

5.1.Introduction :

La simulation constitue actuellement l'outil le plus pratique pour évaluer le comportement d'un système complexe dont la formalisation à l'aide de méthodes analytiques est difficile. Pour tester les performances d'un réseau mobile, on a souvent recours à la simulation. En effet il serait trop coûteux, voire impossible, de mettre en place un réseau à des fins de test pour certains critères. Par exemple, tester des applications sur des réseaux de grande envergure n'est possible en réalité que si l'on dispose de moyens matériels importants. Cependant, dans le cadre d'une simulation, il suffit de changer les paramètres de simulation correspondant à la taille du réseau.

Plusieurs simulateurs pour les réseaux sans fil ont été proposés ces dernières années, parmi lesquels NS-2 [32], GloMoSim [33], JiST/SWANS [34], GTSNetS [35], OMNeT++ [36], Opnet [37], etc. Ces simulateurs offrent tous un environnement avarice de programmation pour l'implémentation et l'évaluation des performances des protocoles de communication.

Nous présentons, dans ce chapitre, l'environnement de simulation conçu, les outils techniques nécessaire pour mettre en œuvre le système, la description des tests effectués et les résultats obtenus.

5.2.But de la simulation :

La simulation est l'expérimentation sur un modèle. C'est une procédure de recherche scientifique qui consiste à réaliser une reproduction artificielle (modèle) du phénomène que l'on désire étudier, à observer le comportement de cette reproduction lorsque l'on fait varier expérimentalement les actions que l'on peut exercer sur celle-ci, et à en induire ce qui se passerait dans la réalité sous l'influence d'actions analogues.

5.2.1.Avantages de la simulation :

La simulation permet en principe d'étudier les modèles de réseau avec n'importe quel niveau de détail. On peut facilement modifier les conditions de fonctionnement du réseau et comparer les mesures d'intérêt d'un scénario à l'autre. Pour un utilisateur la simulation a plus de crédibilité que les modèles analytiques car elle est plus proche du système réel et nécessite moins de simplifications et quasiment pas d'hypothèse spécifique.

On procède généralement à la simulation pour évaluer un nouveau système ou bien pour des raisons de coûts d'évaluation par des mesures réelles. La simulation permet en plus de visualiser les résultats sous forme de graphes faciles à analyser et à interpréter. Si on néglige l'étape de l'évaluation et on passe directement à l'implémentation réelle et la mise en œuvre du système les coûts peuvent être sévères en cas de non-conformité du système avec les exigences temporelles requises pour les applications.

5.3.Présentation de l'outil OPNET :

5.3.1.introduction :

OPNET (Optimum Network Performance) est un outil de simulation de réseaux très puissant et très complet. Basé sur une interface graphique intuitive permet de dessiner et d'étudier des réseaux de communications, des équipements, des protocoles et des applications avec facilité et évolutivité. Modeler est utilisé par les entreprises technologiques les plus performantes pour accélérer leurs procédés de recherches et développements.

L'approche orientée objet associée à des éditeurs graphiques intégrés de Modeler simplifie la composition des réseaux et des équipements. Ceci permet de réaliser facilement une correspondance entre votre système d'informations et votre modèle. Modeler est basé sur une série d'éditeurs hiérarchisés qui parallélisent la structure du réseau réel, des équipements et des protocoles.

