mobilité).

- Channel Control(La gestion dynamique de connexion entre les hôtes.

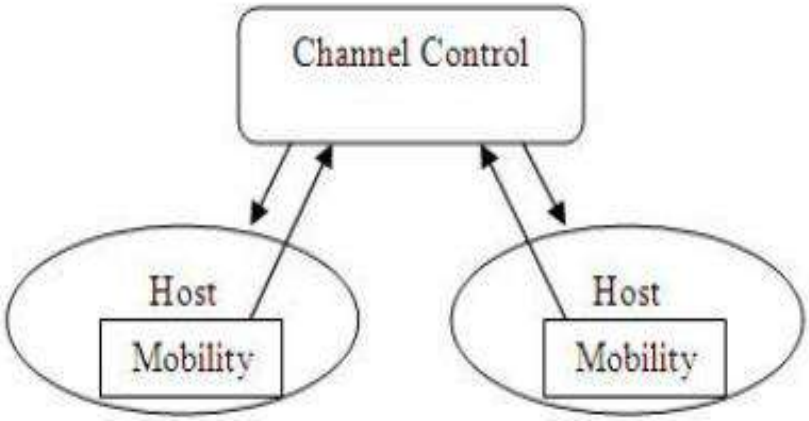

Figure 18-Archetecture de Mobility et Channel Control

5.1.3 la structure d'un hôte mobile :

La structure interne d'un noeud(Hôte)en Mobility FrameWork est schématisé dans la figure suivante **[13]**

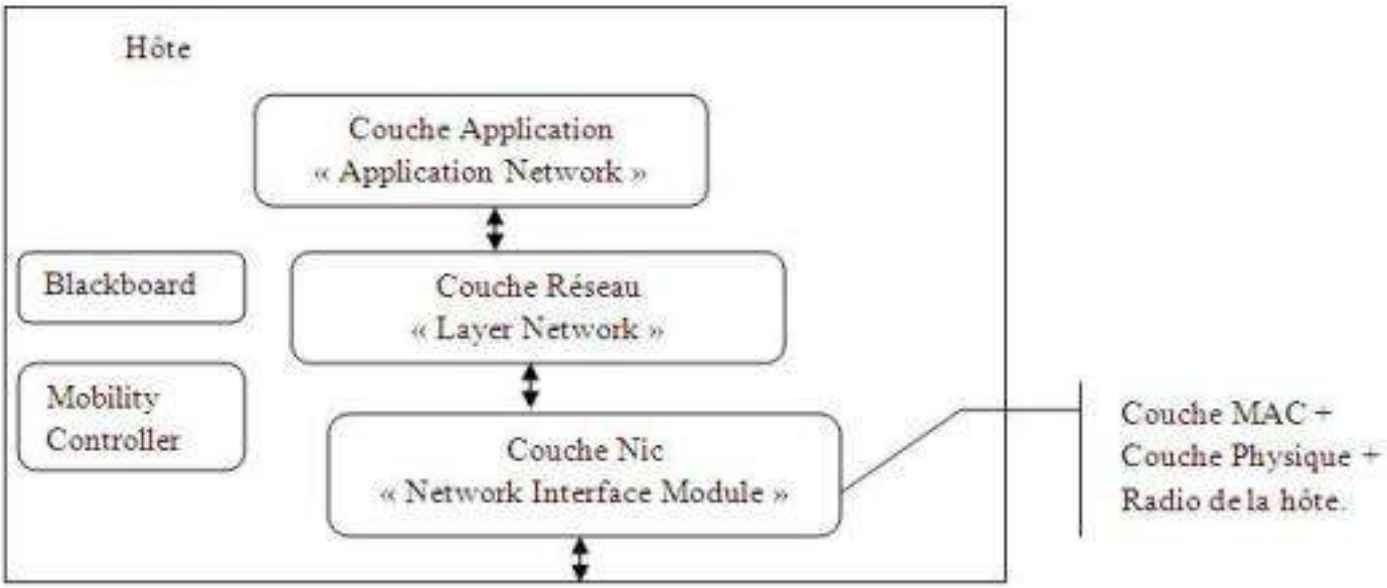

Figure 19-Architecture interne d'un nœud dans MF

6 INET Framework :

INET est une librairie open source pour la simulation des réseaux informatiques dans l'environnement OMNET++. Elle contient IPv4, IPv6, TCP, UDP, des protocoles implémentés, et plusieurs modèles d'application. Elle comprend également un modèle avec MPLS RSVP-TE et de signalisation LDP. La couche liaison sont des modèles de PPP(Point-to-Point Protocol), Ethernet et 802.11.Le routage statique peut être configuré à l'aide du réseau auto configurateur, ou on peut utiliser le protocole de routage mise en oeuvre.

INET prend en charge les simulations des réseaux sans fil et mobiles, ainsi les réseaux ad-hoc et les modèles MPLS, RSVP-TE et LDP , sans oublier le routage dynamique (RIP et OSPFv2)**[14]**

Dans ce paragraphe, nous allons présenter une étude de l'existant de la librairie INET et plus précisément l'implémentation des couches PHY, MAC, IP, RTP, UDP et TCP dans INET [15]

- **La couche PHY**

La couche physique est la partie essentielle d'un noeud sans fil. Il est responsable de l'envoi et la réception de message, détection de collision, et calcul d'erreur sur les bits. La couche physique est divisée en trois parties, qui sont décrites en détail dans les sous-sections suivantes :

- ✓ PhyLayer fournit les interfaces de la couche MAC et de la couche physique des autres noeuds.
- ✓ AnalogueModels sont responsables pour la simulation de l'atténuation d'un signal reçu.
- ✓ le Decider est responsable de l'évaluation et de démodulation des messages reçus.

- **La couche MAC**

La couche MAC IEEE 802.11b est implémentée dans la librairie INET, elle supporte le mécanisme RTS/CTS. Il n'y a que l'algorithme DCF implémenté, le PCF n'est pas implémenté. Le débit de transmission physique est constant pendant la simulation.

Le simulateur OMNET++ contient deux classes de bases pour former la couche MAC :

- BaseMACLayer pour encapsuler et décapsuler les paquets seulement.
- EyesMACLayer fournie un ensemble de fonction dont l'importante est de renvoyer les informations concernant la couche MAC d'un noeud.

- **La couche IP**

L'Internet Protocol (IPSuite) fournit des IPv4, TCP et UDP dans les modèles de simulations OMNET + +. D'abord, il a été développé par plusieurs personnes à l'Université de Karlsruhe. À la fin de l'année 2003, l'auteur de OMNET++, Andras Varga, a repris l'entretien de l'ensemble IPsuite. Il a également créé plus de documentation pour le paquet, ce qui rend plus facile à utiliser et faire des ajustements pour le modèle. Dans le même temps, un modèle l'IPv6 a été créé. Mais le routage IP est de implémenté de manière statique. Alors, l'utilisateur doit configurer les adresses IP et les adresses de groupes multicast dans les tables de routages avant la simulation. C'est au niveau IP que se passe le filtrage des paquets multicast.

Le routage multipoint au niveau IP est déjà implémenté mais de manière statique. C'est à dire que l'utilisateur doit configurer les adresses IP et les adresses de groupes multicast dans les tables de routages avant la simulation. C'est au niveau IP que se passe le filtrage des paquets multicast.

- **La couche RTP**

La couche RTP présente le niveau transport, elle n'est pas encore intégrée dans la librairie INET. Dans la version 2006-10-20, il y a une implémentation de la couche RTP réalisé par Matthias Opptiz et ajouté par Andras Vargas mais elle n'est pas encore intégrée. Le problème de cette

implémentation est qu'elle a été faite dans une ancienne version de l'année 2001 et que l'architecture globale de la librairie INET a changé.

- **La couche UDP**

La couche UDP est implémentée dans la librairie INET. Elle présente le niveau transport et elle est très utilisable dans la simulation car son implémentation est simple et elle est rapide par rapport aux autres couches de même niveau.

- **La couche TCP**

Dans la librairie INET, cette couche est une couche de transport comme UDP et RTP et elle est implémentée à l'aide des sockets.

- **La couche Application**

Elle présente la couche supérieure. De nombreuses applications sont implémentées dans la librairie INET comme Ethernet, TCPApp, PingApp, UDPApp

6.1 Catalogue de modèles :

Les composants de modèle suivants (protocoles, applications et autres modèles) sont disponibles pour INET Framework [15]

	Protocol	Projet
Application	CBR/VBR , HTTP, File Transfer , DHCP…	INET
Transport	TCP,UDP,SCTP,RTp,RTCP	INET
Reseaux	IPv4, ICMPV4 ,ARP ,IGMPv3 ,IPv6	INET
Routage	link-state routing , OSPFv2(1) , OSPF(2) , BGPv4 , BGP(2) ,RIP	INET
Manet Routage	AODV , DYMO , GPSR , DSDV, DSR,OLSR	INET
Fils	PPP, Ethernet, STP ,RSTP ,TTE ,802.1avb ,EPON , TDM/WDm-PON	INET
Sans fils	802.11 , 802.11p , 802.1e ,802.15.4, LTE(User-Plane) ,LTE(Control-Plane)	INET
Mobility/Envirenment	Various Mobility Models	INET

