2008

UFR Ingénieurs 2000

Vivien Boistuaud
Julien Herr

PROJET DE MODELISATION DE RESEAUX AD-HOC AVEC NS-2

Ce document a été réalisé par V. Boistuaud et J. Herr dans le cadre des travaux pratiques de Modélisation des Réseaux coordonnés par Hakim Badis, au sein de l'UFR Ingénieurs 2000 de l'Université de Marne-la-vallée.

Table des matières

Introduction 3
1. Présentation générale du Projet 4
   1. Outils mis en œuvre 4
   2. Mode opératoire 4
   3. Algorithmes de routage Ad hoc étudiés 6
      1. L'algorithme DSR (Dynamic Source Routing) 6
      2. L'algorithme AODV (Ad hoc On Demand Distance Vector) 6
      3. L'algorithme OLSR (Optimized Link State Routing Protocol) 7
2. Etude d'un réseau Ad hoc statique 8
   1. Script de simulation 8
   2. Présentation des résultats de la simulation 8
      1. Bande passante moyenne pour 1 à 25 CBR 8
      2. Probabilité de perte pour 1 à 25 CBR 9
      3. Délai de bout en bout pour 1 à 25 CBR 10
   3. Conclusions sur les algorithmes pour un réseau ad hoc statique 11
3. Etude d'un réseau Ad hoc dynamique 12
   1. Script de simulation 12
   2. Présentation des résultats de la simulation 12
      1. Bande passante moyenne pour 1 à 25 CBR 12
      2. Probabilité de perte pour 1 à 25 CBR 13
      3. Délai de bout en bout pour 1 à 25 CBR 14
   3. Conclusions sur les algorithmes pour un réseau ad hoc statique 15
4. Synthèse sur les algorithmes de routage Ad hoc 16
Conclusion 19

Introduction

Dans le cadre de nos travaux pratiques de modélisation des réseaux au sein de l'UFR Ingénieurs 2000 de l'Université de Paris-Est-Marne-la-vallée, nous avons simulé plusieurs types de réseaux sans fils ad-hoc afin d'évaluer les performances de différents algorithmes de routage ad-hoc.

En utilisant le logiciel de simulation NS2 (Network Simulator 2), dont les simulations se programment en TCL/TK, nous avons évalué par la simulation les principes de fonctionnement et plus particulièrement les performances des algorithmes DSR[1], AODV[2] et OLSR[3].

L'objectif de ce rapport est de présenter notre démarche, les scripts que nous avons mis en place pour évaluer les différents algorithmes dans diverses situations, et les analyses que nous avons tirées de ces expériences.

Dans un premier temps, ce rapport contient une brève description des outils utilisés, du mode opératoire suivi, ainsi que des algorithmes de routage Ad hoc étudiés. Dans un second temps, il présente les expérimentations liées au fonctionnement statique pour un positionnement aléatoire des nœuds. Enfin, il présente les expérimentations liées au fonctionnement avec des nœuds mobiles.

[1] DSR : Dynamic Source Routing
[2] AODV : Ad hoc On Demand Distance Vector
[3] OLSR : Optimized Link State Routing Protocol

1. Présentation générale du Projet

Avant tout chose, nous allons vous présenter les outils et le mode opératoire qui nous a permit d'étudier les protocoles Ad Hoc.

1. Outils mis en œuvre

Afin de simuler les protocoles Ad Hoc, nous utilisons un simulateur de réseau nommé NS-2 (Network Simulator). NS supporte partiellement les protocoles TCP, de routage et multicast sur réseau filaire ou sans fil.

La simulation a exécuter par NS lui ai fournis par l'intermédiaire d'un fichier TCL (Tool Command Language) qui est un langage de script.

Il est à noter que même si lors des séances de travaux pratiques nous étions sous un environnement Linux, il est possible d'utiliser NS2 sous Windows via Cygwin. Pour cela, il suffit de suivre le tutoriel disponible sur le wiki officiel de NS2[4].

Enfin, comme nous l'avons précisé précédemment, les protocoles à étudier sont DSR, AODV et OLSR. Les protocoles DSR et AODV sont directement géré par NS et sont donc très simple à mettre en place. Ce n'est malheureusement pas le cas pour le protocole OLSR. Nous avons cependant trouvé une procédure par ajouter ce protocole à l'outil de simulation. En effet, il existe un patch des sources de NS[5] qui ajoute la connaissance d'OLSR. Une fois les sources patchées, il ne reste plus qu'à recompiler NS.

Autant la théorie semble simple, autant la pratique l'est moins. En effet, il n'existe pas de patch pour notre version de NS (version 2.32) et bien que cela devrait tout de même fonctionner, nous n'avons pas réussis à patcher les sources et donc intégrer OLSR.

2. Mode opératoire

L'étude que nous allons faire se divise en deux fonctionnements distincts. Le premier fonctionnement est une simulation avec des émetteurs et des récepteurs

[4] http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin
[5] http://masimum.dif.um.es/?Software:UM-OLSR:Installation

fixes, placés aléatoirement dans la zone de couverture de l'étude. Le second fonctionnement est, quant à lui, une simulation avec des émetteurs et des récepteurs mobiles, placés aléatoirement dans la zone de couverture de l'étude et se déplaçant aléatoirement dans cette zone.

Afin de garantir des mesures les plus précises possibles, nous allons exécuter six fois chacune des simulations et faire la moyenne des mesures. Il était demandé de faire la moyenne de 30 simulations différentes, mais lorsque nous nous sommes aperçu qu'après plus de 24 heures de simulation, les simulations exécutées ne représentées moins de 30% des tests à réaliser, nous avons préférer revoir le nombre de simulation, estimant quand les chiffres seraient suffisant.