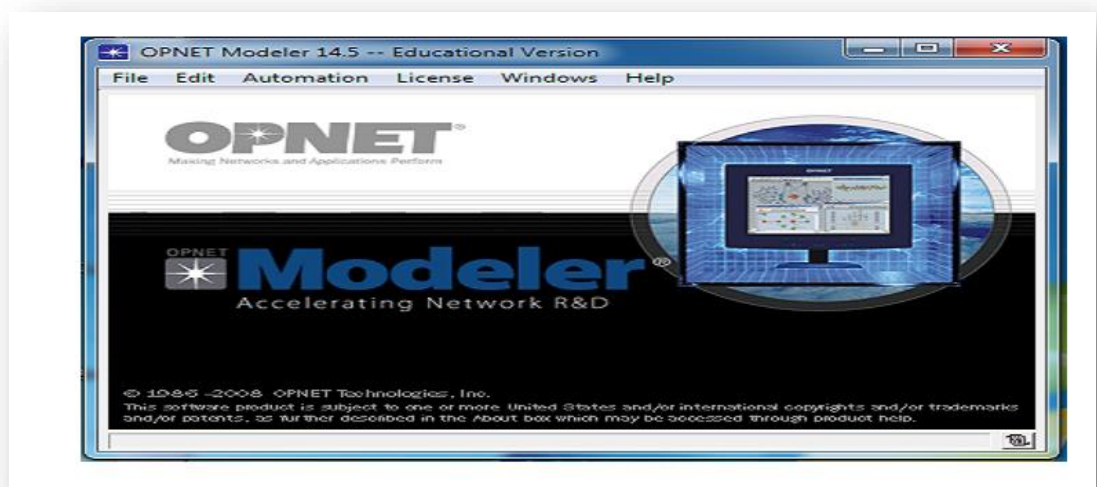


Figure 5.1: Interface d'Opnet

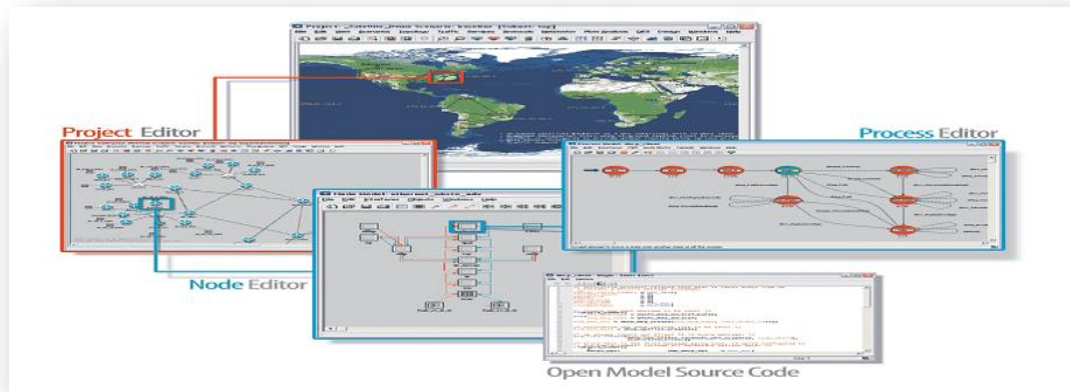


Figure 5.2 Liens hiérarchiques entre les différentes interfaces

5.3.2.simulateur Opnet et les reseaux ad hoc :

Reconnu par la communauté scientifique et plus utiliser par la communauté Ad-hoc.

- ✓ Open source (disponibilité du code).
- ✓ Flexibilité : étude des cas difficiles à reproduire dans la réalité.
- ✓ Faible coût des expérimentations.
- ✓ Reproductibilité des réseaux : tout le monde peut contribuer au code.
- ✓ Exhaustif : de nombreux protocoles et mécanismes mis en œuvre dans Opnet.

5.3.3.La simulation sur le OPNET Modeler :

Une simulation OPNET Modeler c'est :

- ✓ Des modèles des nœuds.
- ✓ Des modèles des processus.
- ✓ Des descripteurs des statistiques.
- ✓ Plusieurs simulations exécutées avec des valeurs de Radom Number Seed (graine) différentes.

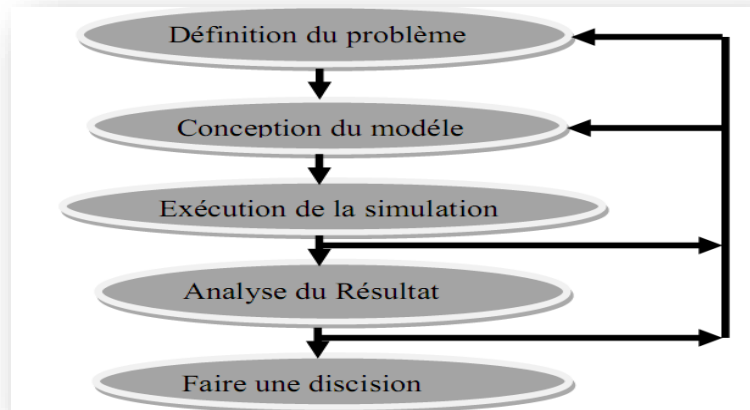


Figure 5.3: Cycle de modélisation et de simulation

5.3.4.Simulation sous OPNET :

OPNET fournit en standard une liste impressionnante d'implémentations de routeurs, de stations de travail, des switches.

On Peut donc construire une simulation de réseaux en utilisant principalement deux méthodes :

- ✓ En utilisant les nœuds pré-programmés fournit par la librairie de OPNET.
- ✓ En commençant tout depuis le début et en définissant soi-même un modèle de lien, des process models décrivant des routeurs et des hôtes.

Cette méthode est bien évidemment plus complexe que la première, et nécessite de bonnes connaissances en matière de programmation et de réseaux. Néanmoins elle est indispensable dans le cas où l'on désire expérimenter un algorithme tout nouveau.

5.3.4.1.Les interfaces principales :

Parmi les nombreuses interfaces que propose OPNET au démarrage les interfaces suivantes :

5.3.4.1.1. Rédacteur de model de réseaux :

Permet de représenter la topologie d'un réseau de communication constitué de nœuds et de liens par l'intermédiaire de boîtes de dialogues (palettes et glisser/poser). Cette interface tient en compte du contexte géographique.