Tableau 2-La liste des principaux composants de modèle disponible dans INET FW

7. Structure d'un nœud mobile dans OMNET++ :

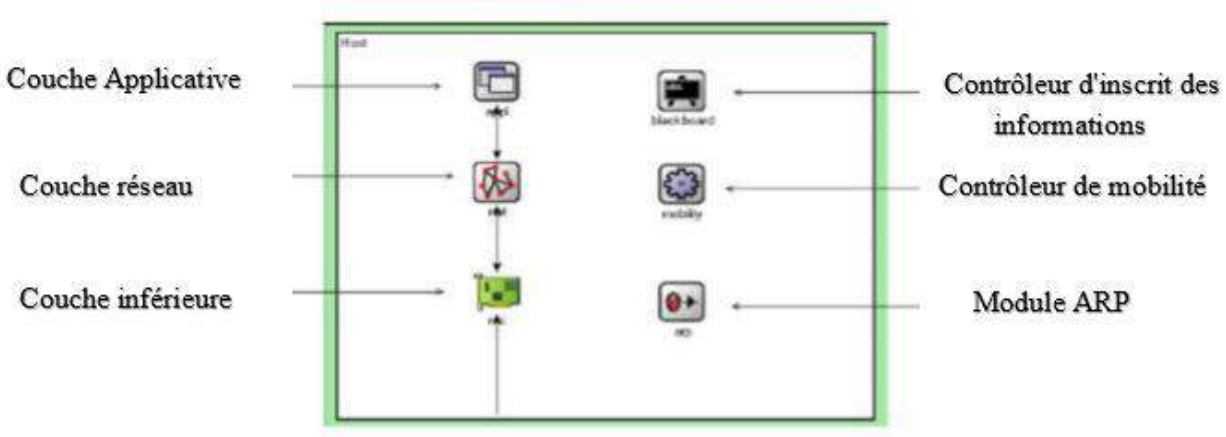

Figure 20-Structure d'un nœud mobile dans OMNET++ [14]

8 L'avantage et inconvénients du simulateur OMNET++ [16] :

8.1 Avantages

- Architecture modulaire permettant l'intégration de nouveau modèle.
- Utilisation du C++ pour le développement du noyau.
- Les classes de base du simulateur peuvent être étendue et personnalisées.
- Conception de modèle rapprochant de la réalité.
- La mise en route avec ce simulateur est assez simple grâce à une conception claire du simulateur.
- Il fournit également une puissante bibliothèque d'interfaces graphiques pour l'animation et la gestion du débogage.
- Des facilités d'utilisation et d'exploitation des résultats grâce à des outils permettant de créer des fichiers facilement récupérables pour d'autres applications.

8.2 Inconvénients

- Description des modèles en langage NED.
- Il y a un manque cruel de protocoles disponibles dans la bibliothèque comparé à d'autres simulateurs Le nombre de protocole n'est pas assez grand.

IV. Conclusion

Dans ce chapitre, nous avons tout d'abord représenté les capacités et le caractère des deux simulateurs NS-2 et OMNET++ et environnements de simulation existant sur les deux simulateurs. Dans le chapitre suivant, nous allons parler de la différence qualitative entre les deux simulateurs

Chapitre III : Comparaison qualitative entre « NS-2 » & « OMNET++ »

Chapitre III : Comparaison qualitative entre « NS-2 » & « OMNET++ »

I.Introduction

Dans cette section, nous allons mener une comparaison entre les deux simulateurs en ce qui concerne le paramétrage des simulations et les modèles implémentés dans :

- L'emplacement des noeuds,
- Le scénario de mobilité,
- Terrain physique,
- Génération de trafic,
- Modules de protocoles implémentés.

Nous définissons en premier lieu les modèles, ensuite un tableau comparatif montre les similitudes et les différences entre les simulateurs dans l'implémentation de ces modèles.

1.Placement des noeuds

- **Random** : les noeuds sont placés aléatoirement sur le terrain physique (les dimensions du terrain sont précisées par l'utilisateur).
- **Uniform** : en se basant sur le nombre de noeuds, le terrain est divisé en cellules. Dans chaque cellule, un noeud est placé aléatoirement.
- **Grid** : le premier noeud se place à une position bien déterminée. Les autres noeuds sont placés en ordre à des distances égales pour former une sorte de grille. L'utilisateur précise toutes les informations nécessaires pour pouvoir construire cette grille.
- **A partir d'un fichier** : un fichier contient l'emplacement de chaque noeud.
- **Manuellement** : à laide d'un environnement graphique, en utilisant le « glisser poser ».

	NS-2	OMNEt++
Random	X	X
Uniform		X
Grid		X
A partir d'un fichier	X	x
Manuellement		X

Tableau 3-Comparatif entre NS-2 et OMNET++ dans le placement des nœuds

2.Scénario de mobilité

- **Aucun** : les noeuds sont fixes
- **Random waypoint** : Le noeud choisit aléatoirement une destination sur le terrain de et s'y rend avec une vitesse moyenne choisie. Lorsqu'il atteint sa destination, le noeud fait une pause. Cette procédure se répète jusqu'à la fin de la simulation.
- **Fichier de trace** : la trajectoire des mobiles est extraite d'un fichier de trace. Cette méthode permet ainsi de décentraliser l'implémentation des modèles à l'extérieur du simulateur.
- **le Linear Mobility Model**: les nœuds se déplacent en ligne droite avec un certain angle et cet angle ne change que lorsque le nœud mobile heurte un mur: il se réfléchit ensuite sur le mur selon le même angle

	NS-2	OMNET++
Aucun	X	X
Random waypoint	X	X
Fichier de trace	X	X
le Linear Mobility Model		X

Tableau 4-Comparatif entre NS-2 et OMNET++ dans le Scénario de mobilité

3.Terrain physique

- **Flat grid :**une grille plate donnant les dimensions du terrain.
- **Un plan bitmap :** le terrain peut indiquer la présence d'obstacles en plus il peut intégrer une image comme arrière-plan tel qu'une carte.

	NS-2	OMNET++
Flat grid	X	X
Un plan bitmap	X	X

Tableau 5-Comparatif entre NS2 et OMNET++ dans le modèle du terrain physique

4.Génération de trafic

- A partir d'un fichier : détermine les sources, les puits, le type du flux qui transite entre eux avec les paramètres nécessaires pour décrire ce flux comme par exemple le temps du commencement du trafic, la taille des paquets à envoyer etc.

- Manuellement : a l'aide de l'outil graphique on accède à chaque noeud individuellement pour le configurer.

	NS-2	OMNET++
A partir d'un fichier	X	X
Manuellement		X

Tableau 6-Comparatif entre NS2 et OMNET++ génération du trafique

5. Modules implémentés

Chacun des simulateurs « NS-2 » et « OMNET++ » implémente un ensemble de modules. Chaque module présente un protocole ou un modèle qui caractérise une couche donnée. Comme le montre le (tableau 7), les deux simulateurs ne présentent pas des similitudes dans leurs modules implémentés. Cette diversité rend la comparaison plus difficile. En effet pour pouvoir comparer le comportement des deux simulateurs, il faut les paramétrer de la même façon (même protocoles, même modèles) pour pouvoir par la suite tirer des conclusions. Mais ce qui rassure dans cette situation c'est le fait qu'on peut la plupart des cas trouvé des intersections entres les deux simulateurs.

Couche	NS-2	OMNET++
Liaison de donnés (Mac)	Ether, WLANAD, WLANIN, LBP, STP ARP, 2F-BLSR	Mgmt, MAC, Radio
Réseau (routage)	routage dans les réseaux ad-hoc (aodv, dsr,dsdv, tora, amodv),	Link-state routing, OSPF (INET), OSPF (Quagga), BGP (INET), RIP (INET), BGP (Quagga), RIP (Quagga), STP, RSTP, MANET: AODV, DYMO-UM, DYMO-FAU, DSDV, DSR, DSR, OLSR
Transport	TCP, UDP, RTP/RTCP, SDP	TCP (INET), TCP (lwIP), TCP (NSC), UDP, SCTP, RTP, RTCP
Application	générateur de trafic (CBR, FTP), http et Telnet	FTP, Telnet, générateur de trafic (IPTrfGen..), Ethernet, Ping App, UDP App, TCP App.

Tableau 7-Ensemble des modules implémentés par les simulateurs NS-2 et OMNET++.

- ✓ Ce tableau indique la différence entre les deux simulateurs.

	NS-2	OMENT++
licence	open Source	open Source
GUI	non	good
Interface	C++ ,OTcl	C++ ;Ned
Modules disponibles	filaire, Wireless, Ad-Hoc	-Réseaux capteurs sans fils -Réseau filaire. -il peut simuler n'importe quel type de réseau.
Documentation et Support Utilisateurs	Bonne documentation.	La documentation est très bonne et contient tous ce qu'on a besoin pour la simulation (définitions, méthodes, modules, implémentations, etc).
extensible	oui	oui
Popularité (2010)	**88%**	**2%**
Parallélisme	non	MPI/PVM

Tableau 8-Comparaison des simulateurs NS-2 et OMNET++ de réseau.

- ✓ Comparaison des paramètres de protocole entre les deux simulateurs

Simulateur / Protocol	NS-2	OMENT++(Inet FW)
IEEE 802.11 CSMA/CA MAC,	X	X
UDP	X	X
FTP	X	X
CBR	X	X
DSR	X	X
AODV	X	X

Tableau 9-Comparatif entre NS-2 et OMNET++ dans les paramètres de protocole implémenter

II. Conclusion

Dans le chapitre precedant nous avons presente les capacites des deux simulateur ns et omnet .dans ce chapitre , une comparaison qualitative a été menée. Cette comparaison permet de spécifier les ressemblances et les différences entre les deux simulateurs, dans la réalisation des simulations (placement des nœuds, mobilité, terrain physique, génération de trafic, paramètres de configuration de la simulation) et dans les modules de protocoles implémentés.

Cette comparaison qualitative servira d'hypothèse pour la réalisation de la comparaison quantitative. C'est l'objectif du chapitre suivant qui consiste a réaliser quelques scénarii pour tirer une comparaison entre les deux simulateurs en terme de résultats et de temps de simulation.

Chapitre IV : Comparaison quantitative entre « NS-2 » & « OMNET++ »

Chapitre IV : Comparaison qualitative entre « NS-2 » & « OMNET++ »

I. Introduction

Dans le chapitre précédent, la comparaison entre NS2 et OMNET++ est théorique. Cette comparaison ne suffit pas pour mesurer les performances de NS2. En effet, pour évaluer ce simulateur, il est nécessaire de passer à l'expérimentation en réalisant un ensemble de scénarii avec les deux simulateurs. Ensuite, la comparaison est faite en termes de résultats et de temps de simulation.