Chacune des mesures aura des caractéristiques de topologie identique :

- La région des mesures sera toujours de 1 000m^2 (1 000m sur 1 000m) ;
- Le nombre de nœuds sera toujours de 50 ;
- La taille de la queue sera toujours de 50 et est de type FIFO ;
- Le réseau sera toujours sans fil (canal Wireless, interface physique, antenne omnidirectionnelle) ;
- Les paquets sont de taille 512 octets et son envoyé par le CBR toutes les 0.1 seconde pendant 30 secondes ;
- La position des nœuds est aléatoire.

Seul le nombre d'émetteurs variera afin d'étudier le comportement des protocoles suivant le nombre d'émetteurs.

Il est à noter que pour les mesures dynamiques, chacun des nœuds va choisir un point de destination et se déplacer vers celui-ci avec une vitesse sélectionnée uniformément entre 0m/s et 10m/s. Une fois le nœud arrivé à sa destination, ce dernier effectue une pause de 5 secondes.

3. Algorithmes de routage Ad hoc étudiés

Un réseau Ad hoc est un réseau créé par une réunion de mobiles ne disposant pas d'infrastructure préexistante. Dans le cadre de notre projet, nous allons étudier trois protocoles Ad hoc : DSR, AODV et OLSR.

Avant de décrire rapidement ces protocoles, il est important de savoir qu'il existe deux types de protocole : les protocoles proactifs et les protocoles réactifs. Les premiers échanges des paquets de contrôle et ainsi mettent continuellement les tables de routage à jour. A l'opposé, les protocoles réactifs ne possèdent pas de table de routage à jour ; ces dernières le sont uniquement lors d'une requête.

1. L'algorithme DSR (Dynamic Source Routing)

DSR est un protocole réactif utilisé dans les réseaux Ad hoc de petit diamètre (entre 5 et 10 sauts) et de vitesse modérée. Afin de trouver une route, cette dernière est déterminée par la source vers la destination. A chaque nœud intermédiaire traversé, l'adresse de ce dernier est inscrite dans le paquet.

Le protocole est divisé en deux phases : la découverte de route et la maintenance de route. Pour la découverte, l'émetteur envoie d'un paquet RouteRequest en inondant le réseau. A la réception d'un de ces paquets, le nœud réagis différemment suivant les cas. Si le nœud est le destinataire, il retourne un paquet RouteReply contenant le chemin. Sinon, soit le nœud a déjà vu le paquet (son adresse est présente dans le paquet) et dans ce cas, il jette le paquet ; soit c'est la première fois que le paquet arrive par le nœud et dans ce cas le nœud inscrit son adresse dans la route (Route Record) et fait suivre le paquet.

Afin de maintenir les routes, le concept utilisé par DSR est simple : chaque nœud est responsable de transférer aux nœuds suivants. Si plusieurs échecs successifs arrivent lors de la transmission d'un paquet, le nœud envoie d'un message RouteError à l'émetteur qui supprime la route et en utilise soit une autre soit en cherche une nouvelle.

2. L'algorithme AODV (Ad hoc On Demand Distance Vector)

AODV est un protocole réactif. Contrairement à DSR, AODV est basé sur la construction de table de routage. En effet, chaque nœud possède sa propre table de routage contenant pour chaque destination, le prochain nœud à contacter.

La découverte d'une route se fait par inondation par l'émetteur d'un paquet RREQ. A là réception d'un de ces paquets, si le nœud connait le chemin pour accéder à la source, il envoie une réponse RREP à l'émetteur qui arrête d'inonder le réseau. S'il le nœud ne connait pas le chemin, il transmet le paquet à ses voisins tout en mémorisant le nœud précédent ayant fait la requête.

En cas de cassure du lien, un message RERR est envoyé à l'émetteur qui décide ou non de recommencer l'envoie du paquet suivant le taux d'utilisation de la route.

3. L'algorithme OLSR (Optimized Link State Routing Protocol)

OLSR est un protocole proactif ayant les mêmes principes de base qu'OSPF, c'est-à-dire qu'il gère un état des liens du réseau. Le concept principal est celui des MPR (Multi-Poin Relays) qui sont des nœuds choisis pour expédier les messages de diffusion pendant le processus d'initialisation.

Afin de maintenir une table de routage à jour, chaque nœud envoie, toutes les deux secondes, un message de type Hello qui contient la liste de tous les voisins directs de l'émetteur ainsi que l'état et le type des liens avec les voisins. Le message Hello permet aussi de spécifier le MPR choisis par l'expéditeur. Il faut noter que les messages Hello ne sont destinés qu'aux nœuds voisins de l'expéditeur et ne doivent donc jamais être routé par un MPR (sauf dans le cas de l'élection des MPR).

2. Etude d'un réseau Ad hoc statique

Nous allons tout d'abord nous intéresser aux protocoles dans une topologie statique, quand chacun des nœuds reste fixe tout au long de la simulation.

1. Script de simulation

Le script que nous avons fait ressemble à ceux que nous avions déjà fait lors des Tps. Il y a cependant quelques particularités à noter :

Tout d'abord, le placement aléatoire des nœuds se fait par l'intermédiaire d'une variable RandomVariable/Uniform

```
…
set size_ [new RandomVariable/Uniform]; # Variable Random uniforme
$size_ set min_ 1;                                  # Valeur minimum
$size_ set max_ 999;                             # Valeur maximum
…
```

Ensuite, nous avons été dans l'obligation de changer le type de la file d'attente pour le protocole DSR : CMUPriQueue au lieu de la file FIFO. En effet, l'utilisation de la file FIFO avec le protocole DSR provoquait une erreur sur notre système.