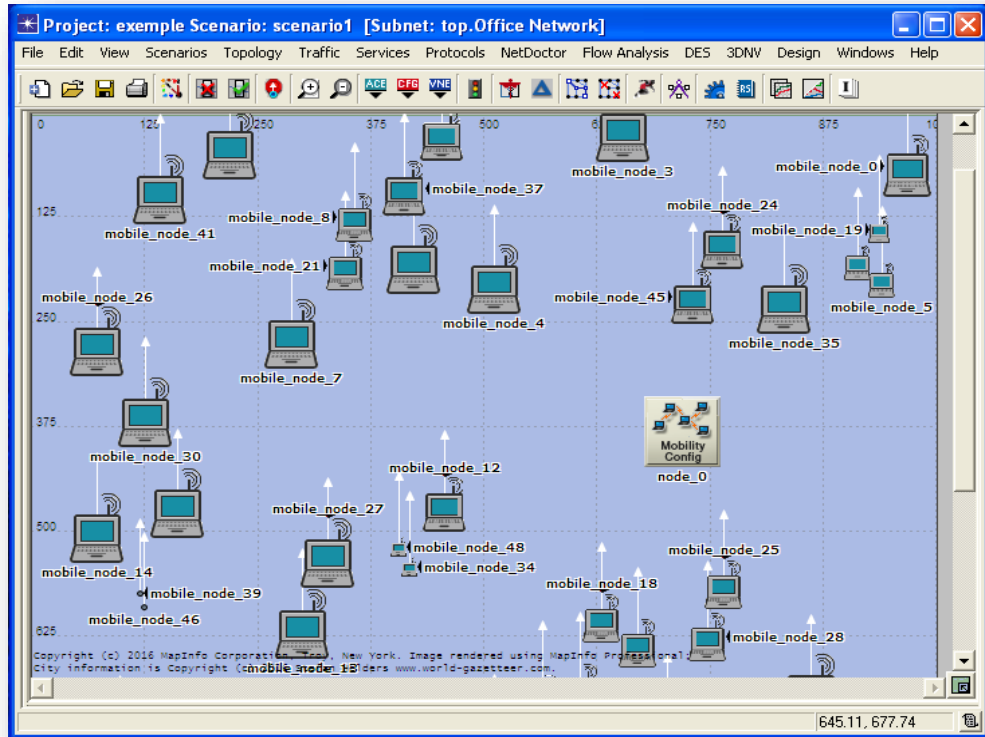


Figure 5.4: l'interface Rédacteur de model réseaux

5.3.4.1.2. Rédacteur de projet :

Interface principale du logiciel .elle permet d'implanter des modèle issus des bibliothèques OPNET ainsi que des modèle créés par l'utilisateur ,de configurer puis lancer des simulations, et de visualiser les résultats des simulations.

Les principales fonctions de cette interface sont disponibles sous formes d'icônes :

5.3.4.1.3. Rédacteur de nœud :

Le Rédacteur de Nœud vous permet de définir la constitution des nœuds (routeurs, stations, de travail, hub,...) le modèle est défini à l'aide de blocs appelés modules. Certains modules sont non programmables : il s'agit principalement des transmetteurs et des récepteurs, de flux de paquets qui permettent de faire transiter un

paquet d'une couche à une autre à l'intérieur d'une même machine. Cette organisation permet d'avoir une vision claire de la pile de protocoles implémentée dans un nœud, et de connaître rapidement leurs interactions. Les statistic wires constituent le second type de lien permettant une communication entre modules :

Comme leur nom l'indique, ils permettent de faire remonter des informations de statistique d'un module à l'autre, comme par exemple la taille et le délai des queues des transmetteurs.

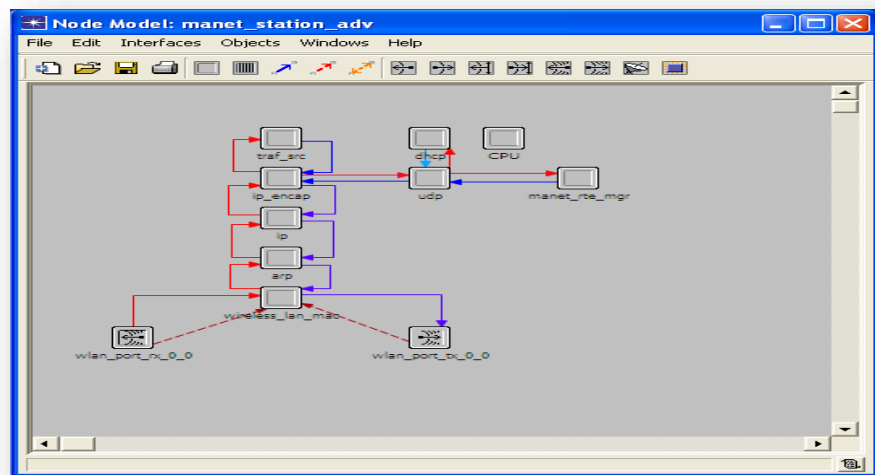


Figure 5.5: l'interface Rédacteur de nœud

5.3.4.1.4. Rédacteur de modèle de processus :

Un processus, ou process, est représenté comme une machine à états. Chaque état peut être dans l'état ouvert (couleur verte) ou fermé (couleur rouge).

L'entrée dans un état ouvert est immédiatement et automatiquement suivie de la sortie de cet état. Par contre on ne sort d'un état fermé que lorsqu'il advient un événement provoque le passage d'un état à un autre. Les opérations exécutées dans chaque état ou pour une transition sont décrites dans c incorporé ou c ++ blocs codes.