II. Paramètres de simulation

Dans les scénarios que nous avons réalisés, un ensemble de paramètres communs aux deux simulateurs est fixé. Ces paramètres sont récapitulés dans le tableau suivant :

Paramètre	Valeur
Modèle de propagation Radio	Tow Ray Ground
Couche MAC	IEEE 802.11
Protocole de routage	AODV
Couche transport	UDP
Type du trafic généré	CBR
Débit du trafic	200 paquets/seconde
Taille paquet	512 octets
Bande passante du lien radio	11 M bits/s
Portée de transmission	250 mètres

Tableau 10-Paramètres communs aux scénarii

La couche MAC est implémentée selon la norme IEEE 802.11b. En particulier, le mécanisme d'accès utilisé dans cette couche est la méthode DFC (Distributed Coordination Function) qui est typiquement le mécanisme CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance). La norme IEEE 802.11b opère à des débits compris entre 1 et 11 Mbits/s.

La variation de débits dépend de la qualité de l'environnement radio (interférences, obstacles, distances entre équipements).

AODV est choisi comme protocole de routage puisqu'il est implémenté par les deux simulateurs.

Le débit du trafic ainsi que la taille des paquets ont été sélectionnés, car d'après des études antérieures comme, nous avons de meilleurs résultats avec ces choix.

Le flux CBR est choisi vu qu'il est simple. Il est caractérisé par un débit constant et typiquement utilisé par les applications vidéo et audio. Ce type d'application exige des délais stricts. Un flux CBR est généré en fixant la taille du paquet (dans notre cas 512 octets) et le temps d'inter-arrivée entre deux paquets consécutifs (dans notre cas 0.005s).

1. Portée de transmission

1.1 Description

Avec NS-2, la portée de transmission peut être déterminée en affectant la valeur désirée en mètres au paramètre « Transmission Range » qui se trouve dans le module « WPHY » (Wireless PHY sical-layer module). Avec OMNET++ la portée de transmission peut être déterminée en affectant la valeur désirée en mètres au paramètre « Communication Range » Dans ce qui suit, nous allons appliquer ces conditions pour la configuration de la portée de transmission à 250 mètres.

1.2 Résultats de simulation

Pour vérifier qu'effectivement la portée de transmission vaut 250 mètres, nous procédons à la simulation suivante : Une topologie contenant deux nœuds et. Le nœud envoie au nœud un flux CBR. Le trafic commence à 1s et s'achève à 250s. Le temps de simulation étant 300s et le protocole de routage étant AODV. Le nœud est fixe, alors que le nœud se déplace vers la droite avec une vitesse égale à 1m/s. La distance initiale entre les deux nœuds vaut 50 mètres.

Figure 21-Scénario pour la vérification de la portée de transmission

En se déplaçant, la distance (N1, N2) augmente, ainsi peu à peu, le mobile N2 quitte la zone de couverture de N1. Cet événement est marqué par le fait que N2 ne reçoit plus les paquets envoyés par N1. Par conséquent, le nombre de paquets reçus devient constant. Si nous mesurons à cet instant la distance (N1, N2), nous devons s'attendre à une valeur égale ou proche des 250 mètres

1.2.1 Appliqué le scénario dans le simulateur NS-2

Cette figure représente fichier scripte.tcl"aodv18-1.tcl" Il contient la position initiale des Noeud 0 et Noeud 1 et génération de mouvements nœud 1 et setup UDP connexion

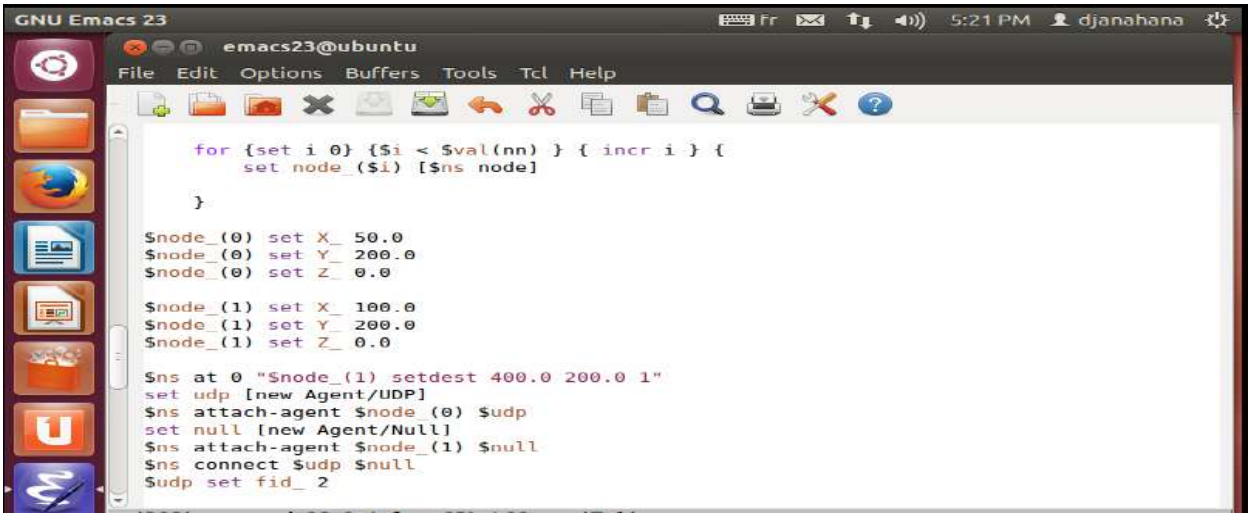

Figure 22-scénario1.tcl

L'exécution de ce script se fait, grâce à ns, en utilisant simplement la commande :
« scénario1.tcl» .

Figure 23-Exécution fichier. tcl

Cette(figure 24) illustre l'envoi des paquets entre Noeud 0 et Noeud 1 dans visualisation Ns-2 « nam »

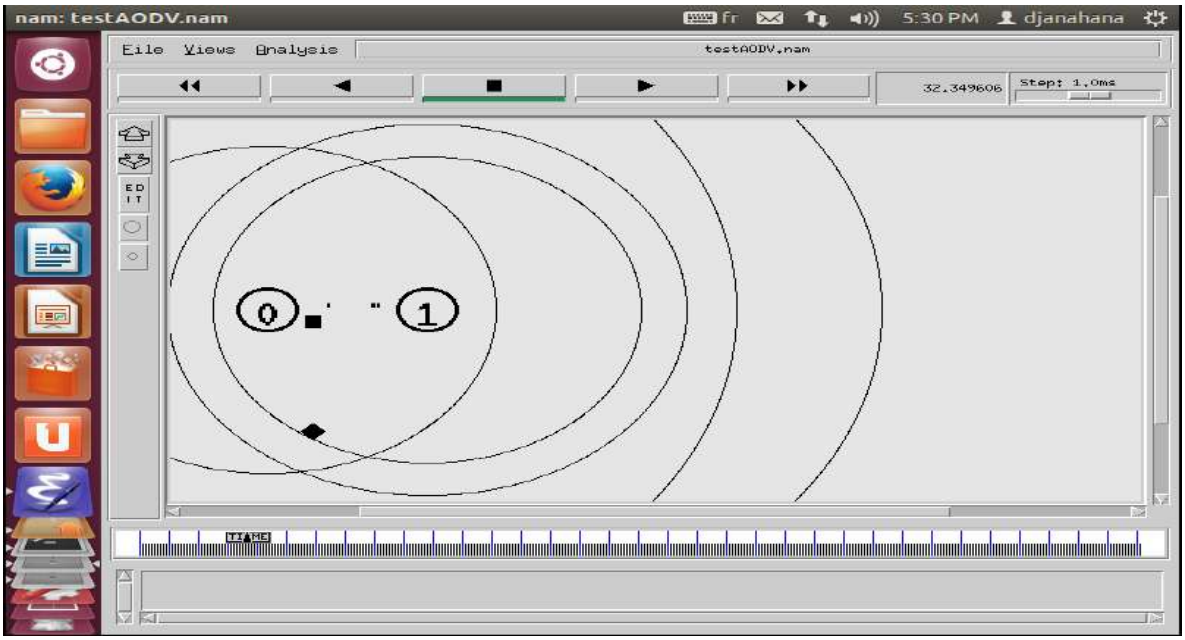

Figure 24-NAM, visualisation représenter la topologie d'un réseau décrit avec NS-2