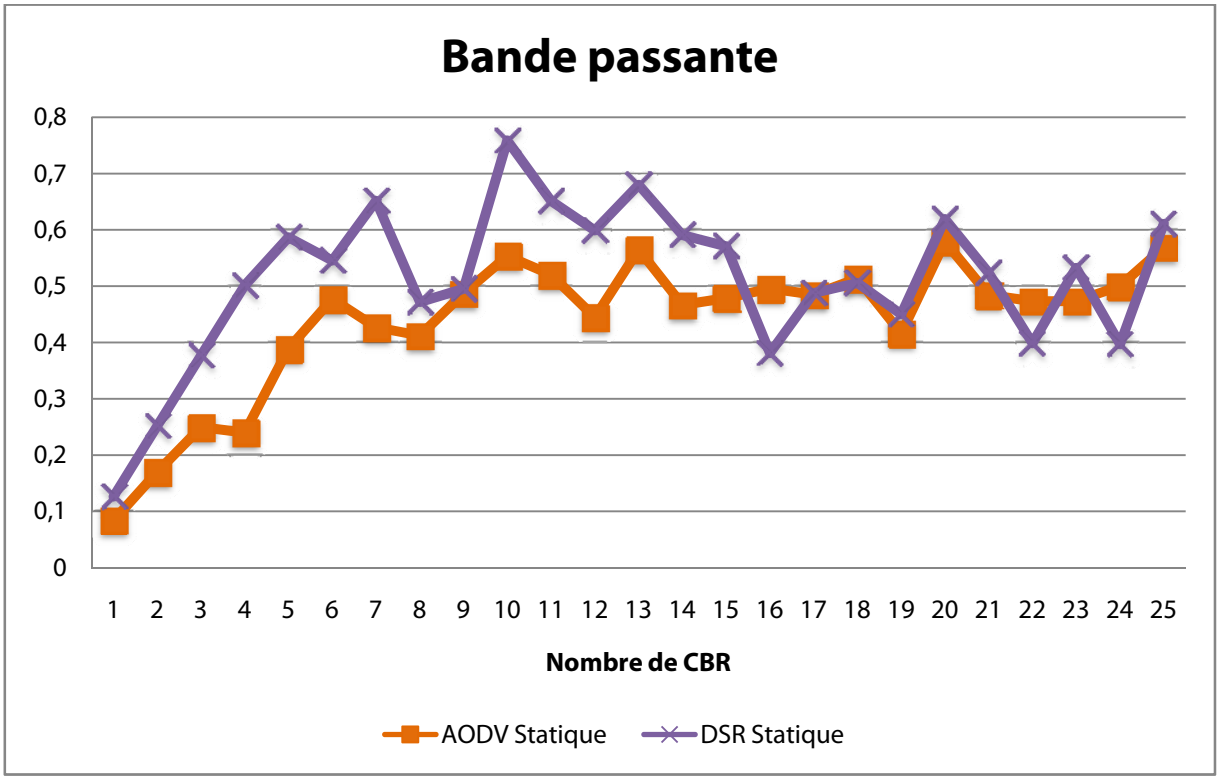
Enfin, nous avons décidé de ne pas choisir aléatoirement les nœuds émetteurs. En effet, la position de l'ensemble des nœuds étant déjà aléatoire, nous avons considéré que les nœuds émetteurs étaient par conséquences aléatoires.=

2. Présentation des résultats de la simulation

Les scripts nous ont permis de relever la bande passante, les pertes de paquets ainsi que le délai de bout en bout pour chacun de protocole. Chacune des mesures est, comme dit précédemment, la moyenne de 6 mesures. On cherche à étudier l'évolution suivant le nombre de nœuds émetteurs (ici de 1 à 25).

1. Bande passante moyenne pour 1 à 25 CBR

Ci-après, le graphique de la bande passante moyenne pour l'ensemble des algorithmes, pour 1 à 25 CBRs différents.

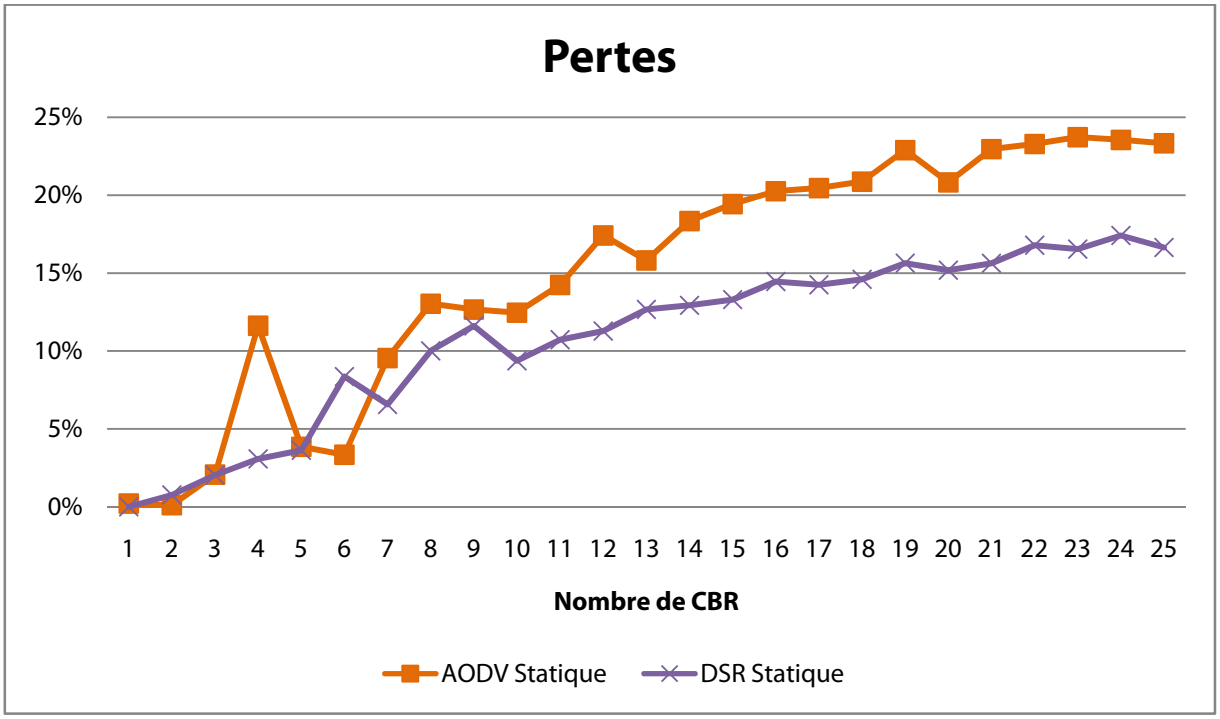


A la vu du graphique, nous pouvons constater deux choses. La première est qu'à partir d'un certain seuil (ici, au bout de 15 émetteurs), le nombre d'émetteurs ne permet plus d'augmenter la bande passante. Cela peut se comprendre par le fait que le réseau est saturé et qu'il n'est plus possible d'envoyer des données.