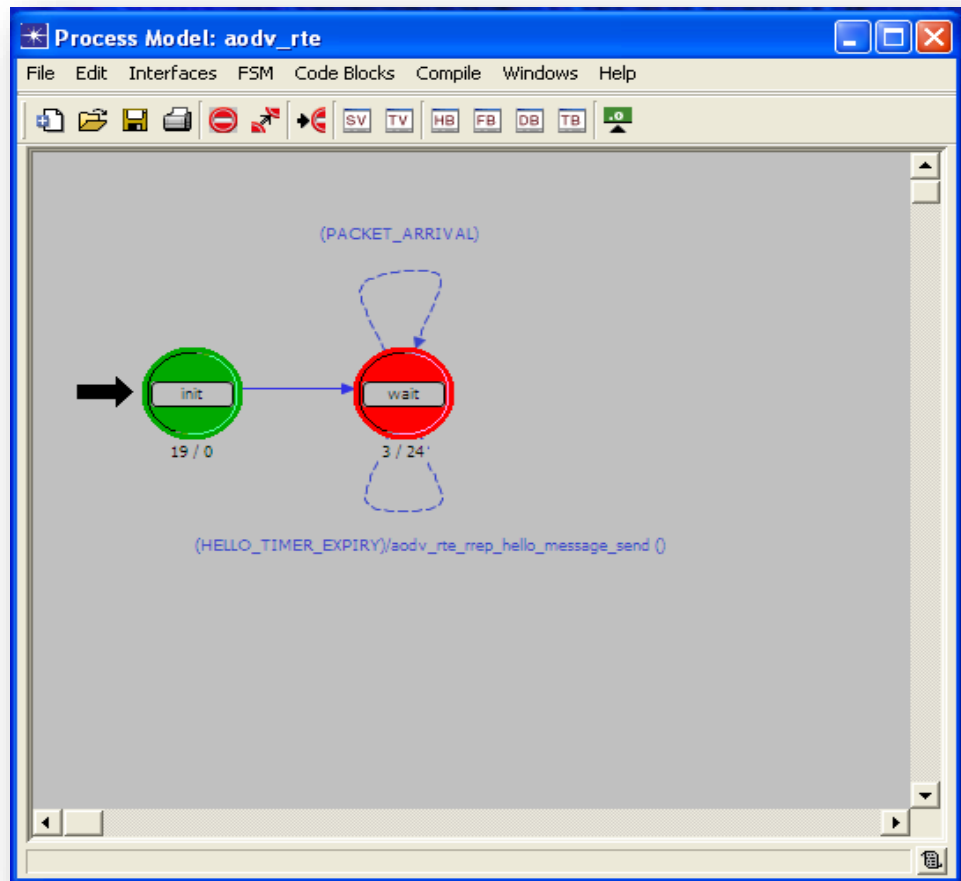


Figure 5.6: l'interface Rédacteur de model de processus

5.4.Simulation d'AODV dans OPNET :

La solution OPNET est capable d'analyser et d'optimiser la topologie du réseau (i.e. analyse des configurations routeurs, switches, firewalls) ainsi que de faire du capacity planning à partir de matrices de flux capturées sur le réseau réel. Modeler dispose donc d'un environnement de développement complet qui va permettre de modifier/enrichir les protocoles déjà fournis en standard mais également d'ajouter de nouveaux protocoles de communications non fournis par défaut.

Le code source des tous les modèles de protocoles est disponible dans Modeler. Il est donc tout à fait possible de modifier/ajouter des protocoles de communication. L'architecture de modélisation dans Modeler comporte trois niveaux :

- **Modèle de réseau :** C'est le niveau le plus élevé de la hiérarchie d'OPNET. Il permet de définir la topologie du réseau en y installant des routeurs, des hôtes,

des équipements tels que des Switchs, reliés entre eux par des liens. Chaque entité de communication (appelée nœud) est entièrement configurable et est définie par son modèle. Le modèle de réseau que nous avons conçu pour simuler et évaluer notre proposition se compose de 50 nœuds placé aléatoirement dans une zone de simulation de 1000M sur 1000M. Le modèle de mobilité que nous avons choisi est le modèle RWP (RandomWayPoint), avec la vitesse des mobiles variées entre 25 et 100 m/s, temps de simulation est de 1600 seconde.

- **Modèle de nœud :** Il va permettre de définir l'architecture interne d'un nœud particulier. Par exemple, la Figure 5.5 présente le modèle interne d'un utilisateur Manet station. On retrouve une architecture qui suit la pile OSI. Si l'on désire accéder au code source des protocoles, il faut alors atteindre le troisième niveau de modélisation, appelé le "Process Model" dans lequel le fonctionnement d'un protocole particulier est réalisé sous forme de machine à états finis. Par exemple, la Figure 5.6 présente des machines à états qui codent le protocole AODV :
 - ✓ **Etat Init :** Pour initialiser l'état et les paramètres (valeurs des variables) du nœud dans un état initial (initialiser toutes les valeurs des statistiques à récolter au cours de la simulation à zéro, création de table de routage vide, table de voisinage vide, etc.).
 - ✓ **Etat wait :** Cet état contient deux transitions vers lui-même.

5.5.Complément pratique sur modeler :

Dans cette section on va présenter la liste des figures qui constitue le protocole AODV sur opnet :

aodv.h : définit les structures des tables AODV.

aodv_pkt_support.h : définit les formats de paquet AODV.

aodv_support.h/ex.c : définit les fonctions de code AODV.

5.6.Métrique de performance mesurée :

Dans le but de tester notre protocole AODV_RREP, la simulation est faite par rapport aux métriques suivantes :

- Le temps de la découverte de route dans le réseau.
- La métrique de délai.

- Le rapport de la retransmission des paquets.
- Total de paquets perdu.
- Le nombre de saut par route.

Les paramètres du protocole AODV qu'on a choisi sont comme présente la figure suivante :