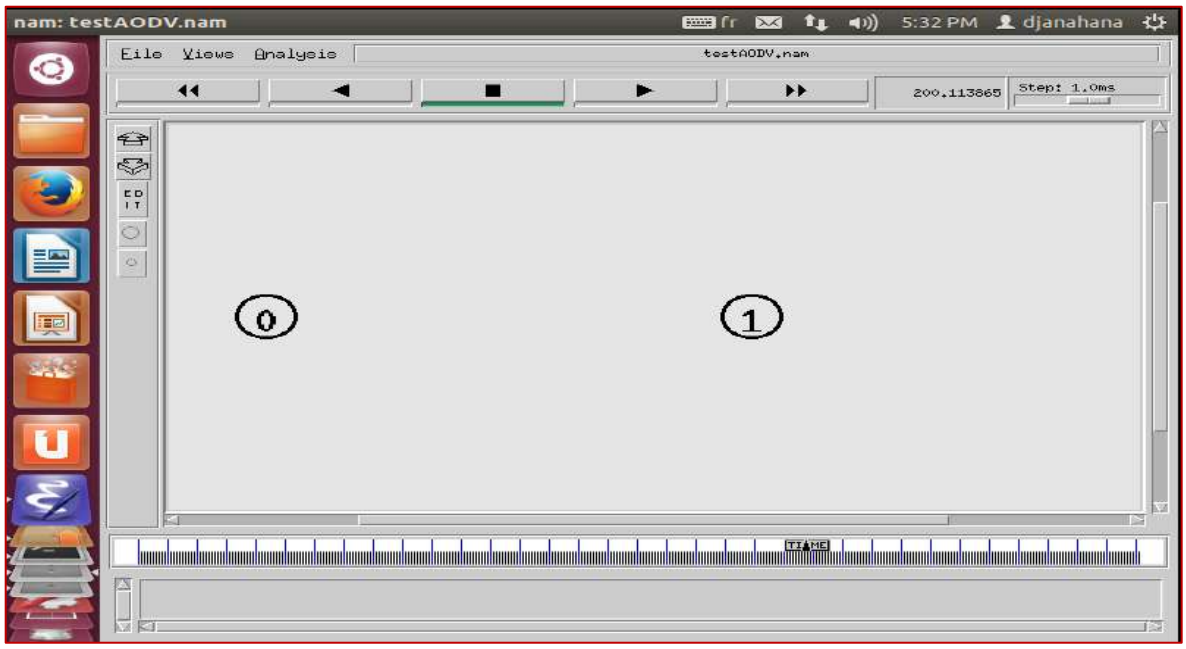

Figure 25-NAM, visualisation représenter la topologie d'un réseau décrit avec NS-2

Lire le fichier de trace fichier.tr voir(figure 26)à l'aide de la commande more. À chaque évènement arrivant au niveau d'un lien, une ligne est écrite dans ce fichier :

- **r** signifie que le paquet est reçu par le nœud destinataire
- **d** signifie que le paquet est rejeté (perdu).
- **s** Une nœud envoie le paquet « send »

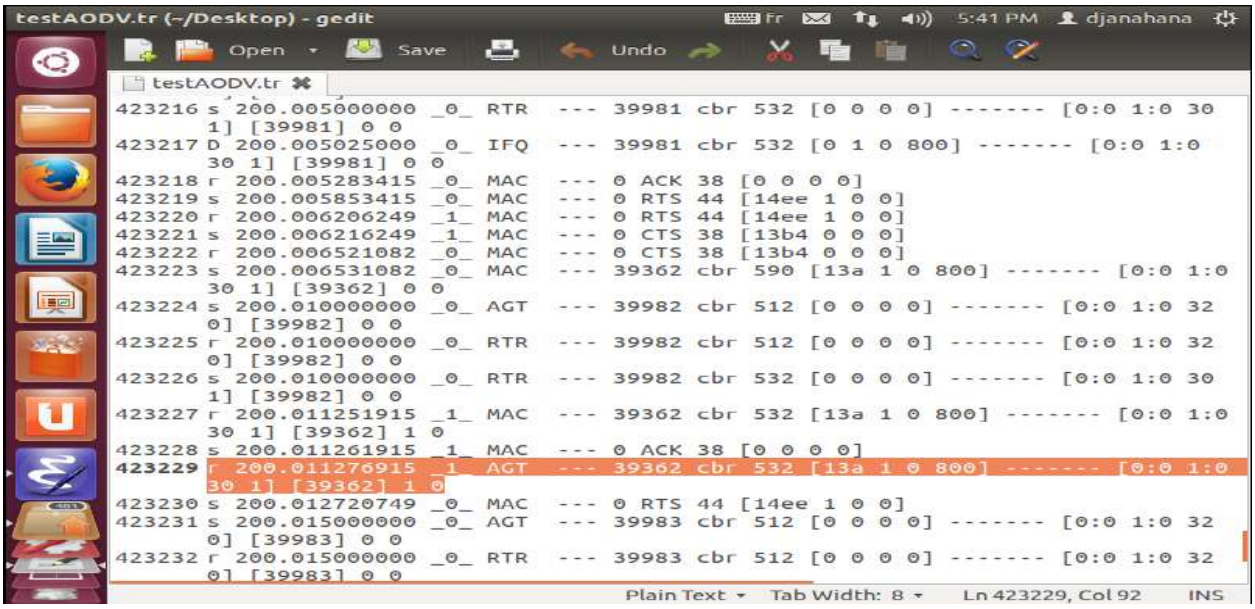

Figure 26-Traces de la simulation

Dans cette (figure 26), nous trouvons View Terrace contient des données résultant des simulations, et indique le nombre de paquets envoyés et le moment où ceux-ci sont envoyés les paquets et le temps d'arrêt envoyer et recevoir des paquets sont présentés dans la ligne rouge. Et indique le nombre de paquets reçus.

1.2.2 Appliqué le scénario dans le simulateur OMNET++

Cette figure représente la source NED du réseau Il contient la position initiale des hostA et hostB

Figure 27-Ficher Ned en mode graphique

```
network WirelessA
{
    parameters:
        string hostType = default("WirelessHost");
        string mediumType = default("IdealRadioMedium");
        @display("bgb=650,500;bgg=100,1,grey95");
        @figure[title](type=label; pos=0,-1; anchor=sw; color=darkblue);
        @figure[rcvdPkText](type=indicatorText; pos=420,20; anchor=w; font=,20; textFormat="packets received: %g"; initialVal
        @statistic[rcvdPk](source=hostB_rcvdPk; record=figure(count); targetFigure=rcvdPkText);
        @signal[hostB_rcvdPk];
        @delegatesignal[rcvdPk](source=hostB.udpApp[0].rcvdPk; target=hostB_rcvdPk);
    submodules:
        physicalEnvironment: PhysicalEnvironment {
            @display("p=580,425");
        }
        visualizer: IntegratedCanvasVisualizer {
            @display("p=580,125");
        }
        configurator: IPv4NetworkConfigurator {
            @display("p=580,200");
        }
        radioMedium: <mediumType> like IRadioMedium {
            @display("p=580,275");
        }
        figureHelper: DelegateSignalConfigurator {
            @display("p=580,350");
        }
        hostA: <hostType> like INetworkNode {
            @display("p=50,158");
        }
        hostB: <hostType> like INetworkNode {
            @display("p=100,158");
```

Figure 28-Fichier Ned en mode text

```
[Config Wireless01]
description = Two hosts communicating wirelessly
network = WirelessA
sim-time-limit = 300s
*.host*.networkLayer.arpType = "GlobalARP"
*.hostA.numUdpApps =1
*.hostA.udpApp[0].typename = "UDPBasicApp"
*.hostA.udpApp[0].destAddresses = "hostB"
*.hostA.udpApp[0].destPort = 200
*.hostA.udpApp[0].messageLength = 512B
*.hostA.udpApp[0].sendInterval = 0.005s
**.udpApp[0].packetName = "UDPData"
*.hostB.numUdpApps = 1
*.hostB.udpApp[0].typename = "UDPSink"
*.hostB.udpApp[0].localPort = 200
#movement
*.hostB.mobilityType = "LinearMobility"
*.hostB.mobility.speed = 1mps
*.hostB.mobility.angle = 0deg
#configure IEEE802.11 MAC
*.host*.wlan[0].typename = "Ieee80211Nic"
*.host*.wlan[0].mgmtType = "Ieee80211MgmtAdhoc"
```

Figure 29-Fichier configuration .ini

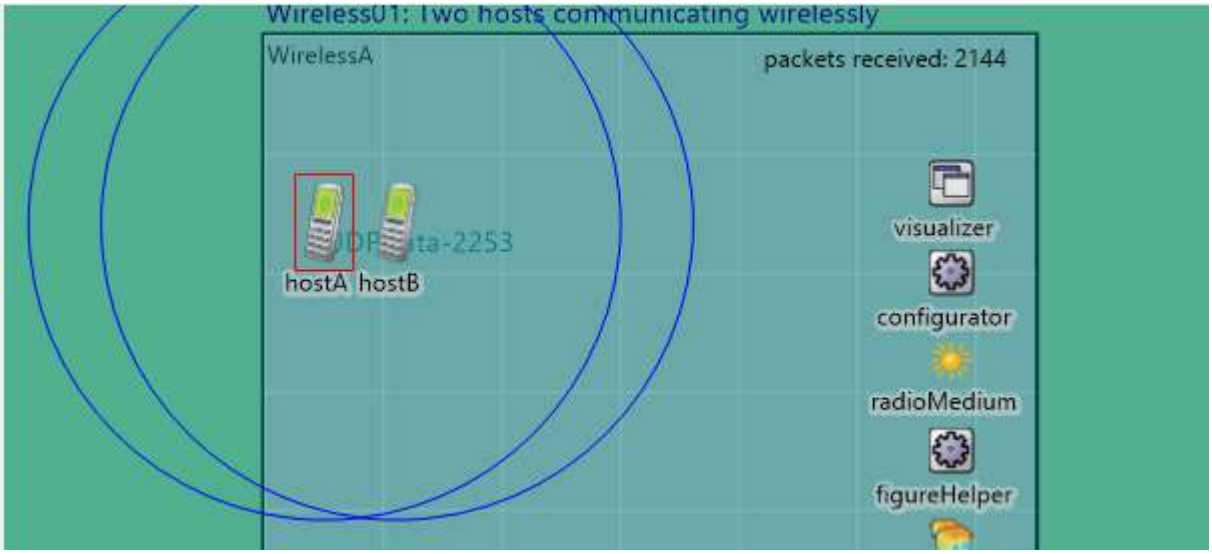

Figure 30-TKenv, visualisation représenter la topologie d'un réseau décrit avec OMNET++

1.3 Etude comparative

La figure suivante présente le nombre de paquets reçus par le mobile N1 en fonction de la distance séparant les noeuds N0 et N1.