La seconde constatation que l'on peut faire, est que le protocole DSR est plus adapté qu'AODV lorsque le nombre d'émetteurs est faible. Cependant, le seuil de saturation semble être le même pour les deux protocoles.

2. Probabilité de perte pour 1 à 25 CBR

Le second graphique permet de voir le pourcentage de paquets perdus pour l'ensemble des algorithmes, pour 1 à 25 CBRs.

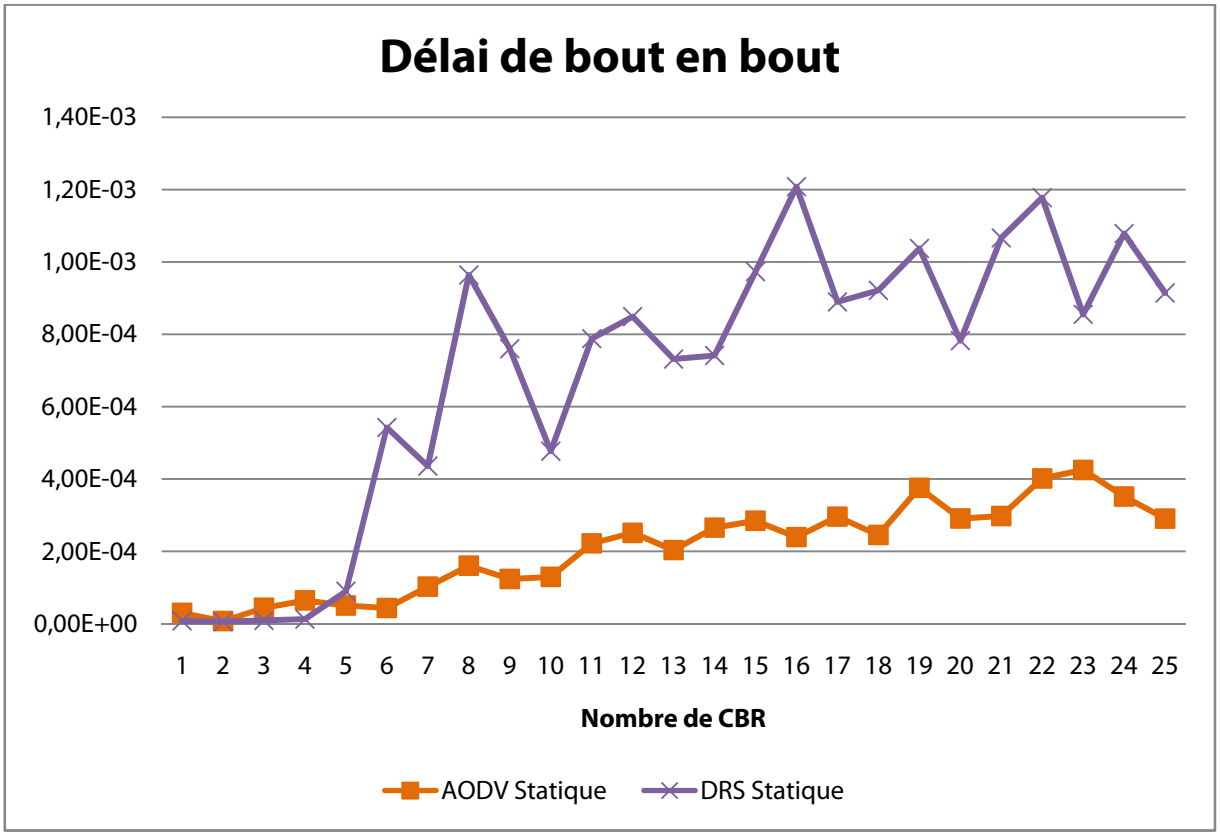


Comme précédemment, à la vue du graphique généré, nous pouvons constater deux choses. Tout d'abord, que plus il y a d'émetteurs et plus les pertes sont importantes.

Mais nous remarquons aussi que le protocole DSR semble plus adapté dès que le nombre d'émetteurs augment puisque les pertes sont moins importantes pour ce protocole dès 10 émetteurs.

3. Délai de bout en bout pour 1 à 25 CBR

Le dernier graphique permet de voir du délai de bout en bout pour l'ensemble des algorithmes, pour 1 à 25 CBRs.



Là encore, nous pouvons constater que pour un nombre limité d'émetteurs, le délai de bout en bout est très faible (moins de 5 émetteurs) et crois au fur et à mesure de l'augmentation du nombre d'émetteurs.

On remarque également que passé ce seuil, le délai de transmission de bout en bout avec le protocole DSR augmente bien plus que celui du protocole ADV. Nous pouvons donc en déduire que le protocole AODV est plus optimisé pour l'acheminement de paquets que DSR.

3. Conclusions sur les algorithmes pour un réseau ad hoc statique

Pour conclure la partie Ad Hoc statique, on constate que le protocole DSR est plus adapté qu'AODV pour des topologies présentant peu d'émetteurs. AODV est, quant à lui, plus adapté pour les topologies présentant un nombre plus importants d'émetteurs, ceci malgré de taux important de perte de paquets.

3. Etude d'un réseau Ad hoc dynamique

La seconde série de mesure a été d'étudier les protocoles en mode dynamique, c'est-à-dire quand l'ensemble des nœuds se déplacent au cours de la simulation.

