

CHAPITRE 5

MAINTENANCE DE LA DORSALE VIRTUELLE

5.1 Introduction

Dans un réseau *ad hoc*, un nœud est libre de se déplacer à n'importe quel moment vers n'importe quelle direction. Il peut également ne plus être disponible; sa batterie est épuisée. Ceci a pour effet que tous les liens joignant ce nœud et ses voisins ne seront plus disponibles. Ce qui résulte en une topologie dynamique dans le temps. Comme nous l'avons mentionné dans le Chapitre 4, une dorsale virtuelle construite à partir d'une approximation de l'ensemble MCDS permet de limiter le trafic de contrôle dans le réseau. Rappelons que la détermination de la dorsale nécessite une connaissance globale de la topologie du réseau. Pour ne pas recommencer la construction de la dorsale à chaque changement la topologie, une solution consiste à appliquer une procédure de maintenance. La procédure de maintenance sera chargée de réarranger la structure de la dorsale. Un nœud qui change de position va chercher à se connecter à la dorsale. L'auto-organisation de la structure de la dorsale est alors indispensable pour aider les protocoles de routage à mieux performer.

Dans un tel environnement, la construction de la dorsale est un phénomène rare vu que la connaissance des positions des nœuds est nécessaire, alors que la fonction maintenance est omniprésente pendant toute la durée de vie du réseau. La maintenance s'applique chaque fois qu'il y a détection de mouvement ou disparition d'un nœud dans le réseau. En d'autres termes, le nœud qui change de position ou désire rejoindre à un réseau existant va appliquer la procédure de maintenance pour pouvoir se connecter à la dorsale. Pour appliquer la procédure de maintenance, il faut y avoir une dorsale déjà en

place et des nœuds sont identifiés comme dominants, ceux qui forment la dorsale, et d'autres comme dominés, ceux qui ont au moins un voisin dans la dorsale. La procédure de maintenance ne sera pas la même pour un nœud dominant et un nœud dominé.

La durée de vie d'une dorsale est alors directement liée à la taille de la dorsale et à l'efficacité du mécanisme de maintenance, puisqu'à chaque fois qu'il y a un changement dans la topologie du réseau, il y aura une nécessité à la maintenance. Ainsi, plus la taille de la dorsale est minimale plus la procédure de maintenance est fiable, plus la durée de vie du réseau est longue.

La maintenance de la dorsale est l'ensemble de procédures permettant de maintenir cette structure, l'ensemble MCDS doit rester connecté, afin de garder ses propriétés tout au long du fonctionnement du réseau. C'est une problématique rarement considérée dans les propositions construisant la dorsale. Une approche consiste à reconstruire périodiquement la structure. Cependant, elle peut engendrer des délais induits importants par rapport à la fréquence de changement de la topologie qui peut dégrader la performance du réseau. Ainsi, nous proposons une procédure de maintenance complète qui sera appliquée de façon distribuée [Mnif et Kadoch (2006a)], c'est-à-dire chaque nœud qui change de position et se déconnecte de ses voisins la dorsale va appliquer cette procédure pour se connecter à la dorsale par l'intermédiaire du nœud le plus proche suite à un échange de message de contrôle utilisé pour la découverte de voisinage. La performance de cette procédure sera évaluée au moyen de simulations.

5.2 Préliminaires

5.2.1 Terminologie et définitions

Nous traitons le cas général où nous avons à construire des k -CDS. Le k indique qu'un dominé est situé à k -saut par rapport à son dominant, la figure 34 illustre un exemple pour $k=2$. Il existe des travaux de recherche qui s'intéressent à la construction d'une dorsale à k -saut comme par exemple [Rubin et Vincent (2001)] et [Srivastava et Ghosh

(2002)]. Pour cela, nous voudrions concevoir une procédure générale qui peut servir également à ce type de dorsale. Il est à noter que dans notre recherche nous ne nous sommes intéressés principalement qu'au cas $k=1$. Toutefois, nous allons comparer la performance d'une telle procédure de maintenance pour différentes valeurs de k .