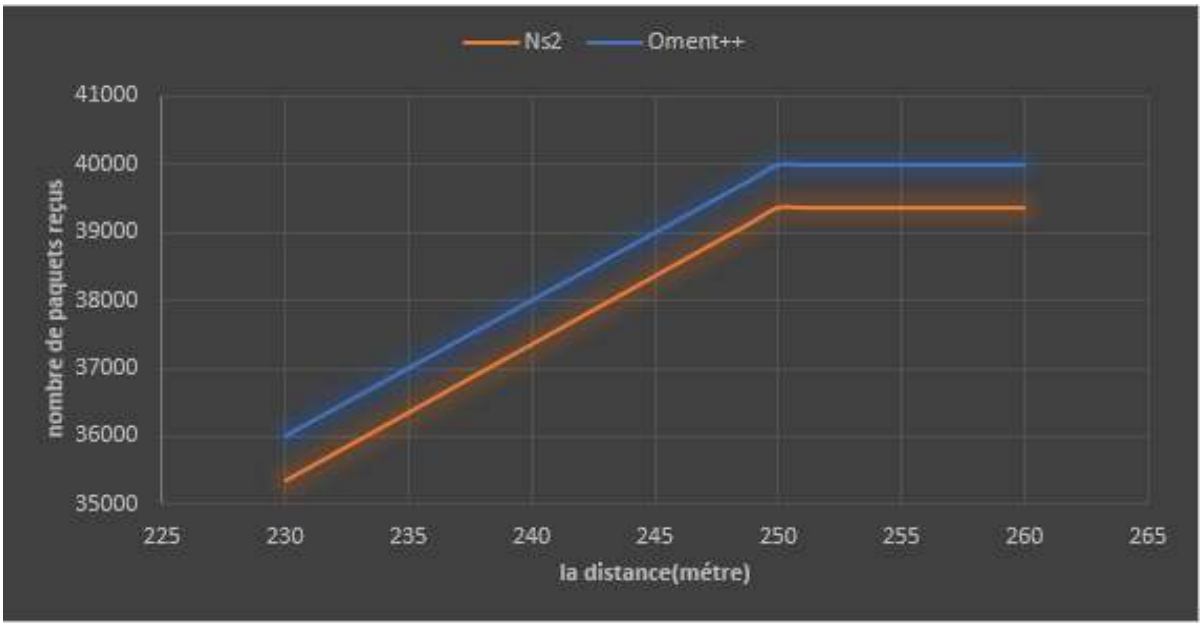

Figure 31-Nombre de paquets reçus par le mobile N1 en fonction de la distance (N0 ,N1)

Cette (figure 31) montre bien que la portée de transmission de NS-2 vaut 250 mètres. Par conséquent le taux de paquets reçus avec les deux sera à 39362 paquets pour NS-2 et 40000 paquets pour OMNET++). En effet, si nous fixons cette distance, deux mobiles ne peuvent communiquer que si la distance qui les sépare est inférieure ou égale à 250 mètres. Alors qu'en réalité le nœud

source peut réussir à envoyer des trames même si la distance qui le sépare du nœud destinataire excède les 250 mètres, car dans ce cas, le calcul des interférences et du bruit décidera si la communication est possible ou pas.

2. Méthode d'accès au medium

2.1 Description

La méthode d'accès au médium que nous adoptons pour les simulateurs NS-2 et OMNET++ est : « CSMA/CA » avec le mécanisme « RTS/CTS ». En bref, cette méthode permet l'accès au média en mode avec contention. Son principe de fonctionnement est le suivant : Une station voulant transmettre, écoute le support de transmission, si le support est occupé (une autre station est en train de transmettre), alors la station remet sa transmission à plus tard. Si le support est libre pour un temps « DIFS » (Distributed Inter Frame Space), alors la station va transmettre une trame « RTS » qui a pour but de réserver le canal pendant la durée nécessaire à la transmission.

Une fois la trame « RTS » reçue par la station destinataire, celle-ci va attendre un temps inférieur à « DIFS » (temps « SIFS ») pour ensuite transmettre à son tour une trame « CTS » qui a pour but d'avertir toutes les stations à sa portée, que le canal sera occupé pendant le temps recalculé de la trame « RTS » précédente. Ceci permet en effet d'éviter d'éventuelles collisions pendant la transmission des flux de données. Ce mécanisme peut être considéré comme une tentative d'introduire la qualité de service sur le lien WLAN afin d'être apte à transférer des flux multimédia (le cas du trafic CBR). **[Site 4]**

2.2 Résultats de simulation :

Le but du scénario suivant est de créer une compétition sur l'obtention du canal de transmission. Ce scénario consiste à placer quatre nœuds équidistants de 50 mètres. (Figure 32) Les nœuds N0 et N3 envoient simultanément un flux CBR (de 0.1s à 30s) aux nœuds N1 et N3. Le temps de simulation étant 50s. La distance (N0, N3) est égale à 150 mètres. Dans cette situation, les quatre nœuds se situent dans la zone de couverture de N1 et la probabilité de collision sera élevée.

Figure 32-Scénario de test de la méthode d'accès

2.2.1 Appliqué le scénario dans le simulateur NS-2 :

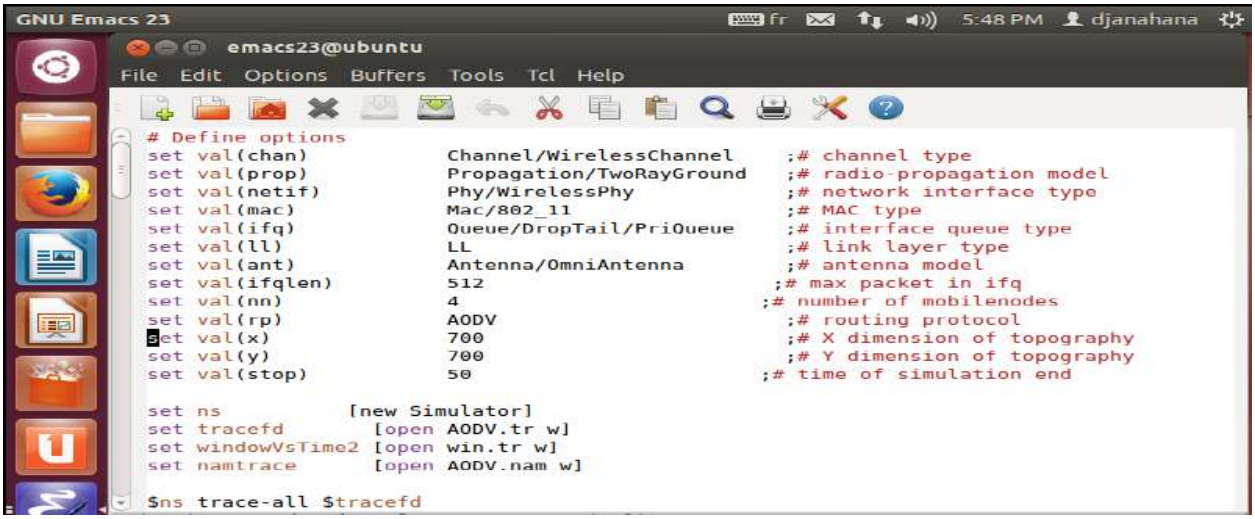

Figure 33-Le script fichier.tcl définition des variables

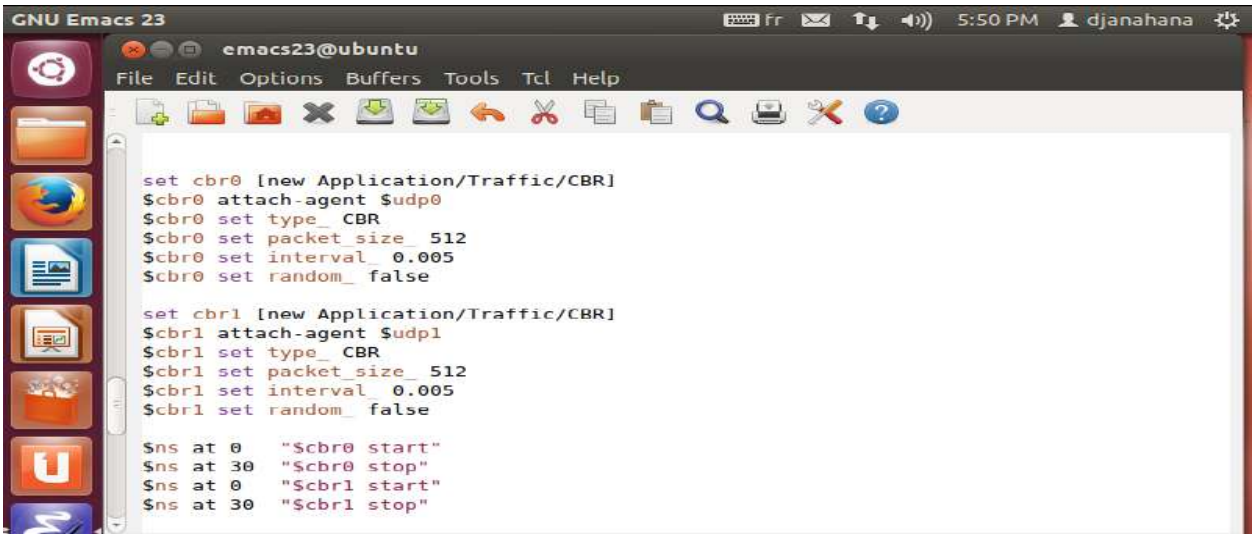

Figure 34-scénario2.tcl

- ✓ Noeud 0 envoie des paquets vers Noeud 1
- ✓ Noeud 2 envoie des paquets vers Noeud 3
- ✓ Toutes les **0.005 s** secondes envoie un paquet
- ✓ Le début des temps d'envoyer des paquets **0.1s**
- ✓ Temps arrêt l'envoi des paquets 30s

Après l'application de fichier de simulateur nous trouvons les résultats dans le fichier.tr.