1. Script de simulation

Les scripts de simulation sont identiques à ceux utilisé pour les mesures statiques. Cependant, il est important de faire un point sur notre façon de déterminer aléatoirement la mobilité des nœuds.

En effet, pour générer le déplacement des nœuds dans le modèle dynamique, nous n'avons pas utilisé le Random Waypoint mais une procédure dans le script Tcl :

```
…
proc setNewDest {node currentX currentY} { … }
…
```

Cette procédure permet de prévoir la date d'arrivée au point de destination du nœud en fonction de sa vitesse et de la distance. La position de destination du nœud est déterminée de la même manière que le placement du nœud sur la grille, puisqu'il s'agit ici d'un point sur la grille.

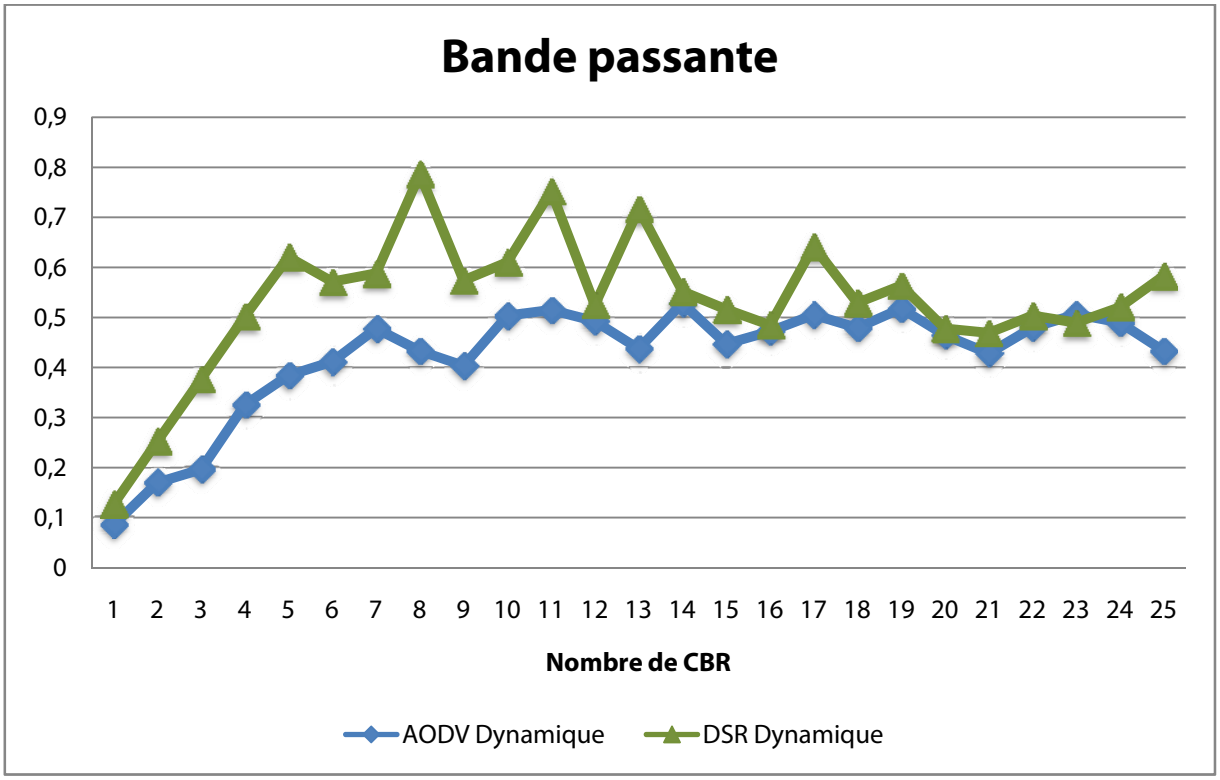
Etant donné que nous avons la position courante du nœud et la position à atteindre, ainsi que sa vitesse de déplacement. Par un simple calcul, la date d'arrivée est facilement calculable. A cette date d'arrivée suivie de 5 secondes de pause, nous demandons à ns2 d'effectuer un nouvel appel à cette procédure afin de définir un nouveau déplacement du nœud passé en paramètre. Nous pouvons ainsi effectuer autant de déplacement qu'il est possible pendant le temps de simulation défini dans le script Tcl.

2. Présentation des résultats de la simulation

Ci-après, nous trouvons les graphiques résumant les mesures de la simulation Ad Hoc dynamique.

1. Bande passante moyenne pour 1 à 25 CBR

Tout d'abord, nous voyons le graphique représentant la bande passante.