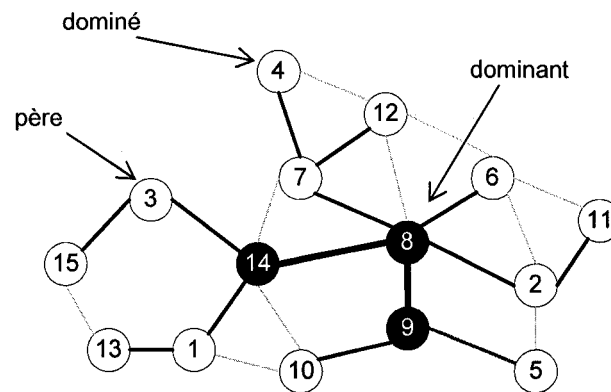


Figure 34 Exemple de 2-CDS

Dans ce qui suit, nous introduisons la terminologie suivante :

- **dominant** : un nœud membre de l'ensemble CDS,
- **dominé** : un nœud qui possède au moins un dominant voisin à k_{cds} sauts,
- **père** : un nœud intermédiaire entre un dominant et un dominé, un nœud père fait partie aussi de l'ensemble des nœuds dominés.

5.2.2 Diagramme d'état

Notre approche pour la maintenance consiste à définir un certain nombre d'états qu'un nœud pourra prendre. Un changement de position pour un nœud entraînera un changement dans les liens radio avec ses voisins. En se basant sur cette observation, nous définissons quatre états possibles pour un nœud :

- **Dominant** : membre de l'ensemble CDS, membre de la dorsale,

- **Dominé** : nœud possédant au moins un dominant à moins de k_{cds} sauts,
- **Actif** : nœud en processus d'être dominant ou dominé,
- **IDLE** : nœud attend une décision d'initialisation.

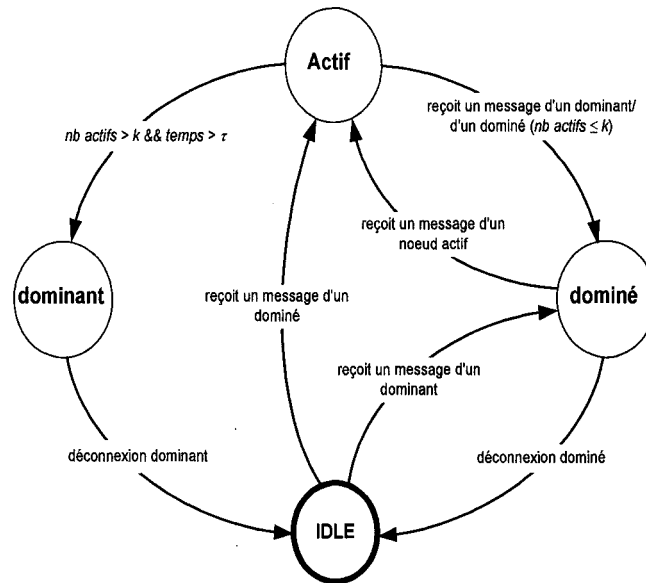


Figure 35 Diagramme d'états

Le diagramme de la figure 35 explique comment on peut passer d'un état à un autre. Le principe de fonctionnement de ce diagramme est comme suit : une fois qu'un nœud change de position, qu'il soit dominant ou dominé, cela va changer son état à IDLE, un état d'initialisation en attendant la réception d'une réponse du voisinage. Si dans le voisinage il existe ou moins un dominant alors le nœud, dans l'état IDLE, va passer à l'état dominé. S'il n'y a aucun dominant au voisinage et qu'il existe un ou plusieurs dominés alors le nœud dans l'état IDLE va alors changer son état à Actif. Un nœud dans un état Actif va changer d'état à dominant s'il est voisin direct d'un dominant et s'il existe k nœuds ou plus, dans l'état actif pour le même dominant et que le temps depuis qu'il a envoyé son premier paquet *hello* est supérieur à τ (où τ un temporisateur qui sera défini ultérieurement). Un nœud Actif peut devenir dominé s'il reçoit un message d'un dominant ou d'un dominé qui se retrouve à une distance $\leq k$ sauts. Il est à noter qu'un nœud dans un état dominé recevant un message *hello* d'un nœud actif va changer son

état à actif. Car, comme nous venons de voir, un nœud dominé peut devenir Actif et par la suite il devient dominant s'il existe plus que k nœuds Actif pour le même dominant. Dans le cas où un nouveau nœud veut rejoindre un réseau en cours, il va se mettre dans l'état IDLE et suivant son voisinage va changer son état à dominé ou dominant en suivant la même logique décrite ci-haut.