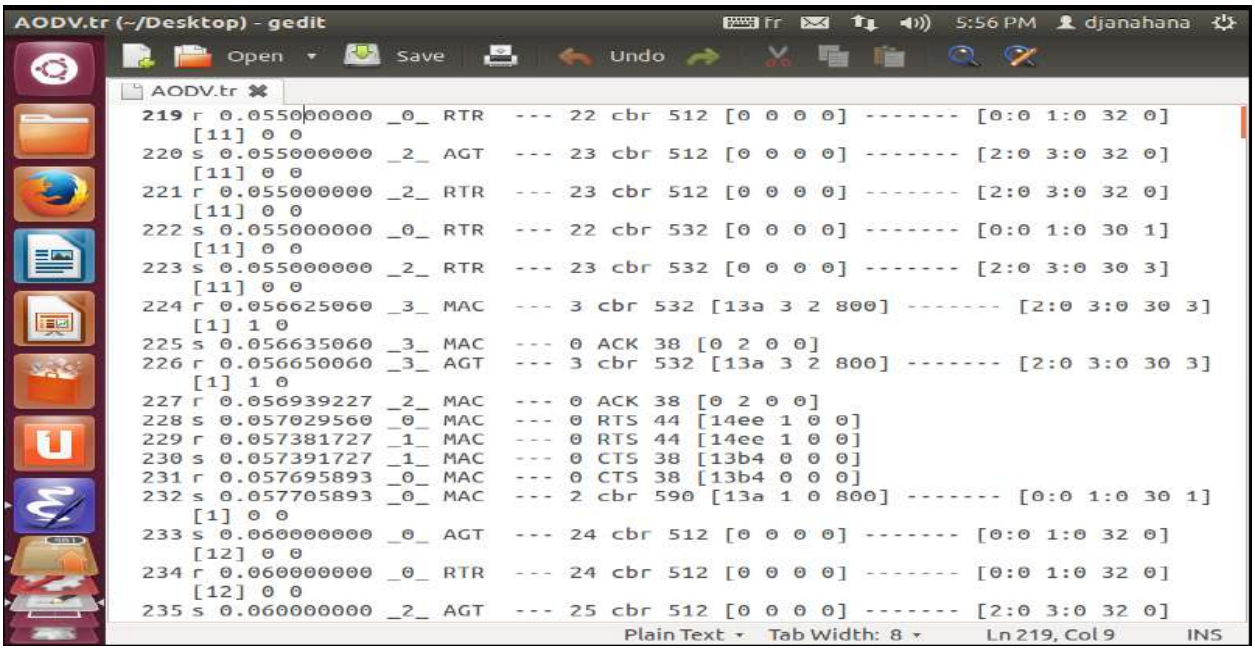

Figure 35-fichier.tr

2.2.2 Appliqué le scénario dans le simulateur OMNET++ :

```
description = Two hosts communicating wirelessly
network = WD
sim-time-limit = 50s


*.host*.networkLayer.arpType = "GlobalARP"
**.host{1..2}.numUdpApps = 1
**.host{1..2}.udpApp[*].typename = "UDPBasicApp"
**.host{1..2}.udpApp[0].localPort = 100
**.host{1..2}.udpApp[0].destPort = 100
**.host{1..2}.udpApp[0].messageLength = 128 bytes
**.host{1..2}.udpApp[0].sendInterval = 0.01s
**.host1.udpApp[0].destAddresses = "host3"
**.host2.udpApp[0].destAddresses = "host4"

**.host{3..4}.numUdpApps = 1
**.host{3..4}.udpApp[*].typename = "UDPSink"
**.host{3..4}.udpApp[0].localPort = 100

*.host*.wlan[*].typename = "WirelessNic"
*.host*.wlan[*].macType = "CsmaCaMac"
*.radioMediumType = "IdealRadioMedium"
*.host*.wlan[*].radioType = "IdealRadio"
```

Figure 36-Fichier configuration .ini

```
    submodules:
        physicalEnvironment: PhysicalEnvironment {
            @display("p=580,425");
        }
        visualizer: IntegratedCanvasVisualizer {
            @display("p=580,125");
        }
        configurator: IPv4NetworkConfigurator {
            @display("p=580,200");
        }
        radioMedium: <mediumType> like IRadioMedium {
            @display("p=580,275");
        }
        figureHelper: DelegateSignalConfigurator {
            @display("p=580,350");
        }
        host1: <hostType> like INetworkNode {
            @display("p=50,158");
        }
        host2: <hostType> like INetworkNode {
             @display("p=150,158");

        }
        host3: <hostType> like INetworkNode {
             @display("p=100,158");
        }
        host4: <hostType> like INetworkNode {
            @display("p=200,158");
        }
}
```

Figure 37-Fichier Ned en mode text

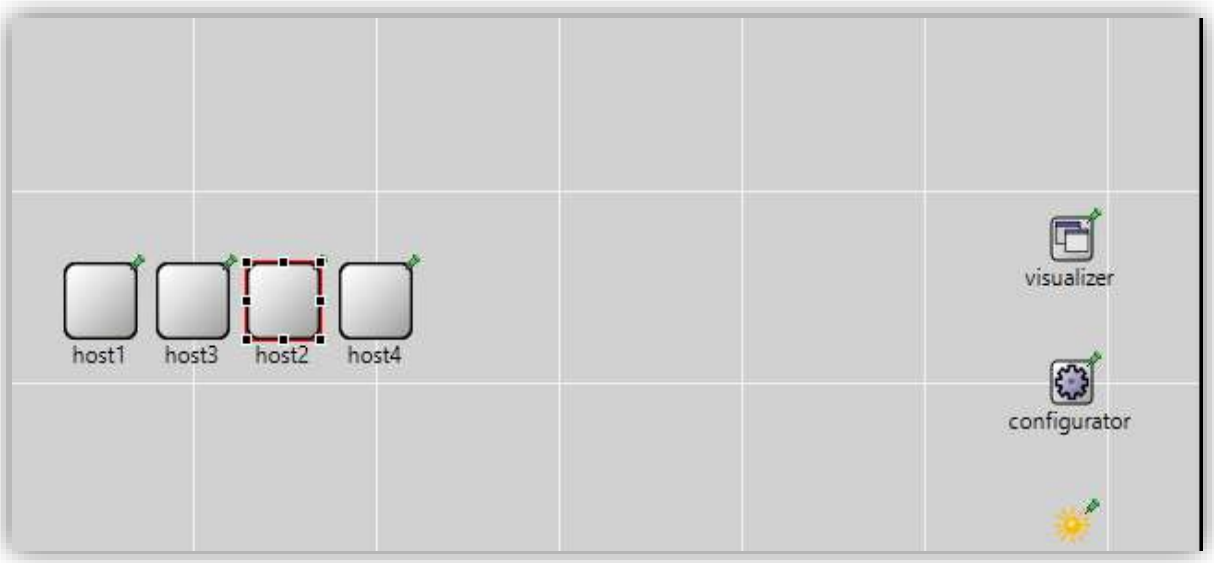

Figure 38-Ficher Ned en mode graphique

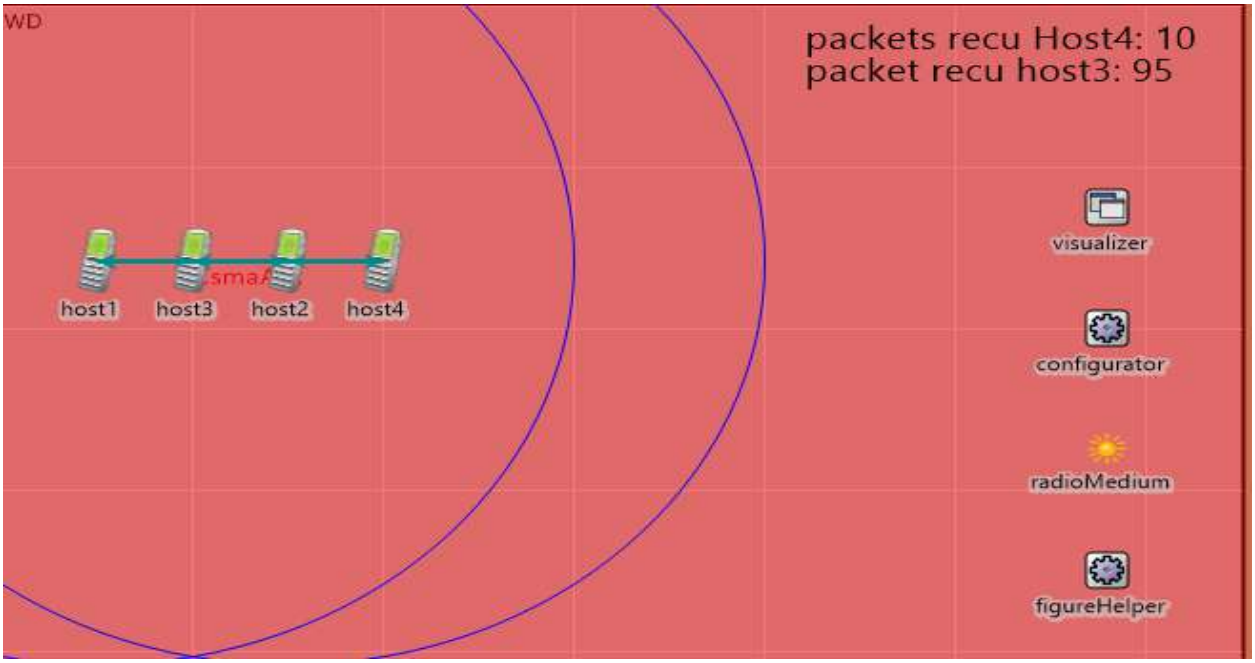

Figure 39-TKenv, visualisation représenter la topologie d'un réseau décrit avec OMNET++

2.3 Etude comparative :

Après avoir implémenté ce scénario avec les deux simulateurs, Après la simulation du scénario avec les deux simulateurs, nous avons constaté qu'aucun paquet n'est reçu par les nœuds destinataires N1 et N3 dans simulateur NS-2. Ceci est dû au phénomène suivant. Au début, chacun des nœuds N0 et N2 possède dans la file d'attente de la couche MAC un paquet à émettre. En

écoutant le canal, ils ne trouvent aucun signal car le canal est encore vide. Par conséquent, chacun d'eux envoie un message « RTS » pour demander l'accès au canal. Les deux trames « RTS » tombent en collision au niveau de N1. Ainsi, ce dernier ne reçoit pas la trame « RTS » venant de N0. D'autre part, au niveau de N3 les trames « RTS » venant de N0 et N2 tombent en collision pendant que N3 est en train de recevoir la trame « RTS ». Donc, N3 ne réussit pas à recevoir la trame de N2.