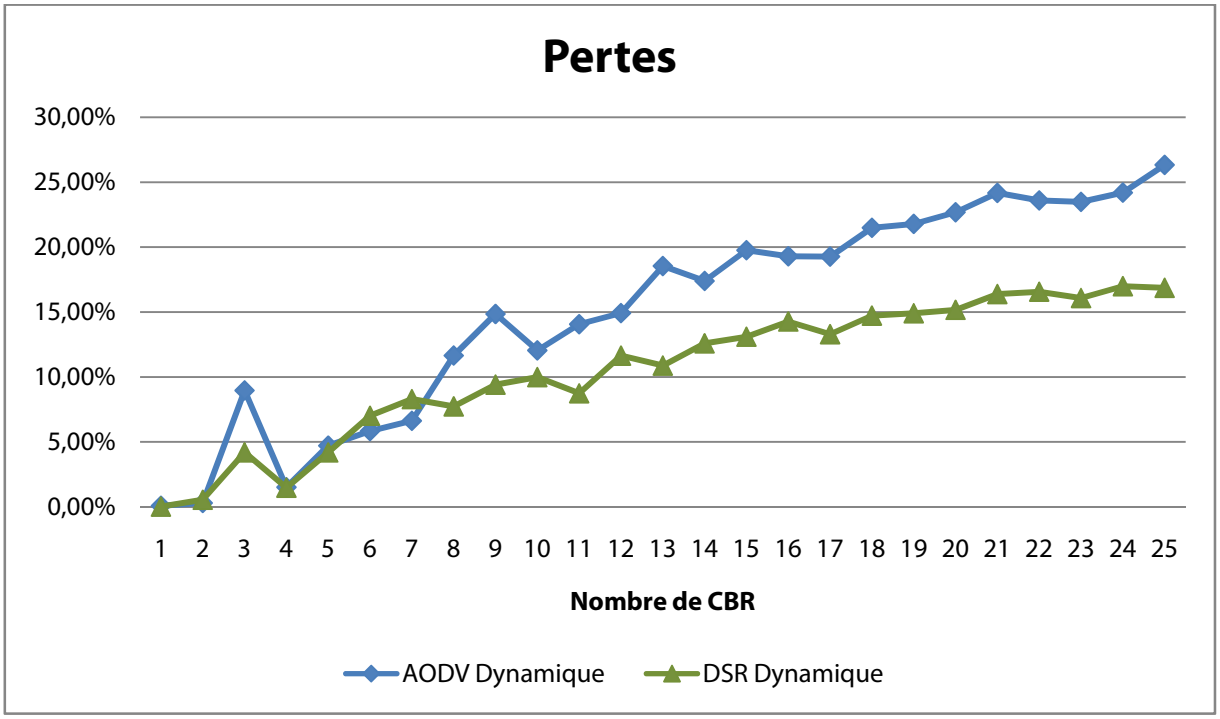


Tout comme lors de mesures statiques, la bande passante augmente au fur et à mesure de l'augmentation du nombre d'émetteur jusqu'à atteindre le seuil de saturation du réseau. Là encore, le protocole DSR semble plus adapté lorsque le nombre d'émetteurs est faible.

Il est étonnant de noter que les débits entre un réseau statique et dynamique sont très proches, voir identiques.

2. Probabilité de perte pour 1 à 25 CBR

Nous voyons ensuite le graphique montrant le taux de perte pour l'ensemble des algorithmes, pour 1 à 25 CBRs.

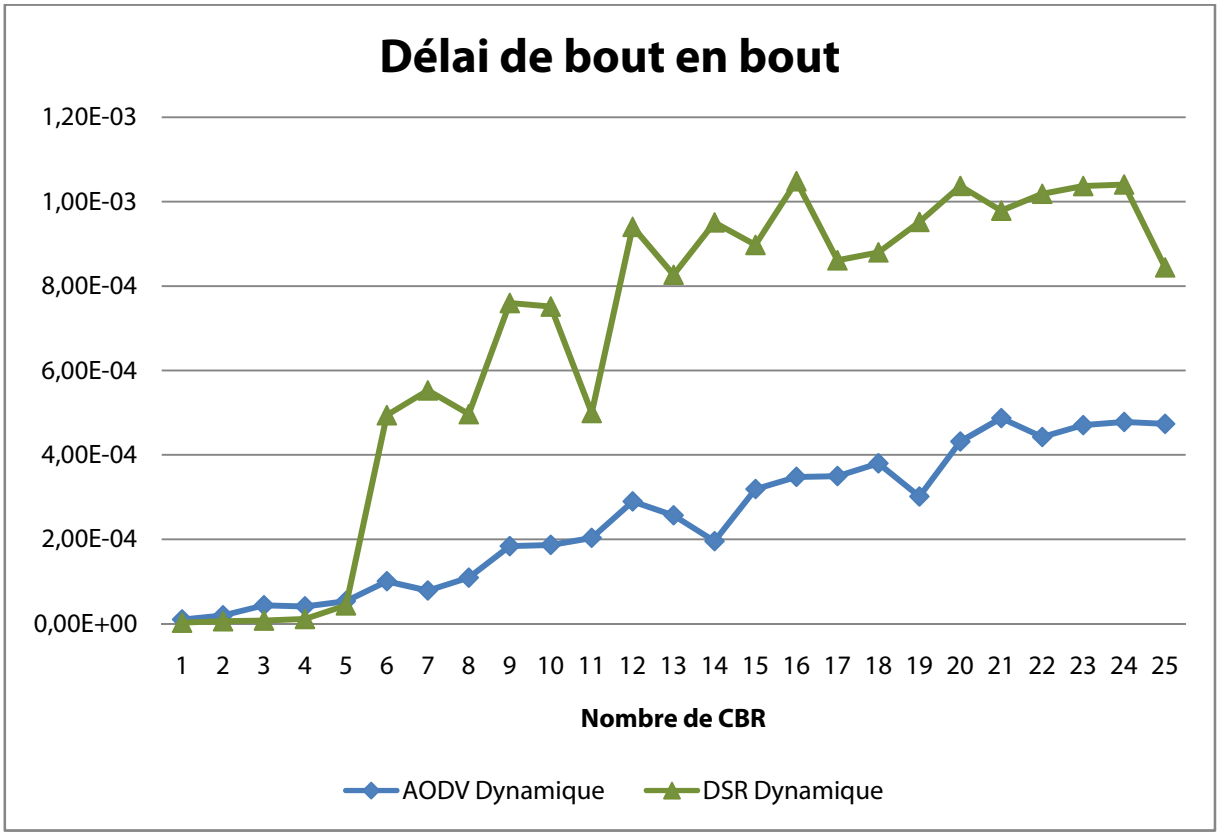


Il est à nouveau étonnant de remarquer que les commentaires que nous avions faits pour les mesures statiques sont identiques pour les mesures dynamiques.

En effet, on constante que les pertes augmentent avec le nombre d'émetteurs. On remarque également que pour un nombre d'émetteurs faible (jusqu'à 8 émetteurs), les protocoles sont semblables et que pour un nombre d'émetteurs élevé, le protocole DSR est plus performant.

3. Délai de bout en bout pour 1 à 25 CBR

Enfin, nous voyons le graphique montrant le délai de bout en bout pour l'ensemble des algorithmes, pour 1 à 25 CBRs.