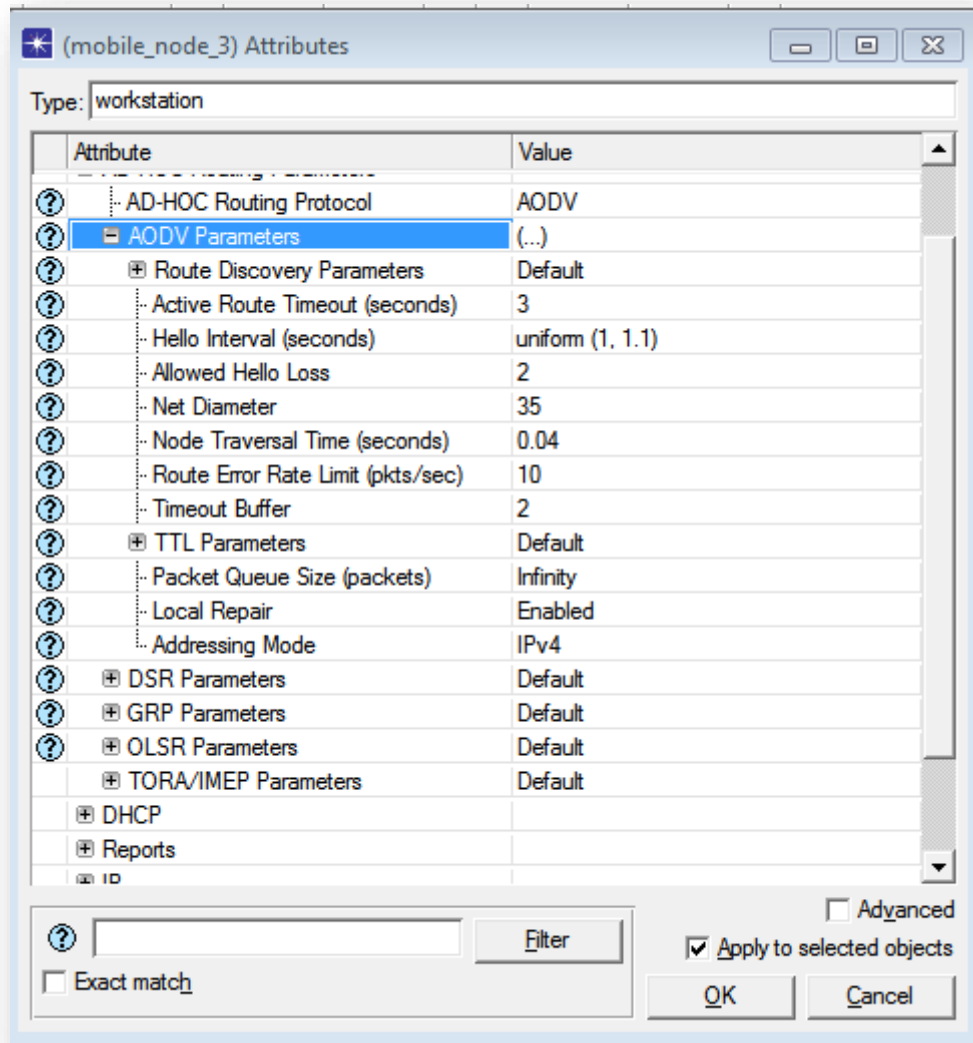


Figure 5.7 : Les paramètres d'AODV

Les paramètres de WLAN choisi sont :

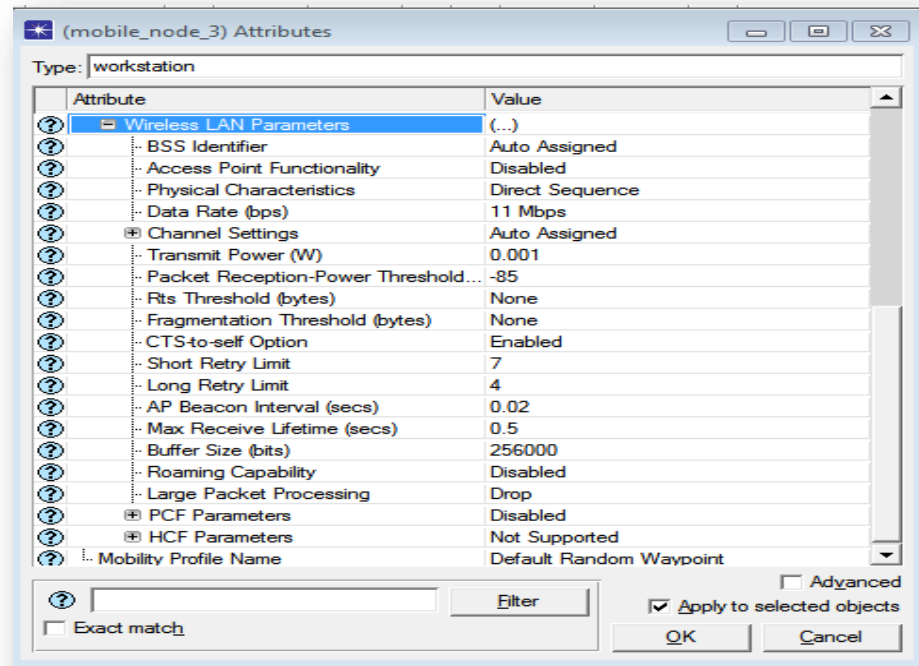


Figure 5.8 : Les paramètres de WLAN

Les paramètres du trafic généré sont :

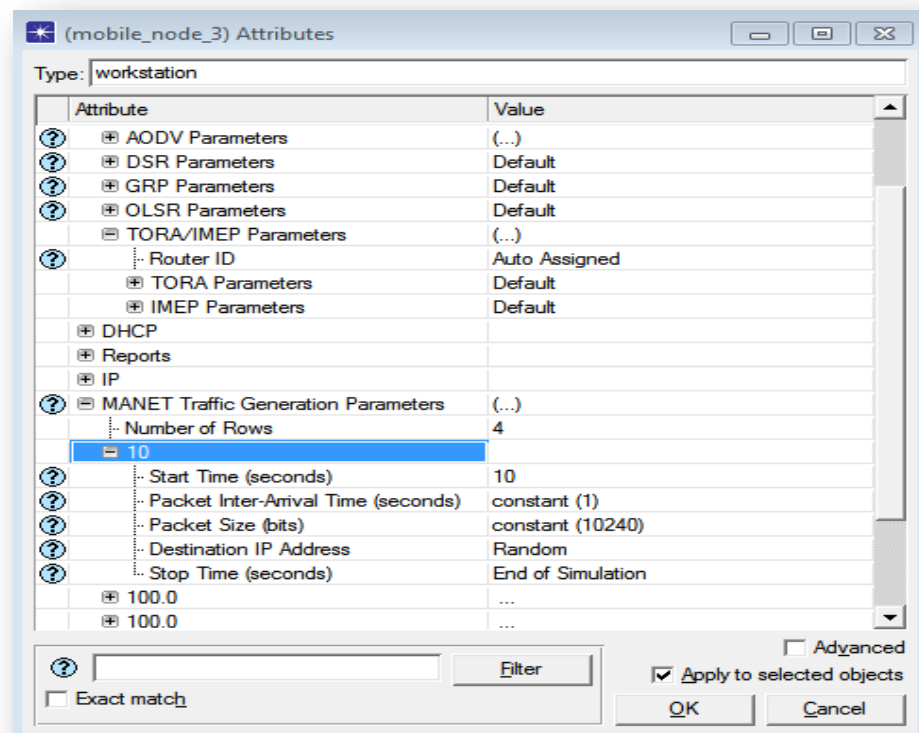


Figure 5.9 : Les paramètres du trafic générer

5.7. Discussion des résultats de simulation :

Dans ce qui suit, nous allons présenter les résultats de simulation de notre proposition (le protocole AODV_RREP) on comparaison avec le protocole AODV standard ,La simulation à donner les résultats suivants :