Après la simulation du scénario avec le simulateur OMNET++, nous avons constaté que chaque paquet envoyé est reçu par les nœuds destinataires N2 et N4donc nous avons trouvé les résultats prévu. Par contre les résultats de la simulation avec NS_2 qu' ils sont incorrects .En effet ,aucun paquet n'est reçu par les nœuds destinataires N2 et N4. Ceci est dû au phénomène suivant. Au début, chacun des nœuds N1 et N3 possède dans la file d'attente de la couche MAC un paquet à émettre. En écoutant le canal, ils ne trouvent aucun signal car le canal est encore vide. Par conséquent, chacun d'eux envoie un message « RTS » pour demander l'accès au canal. Les deux trames « RTS » tombent en collision au niveau de N2. Ainsi, ce dernier ne reçoit pas la trame « RTS » venant de N1. D'autre part, au niveau de N4 les trames « RTS » venant de N1 et N3 tombent en collision pendant que N4 est en train de recevoir la trame « RTS » . Donc, N4 ne réussit pas à recevoir la trame de N3 .

Avant de retransmettre son message « RTS », chacun des nœuds N1 et N3 attend un temps aléatoire. Si les temps d'attente sont égaux, il y aura une retransmission simultanée et on retombe de nouveau dans le même phénomène (collision entre les paquets « RTS ») jusqu'à la fin du trafic. D'après les résultats de simulation, aucun paquet n'est reçu par son destinataire : très probablement, il y a synchronisation dans l'envoi des messages « RTS «

et par conséquent aucun paquet « RTS » n'est reçu par les nœud N2 et N4. Il en résulte qu'aucun paquet « CTS » n'est reçu par les nœuds N1 et N3 et par la suite aucun paquet de donnée n'a quitté la couche MAC. En conclusion, ce scénario a montré que le simulateur NS-2 présente un point faible : le choix d'un nombre « SEED » identique pour les deux nœuds N1 et N3. En effet, normalement, chaque mobile possède un nombre « SEEDmobile ». Ce nombre, est créé à partir du nombre « SEEDsimulation » global qui est fixé dans la configuration de la simulation. À partir de chaque « SEEDmobile », un nombre aléatoire est généré. À partir de ce nombre, le mécanisme de « Backoff » calcule le temps de retransmission. Ainsi, les temps de retransmission des deux sources seront différents. S'ils ne sont pas différents, les nombres « SEEDmobile » des nœuds mobiles sont

égaux et la retransmission sera simultanée. Ceci constitue un problème qui a été retenu avec le simulateur NS-2 contrairement à OMNET++ qu'il a donné les résultats correcte.

3.Scalabilité, simulation et autres facteurs

3.1. Description

Dans cette section, nous allons considérer un scénario Une topologie de surface rectangulaire (1000 m, 500 m), contenant 50 noeuds dont 10 sources et 10 puits. Les sources envoient un flux CBR de l'instant t = 0.1s jusqu'à l'instant t = 30s. Le temps de simulation vaut 80s. Tous les noeuds sont mobiles. Le placement des noeuds ainsi que leurs chemins de mobilité sont aléatoires. Pour se faire, nous avons choisir le modèle de placement des noeuds « Random »

3.2. Résultats de simulation

3.2.1 Appliqué le scénario dans le simulateur NS-2 :

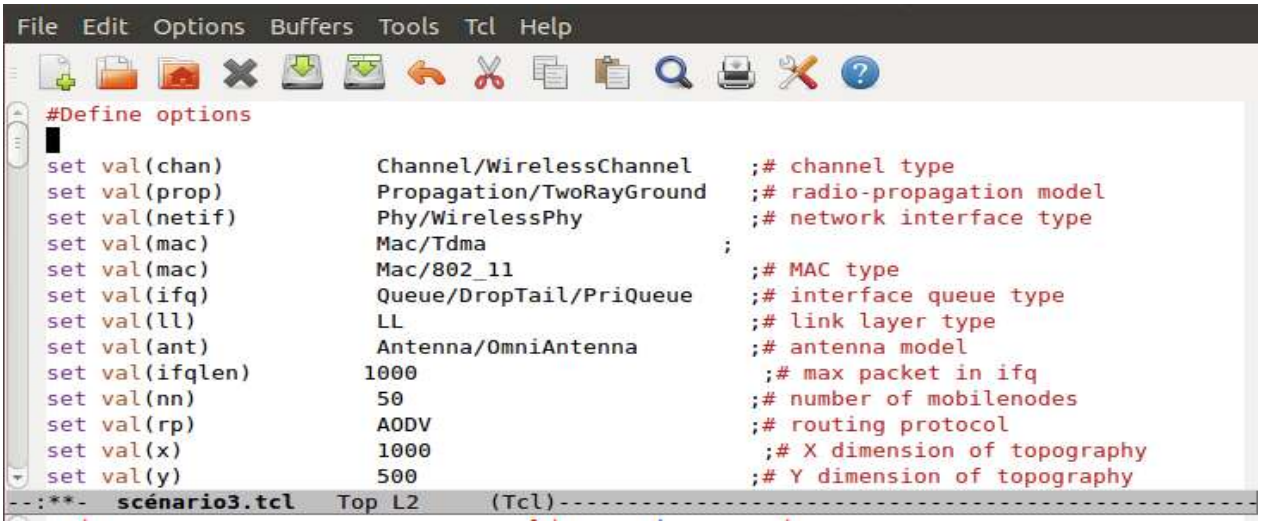

Figure 40- scénario3.tcl définie les variable pour les paramètres

3.2.2 Appliqué le scénario dans le simulateur OMNET++ :

```
*.host[1..10].udpApp[0].typename = "UDPBasicApp"
.host[1..10].udpApp[0].startTime =0.1s
*.host[1..10].udpApp[0].destPort = 100
*.host[1..10].udpApp[0].localPort = 100
*.host[1..10].udpApp[0].messageLength = 512B
*.host[1..10].udpApp[0].sendInterval = 0.01s  #0.02s  #0.005s
.host*.udpApp[0].stopTime = 30s
.host[1].udpApp[0].destAddresses = "host[11]"
.host[2].udpApp[0].destAddresses = "host[12]"
.host[3].udpApp[0].destAddresses = "host[13]"
.host[4].udpApp[0].destAddresses = "host[14]"
.host[5].udpApp[0].destAddresses = "host[15]"
.host[6].udpApp[0].destAddresses = "host[16]"
.host[7].udpApp[0].destAddresses = "host[17]"
.host[8].udpApp[0].destAddresses = "host[18]"
.host[9].udpApp[0].destAddresses = "host[19]"
.host[10].udpApp[0].destAddresses = "host[20]"
*.host[11..20].numUdpApps = 1
```

Figure 41-fichier.ini config paramater.

3.3. Etude comparative :

Dans cette comparaison, nous allons calcule les paquets perdus dans plusieurs cas, début 50 paquet/s et 100 paquet/s et 150 paquet/s et dans le cas avec mobilité et sans mobilité. Entre les deux simulateur « OMNET++ et NS-2 » .

La Formule pour calcule le paquet perdu :

$$\text{Taux paquet perdus} = \left[\frac{\text{Nombre des paquests envoyés} - \text{Nombre paquests reçus}}{\text{Nombre des paquets envoyés}}\right] * 100$$

Parmi les facteurs les plus importants et qui agissent sur le taux de paquets reçus, nous citons le débit du trafic, et la mobilité. Les figures 42_43 montrent la variation du taux de paquets perdu en fonction de ces facteurs.

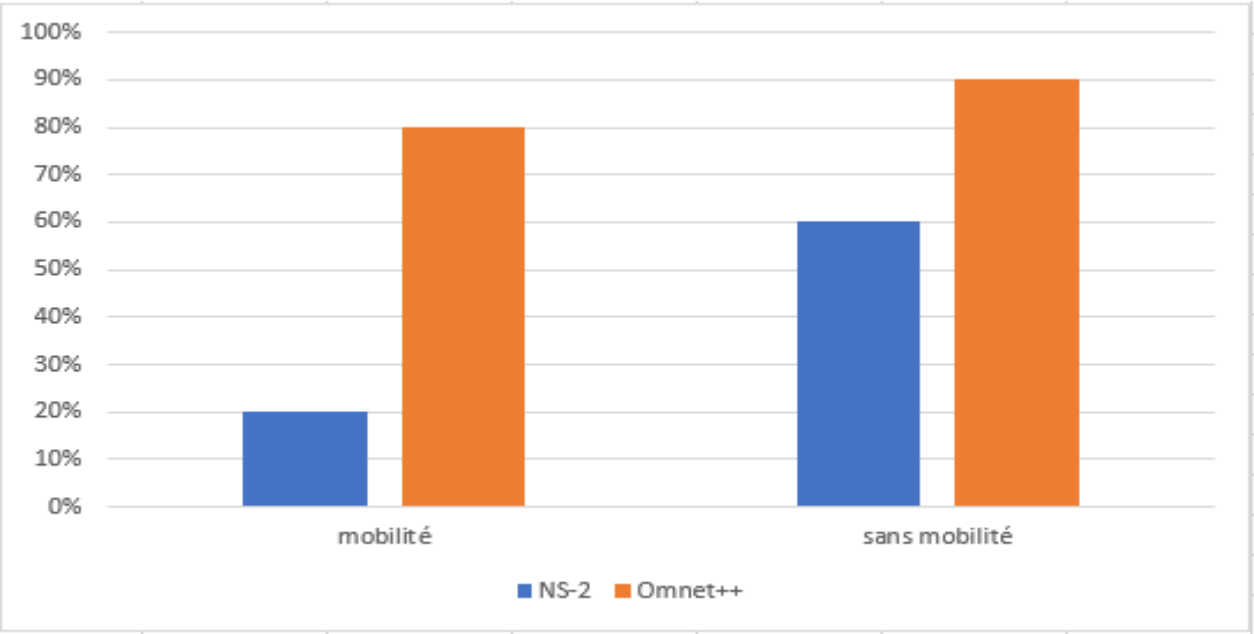

Figure 42-Taux paquet perdu en fonction de la mobilité

La figure 42 montre que la mobilité est un facteur important qui augment la performance du réseau en terme de paquets perdu. Lorsqu'on a un nœud qui se déplacent dans une surface, il y a plus de chances pour le nœud d'avoir le nœud puits comme voisin est le message délivrer sera reçu par le nœud destinataire, En effet, dans le cas sans mobilité, pour atteindre le nœud destinataire, un message peut effectuer plusieurs sauts, donc il peut être atténué ou même perdu.