La remarque que nous avons faite pour les deux mesures précédentes est également applicable ici. En effet, la différence entre les mesures statiques et dynamiques est faible et les commentaires que l'on peut faire sont identiques.

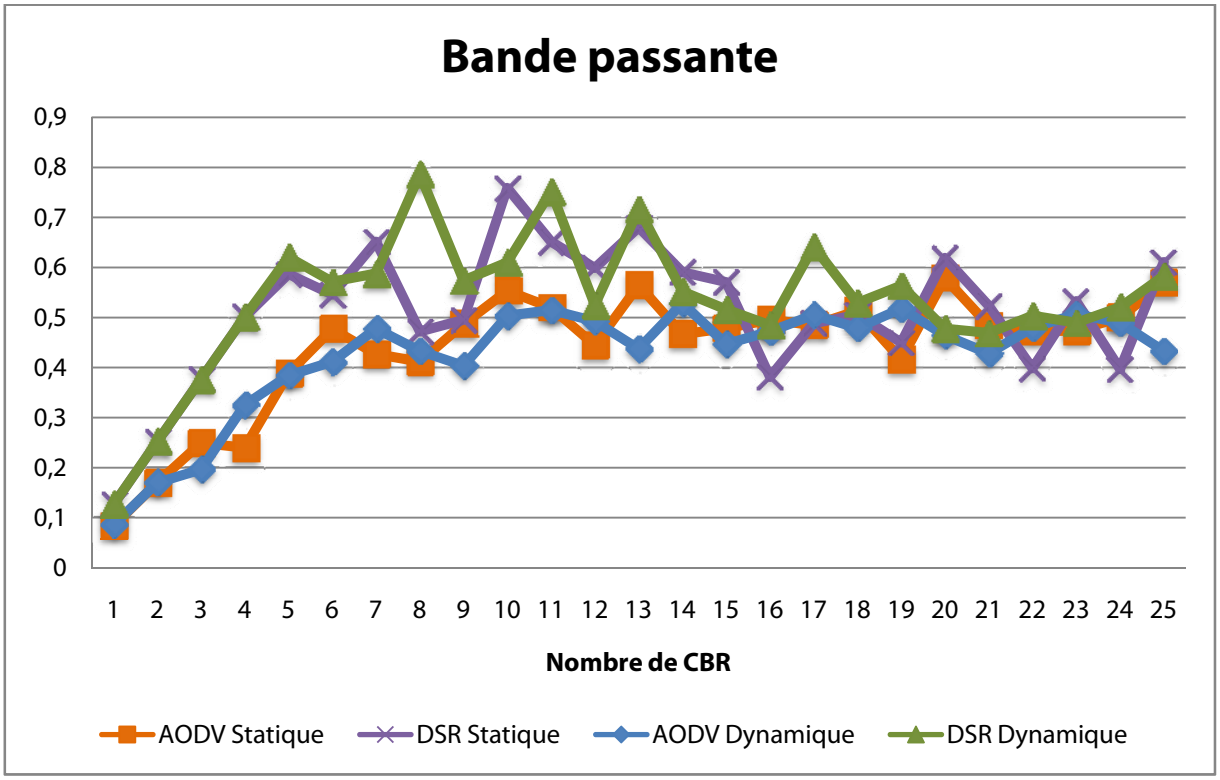
En effet, nous pouvons voir que le délai de bout en bout augmente avec le nombre d'émetteur mais que ce dernier est très faible avec peu d'émetteurs (moins de 6 émetteurs). De plus, nous remarquons également que le délai de bout en bout est beaucoup plus court pour le protocole AODV dès que le nombre d'émetteurs dépasse 5.

3. Conclusions sur les algorithmes pour un réseau ad hoc statique

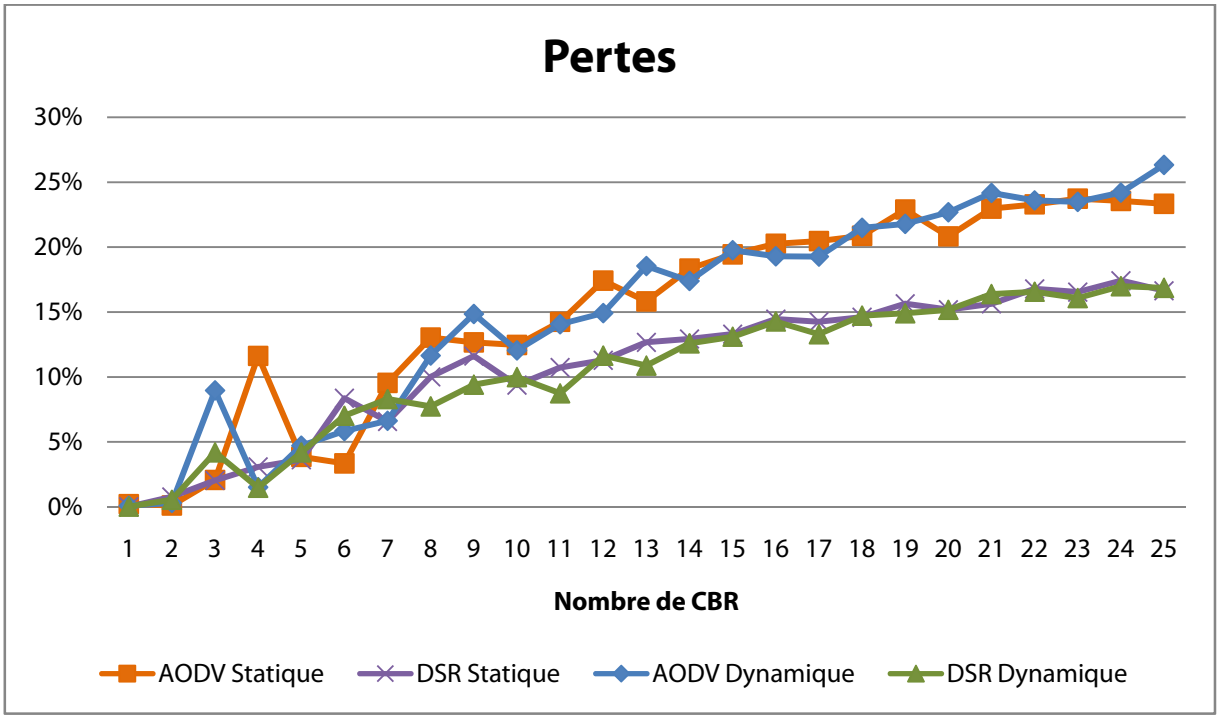
Pour conclure la partie Ad Hoc dynamique, on constate que le protocole DSR est plus adapté qu'AODV pour des topologies présentant peu d'émetteurs. AODV est, quant à lui, plus adapté pour les topologies présentant un nombre plus importants d'émetteurs, ceci malgré de taux important de perte de paquets.