5.7.1. Temps de la découverte de route dans le réseau :

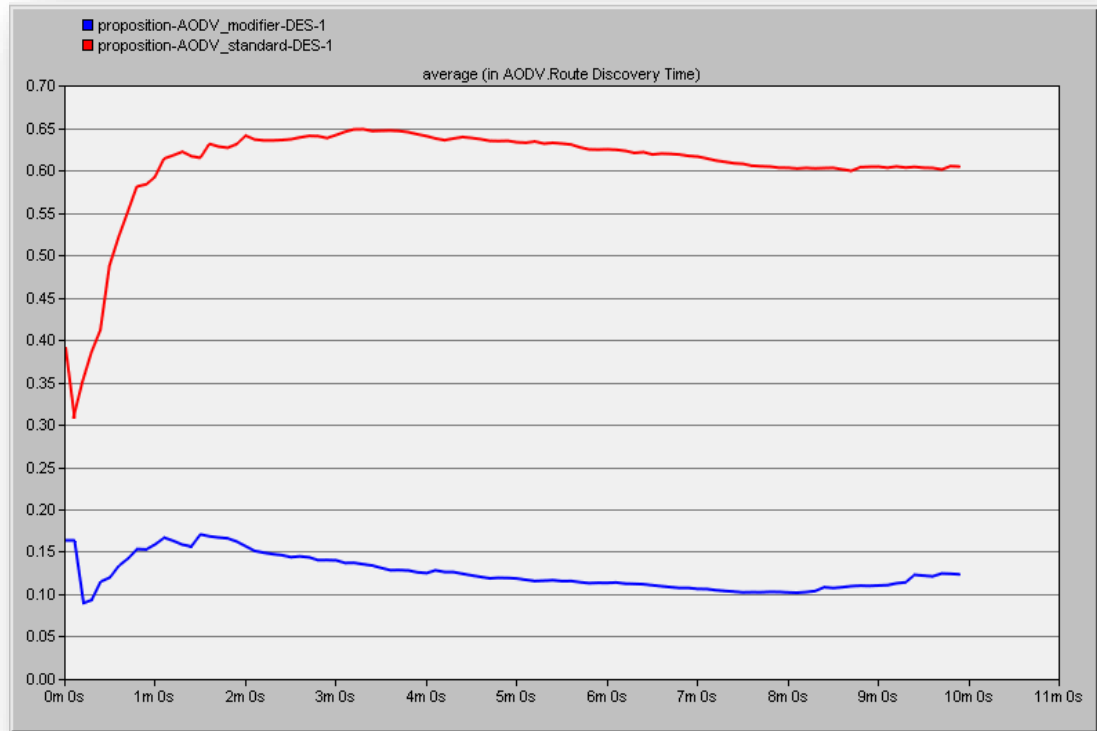


Figure 5.10: Temps de la découverte de route dans le réseau

❖ Discussion de résultat :

La figure 5.10 illustre le temps de la découverte de route dans le réseau. Nous pouvons observer, dans ce graphe, que le temps de la découverte de route dans AODV_Standard est plus grand que dans AODV_Modifier, Puisque dans notre proposition les itinéraires choisis sont plus fiables et à chaque transmission on n'a pas besoin de la découverte de route à chaque itinéraire puisque on a toujours les chemins enregistrés dans la table de routage secondaire.

5.7.2. La métrique de délai (Delay) :

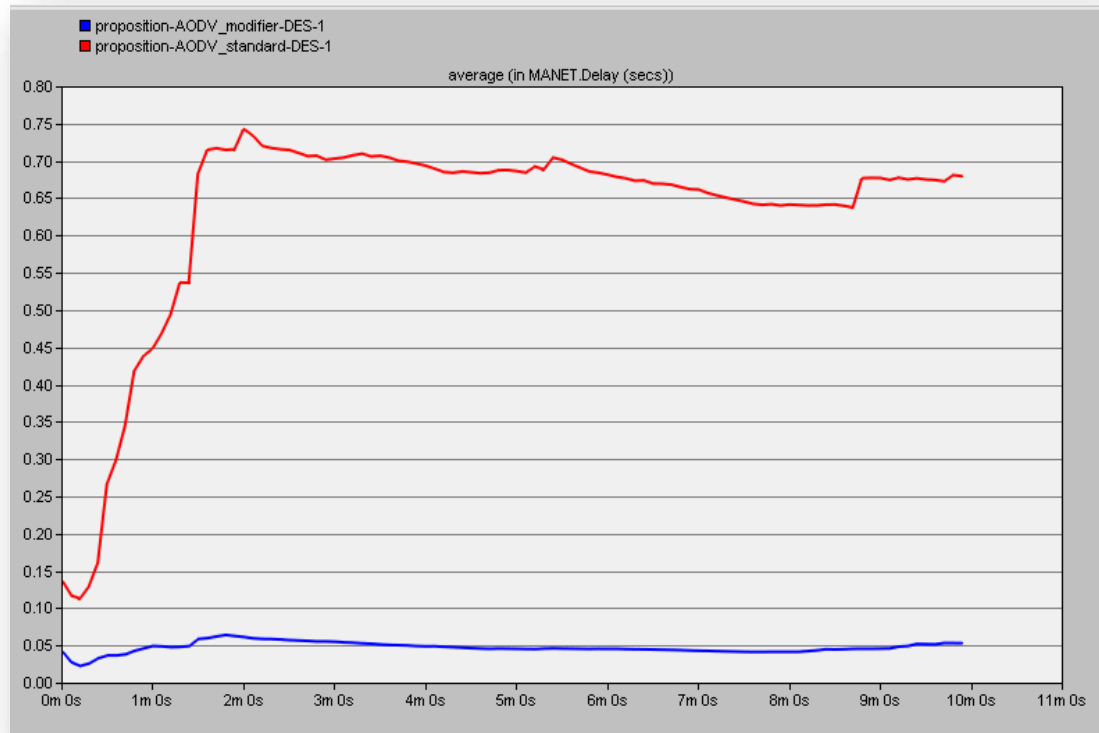


Figure 5.11 : La métrique de délai (Delay)