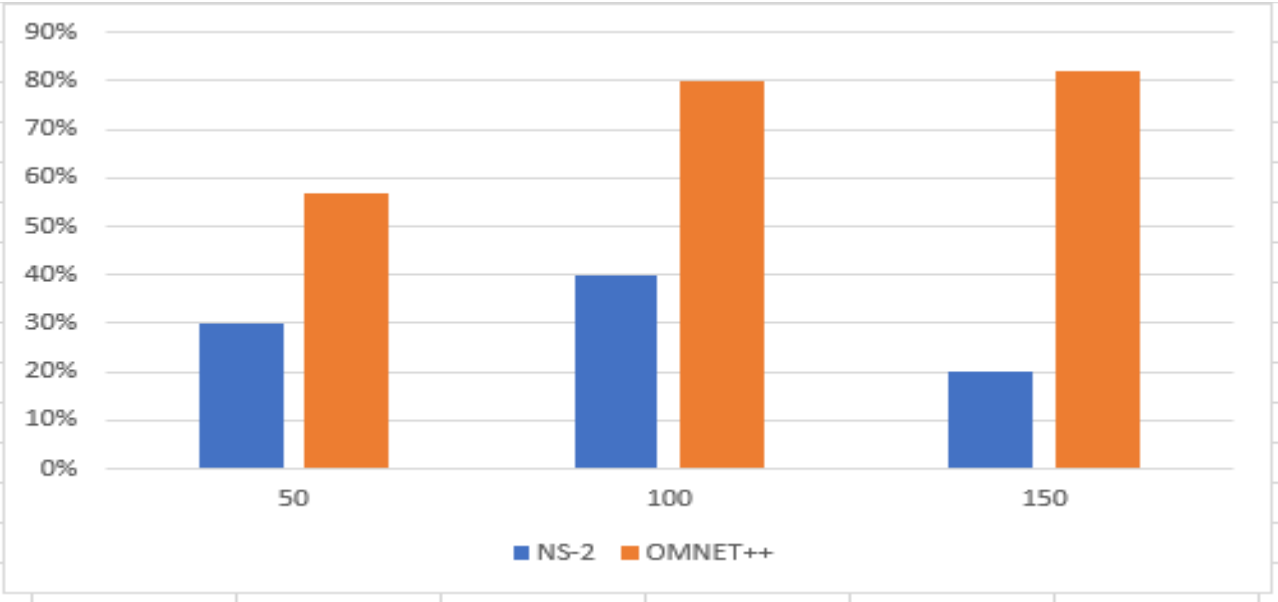

Figure 43-Taux paquets perdus en fonction du débit

Le taux de paquets reçus diminue lorsque le débit du trafic augmente car dans ce cas le nombre de collisions augmente. D'après la figure 43 nous remarquons que OMNET confirme ce résultat alors que NS-2 fournit des valeurs non logiques pour le débit 150 paquets/s. Ceci est dû à une injustesse dans l'implémentation du protocole de routage.

Conclusion

À travers ce dernier chapitre, nous avons vu que NS-2 présente un ensemble d'avantages par rapport à OMNET++. En revanche, nous avons montré que NS-2 possède les inconvénients : Les statistiques qu'il fournit ne sont pas rigoureuses.

- Il y a des cas ou il ne tient pas des facteurs bruit et d'interférence.
- La mauvaise gestion du nombre « *SEED* ».
- L'implémentation du protocole de routage AODV nécessite un perfectionnement.

Conclusion générale

L'objectif de ce mémoire de fin d'études était d'évaluer le simulateur Ns-2 en le comparant avec le simulateur OMNET++ et avec les résultats théoriques envisagés. Le choix de l'utilisation de OMNET++ est justifié par le fait qu'il est considéré comme étant populaire et a été testé par plusieurs chercheurs.

A travers notre comparaison, nous avons constaté que malgré les avantages fournis par OMNET++ tels que la facilité d'utilisation et la convivialité de ses interfaces, nous avons démontré qu'il présente plusieurs inconvénients liés à l'implémentation de certains de ses protocoles et à la simplification de quelques modèles, ce qui peut engendrer l'inexactitude de certains résultats. Ceci nous mène à mettre des réserves sur les outils de simulation d'une façon générale et par conséquent ‹sur tous les protocoles qui n'ont été validés que sur la base de simulations ‹et jamais par une implémentation réelle ou une preuve formelle. Les perspectives de ce travail seraient d'ajuster les incorrections qu'on a pu observer, à fin de faire converger les deux simulateurs vers des résultats proches de la réalité. Dans notre étude comparative nous nous somme basés sur des protocoles de réseaux sans fil Ad hoc, nous pourrons évaluer aussi les protocoles filaires supportés par NS2 et déduire ainsi les performances de ce simulateur pour les réseaux filaires .

Référence

[1] : Tayeb LEMLOUMA min projet thème « Le Routage dans les Réseaux Mobiles Ad Hoc » Proposé **par** Dr. Nadjib BADACHE.2000

[2] : FERROUDJ SONIA et HADJI THIZIRI. Mémoire thème « Etude comparative des deux protocoles de routage DSDV et DSR dans le cadre des réseaux ad hoc ». Université A/Mira de Bejaïa, faculté des sciences spécialité informatique 2016.

[3] : Tayeb LEMLOUMA Mémoire thème « Le Routage dans les Réseaux Mobiles Ad Hoc ».

[4] : BABOURI Karima, thème « Les simulateurs réseaux Technologie réseau » 2013/2014

[5] : *P. Anelli & E. Horlait : NS-2: Principes de conception et d'utilisation, Version 1.3*

[6] **:** Dr. Elbiaze , Rapport de stage Concepteur Développeur en Ingénierie Logicielle eXia 2006-2007 « OBS et NS-2 » , : 2006-2007

[7]: K. Fall and K. Varadhan, Eds., "The ns Manual, The VINT Project UC Berkeley, LBL, USC/ISI, and Xerox PARC," Apr. 2002, disponible au site: http://www.isi.edu/nsnam/ns/.

[8]: Priya Mahadevan, Adolfo Rodriguezn, David Becker, Amin Vahdat, « MobiNet: A Scalable Emulation Infrastructure for Ad Hoc and Wireless Networks », UC, San Diego, 2004.

[9] : Khaled BOUCHAKOUR, thème « Routage hiérarchique sur les réseaux de capteurs sans fil:Protocole KhLCH (K-hop Layered Clustering Hierarchy) ». Ecole Doctorale Sciences et Technologies de l'Information et de la Communication (STIC) Option : Système d'Information et de Connaissance (SIC). MEMOIRE Présenté pour l'obtention d'un diplôme de MAGISTER EN INFORMATIQUE.2012

[10] : KHADIDJA AYAD Mémoire MAGISTER Thème "Sécurité du routage dans les réseaux ad hoc mobile" Option : Informatique Répartie et Mobile Présenté par : 14 Novembre 2012

[11]: TRAN Alexandre & LAYOUNI Majid « Simulator Network 2 NS-2 » Session 2009

[12] : **Leila Imane Naiar , Théme<<Analyse Graphique pour la surveillance dans un réseaux de capteurs sans fils >>Juillet 2012**

[13] : **DKHIL Hassen ,Théme<<Implémenter le protocole Greedy Perimeter Stateless Routing de type VaNet & Adhoc sur OMNet++>>,01/05/2009**

[14] : Mme HELAILI Nabila.&&Mme MEKHNACHE Salima.théme <Simulation de Routage dans les Réseaux de Capteurs Sans Fils >, Promotion 2016-2017

Référence

[15] : https://wapiti.telecom-lille.fr 2018

[16]: **Witold Drytkiewics , Steffen Sroka , Vlado Handziski ,An-dreas Kopke , Hologer Karl ,"A Mobility Framework for OMNET++" , unoversity Berlin 2003**

[17]: Pierre-Jean Erard, Pontien Déguénon « Simulation par évènements discrets » presses polytechnique et universitaire Romandes, 1996 1iere édition .

Web graphies

[Site 1] http ://nsnam.isi.edu/nsnam/index.php/User_Information

[Site 2] Tutorial for the Network Simulator NS, http://www.isi.

[Site 3] http://www.memoireonline.com/12/07/738/m_effets-mobilite-protocoles-routage-reseaux-ad-hoc29.html

[Site4] http://igm.univ-mlv.fr/~dr/XPOSE2002/Sansfils/index.php?rubrique=Principe

[Site5] https://www.memoireonline.com/12/07/738/m_effets-mobilite-protocoles-routage-reseaux-ad-hoc28.html