4. Synthèse sur les algorithmes de routage Ad hoc

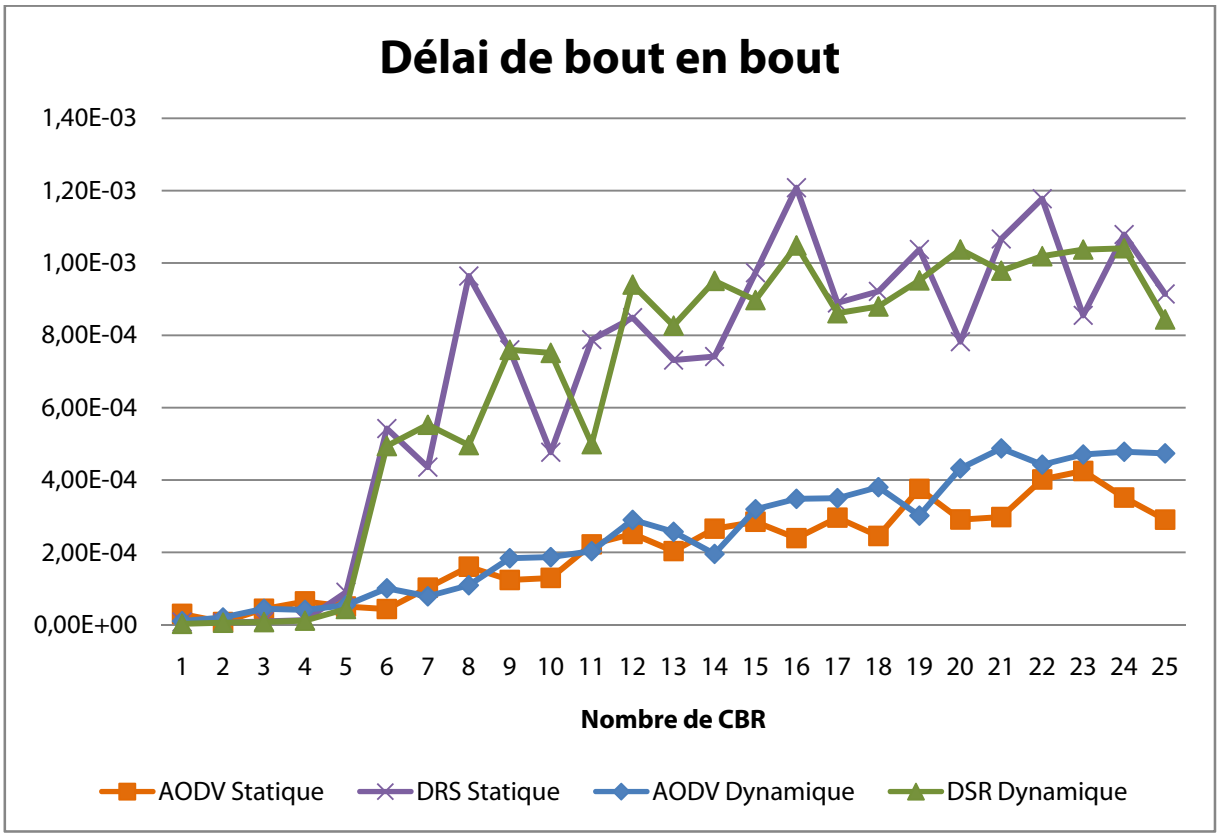
Afin de confirmer nos réflexions précédentes, nous allons maintenant voir des graphiques cumulant l'ensemble des protocoles aussi bien dynamique que statique.



Comme nous l'avions déjà remarqué précédemment, le protocole DSR offre une meilleure bande passante pour un nombre faible d'émetteurs.



De même, les pertes sont plus importante pour le protocole AODV, quelque soit la fonctionnement : statique ou dynamique.



Enfin, nous constatons bien que le délai de bout en bout est toujours plus important pour le protocole DSR dès que le nombre d'émetteurs dépasse un certain nombre (ici, 5).

Au vu de ces résultats, on peut donc conclure que la différence entre le protocole AODV et DSR est faible. Cependant, il est préférable d'utiliser le protocole DSR si l'on est garanti que le réseau comportera peu d'émetteur. En effet, ce protocole permet une meilleure bande passante.

Dans le cas de la mise en place d'un protocole Ad Hoc comportant un nombre important d'émetteurs, le choix du protocole s'opère suivant les besoins. Si le besoin est d'avoir un délai de bout en bout faible, le protocole AODV est dans ce cas à privilégier. Par contre, le protocole DSR est préférable si les pertes de paquets doivent être faibles.

Conclusion

Ce projet nous a tout d'abord permis d'apprendre à comparer différents protocoles réseaux par l'intermédiaire du simulateur de réseau NS.

Cela nous a également permis d'étudier les protocoles AODV et DSR, et ainsi de comprendre dans quelle utilisation l'un était préférable à l'autre.