❖ Discussion de résultat :

La figure 5.11 montre que le délai de bout en bout dans AODV_Modifié est plus faible par rapport au protocole AODV_Standard. Cela est dû à la sélection des liens selon le critère de délai, ainsi le nombre d'interruptions de chemins qui est très faibles.

5.7.3. Le rapport de la retransmission des paquets :

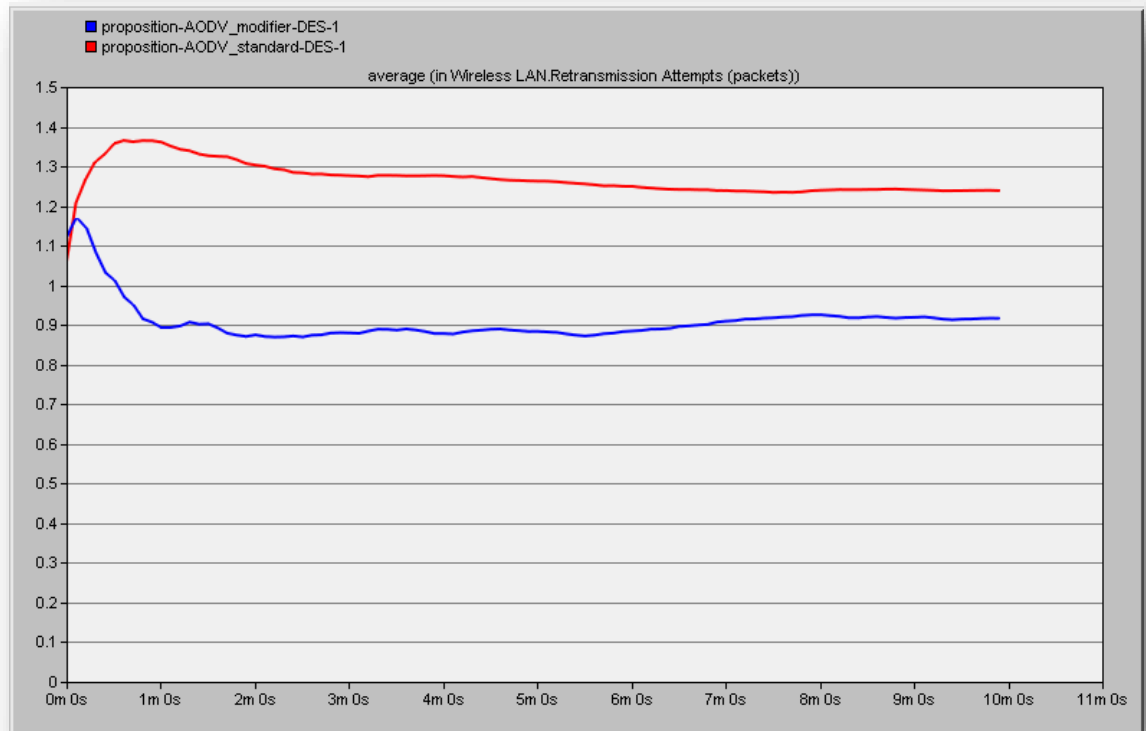
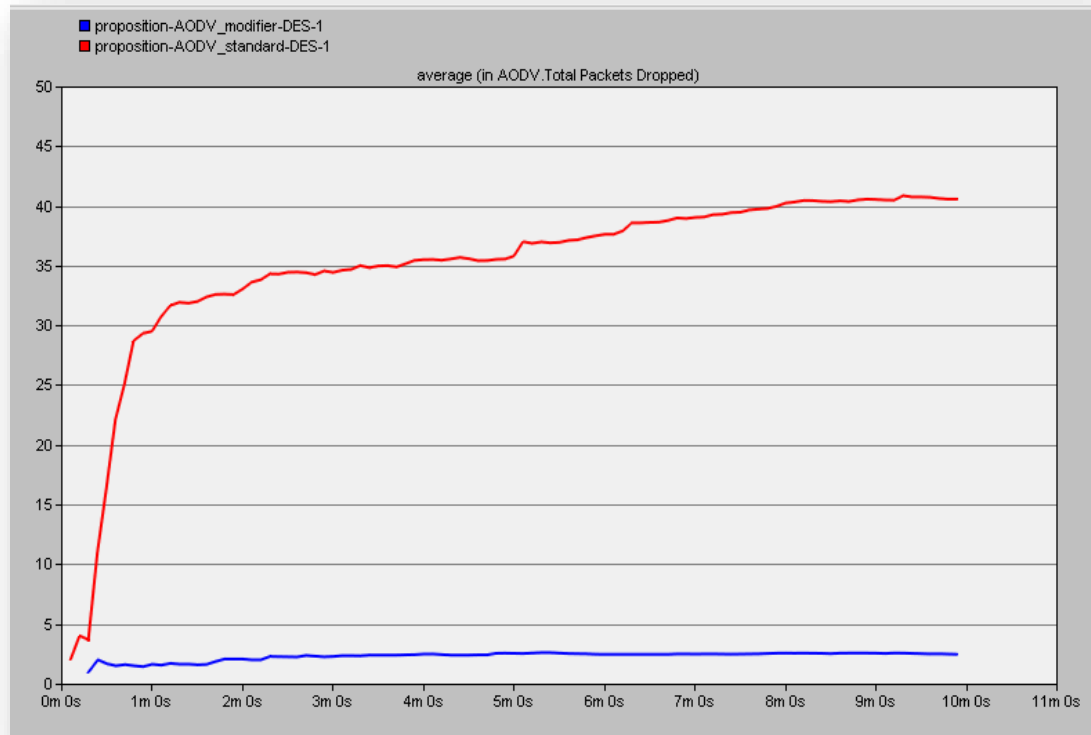


Figure 5.12 : Le rapport de la retransmission des paquets

❖ Discussion de résultat

Nous pouvons observer, dans ce graphe (figure 5.12), clairement que le rapport de la retransmission des paquets dans AODV_Standard est plus grand que dans AODV_Modifier. On peut expliquer ces résultats par le fait que dans le protocole AODV_Modifier, il y aura toujours des chemins secondaire pour chaque transmission et cela affecte sur les paquets c'est-à-dire chaque nœud envoie un message et dans le cas qu'il y aura une interruption il a un nouveau chemin pour continuer sa transmission et il évite a chaque coupure la retransmission des paquets.

5.7.4.Total de paquets perdu :**Figure 5.13 : Total de paquets perdu****❖ Discussion de résultat :**

La figure 5.13 illustre le total de paquets perdus dans le réseau. Nous pouvons observer, dans ce graphe, clairement, que le rapport de paquets perdus dans AODV_Standard est plus grand que celui obtenu dans AODV_Modifier, puisque les itinéraires sont plus stables et durables par rapport à AODV.

5.7.5. Le nombre de saut par route :

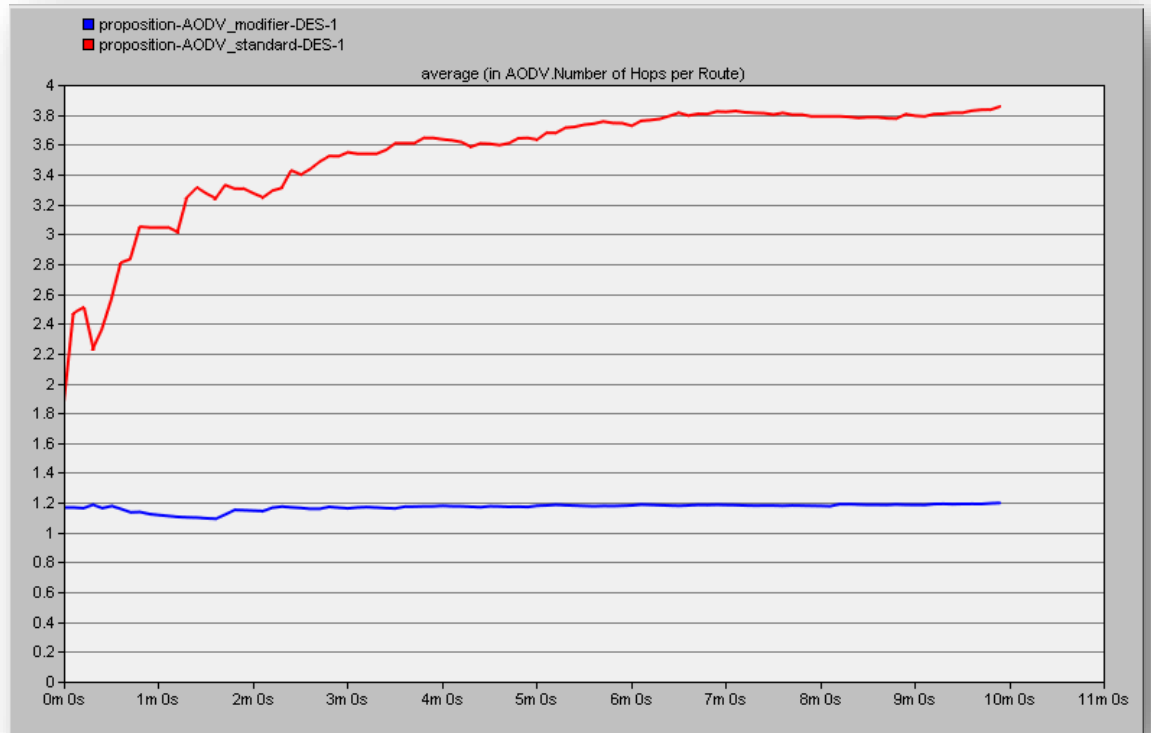


Figure 5.14 : Le nombre de saut par route

❖ Discussion de résultat

La figure 5.14 illustre le nombre de saut par route dans le réseau. Nous pouvons observer, dans ce graphe, clairement que le nombre de saut par route dans AODV_Standard est plus grand que dans AODV_Modifier. Puisque les itinéraires dans le protocole AODV_Standard sont moins stables. Par contre dans notre protocole les itinéraires sont fiables et à chaque transmission chaque nœuds a plusieurs chemin pour atteindre sa destination donc il évite à chaque la découverte et le rétablissement des nouveaux chemins.

5.8.conclusion

On a présenté dans ce chapitre les différents concepts de simulation, ainsi que le simulateur et les paramètres de simulation qu'on a choisie dans ce travail. Après la comparaison des résultats de performances de notre proposition avec celles du protocole AODV. Nous pouvons conclure que notre proposition améliore considérablement le protocole de routage AODV.