Plus formellement, la procédure de maintenance s'écrit :

```

forall x / etat(x) = IDLE do
  if  $N_D(x) \neq \emptyset$ 
    j = argmin {plus_proche(u) |  $u \in N_D(x)$ }
    etat (x) = dominé
    ID_dominant(x) = j
    nb_saut (x) = 1
  elseif  $N_d(x) \neq \emptyset$ 
    j = argmin {nb_saut(u) |  $u \in N_d(x)$ }
    etat (père(j, nb_saut = 1)) = dominant
    for m=2 to nb_saut
      etat(père(j,m)) = Actif
       $N_A(u) = N_A(u) \cup$  père(j,m)
    end
    D = D  $\cup$  j
  end
end
for all u  $\in$  D
  for all x  $\in$   $N_A(u)$ 
    if (nb_saut(x) < k &&  $N_d(x) \neq \emptyset$ )
      etat (x) = dominé
      ID_dominant(x) = u
    else if (nb_actifs (u) > k)
      etat (x) = dominant
      nb_saut (x) = 0
      ID_dominant(x) = NULL
    end
  end
end
return x(ID_dominant, état, nb_saut)

```

/* il existe un nœud dominant au voisinage de x */
/* s'il existe plus qu'un nœud dominant au voisinage de x, on choisit celui le plus proche avec $N_D(x)$ l'ensemble des nœuds dominants au voisinage de x,*/
/* mettre à jour les attributs de x */
/* il existe un nœud dominé au voisinage de x */
/* s'il existe plus qu'un nœud dominé au voisinage de x, on choisit celui qui a le nb_saut le plus faible */
avec $N_D(x)$ l'ensemble des nœuds dominants au voisinage de x,*/
/* le nœud x ne peut pas se connecter au dominant de j, j=k*/
/* tous les pères du nœud j vont changer d'état à Actif et le premier va changer en dominant
 $N_A(u)$ l'ensemble de nœuds Actifs du dominant u*/
/* Un père à nb_saut d'un dominé est noté par père(dominé, nb_saut)*/
/* le nœud j va être ajouté à l'ensemble des dominants D dans tout le réseau*/
/* mettre à jour les attributs de x */
/* On effectue un test sur le nombre des nœuds actifs du dominant u
/* Un nœud x possède les attributs suivant ID_dominant, état et nb_saut avec nb_saut le nombre de saut depuis le dominant (nb_saut \leq k) */

5.2.3 Format du paquet *hello* modifié

Dans les protocoles de routage actuels, on utilise le paquet *hello* pour la découverte de voisinage. Nous avons donc choisi d'utiliser ces mêmes paquets classiques dans lesquels nous allons ajouter quelques champs (figure 36). Pour un dominant, il est nécessaire de mettre à jour une liste de ses dominés et des pères associés. Ceci peut se faire en obtenant ces informations des paquets *hello*.

Source (32 bits)			
Destination (32 bits)			
TLL (4 bits)	Id (16 bits)	Type (8 bits)	Etat (4 bits)
Degré (8 bits)			
dominant (32 bits)			
père 1 (32 bits)			
...			
père k-1 (32 bits)			

Figure 36 Format du paquet *hello* modifié

Le paquet *hello* comprend les champs classiques tels que les adresses source et destination, le TTL¹⁴, le type de paquet¹⁵ et l'identifiant du paquet.

Nous ajoutons alors un certain nombre d'informations qui seront utiles pour la maintenance de la dorsale, telles que :

- état du nœud : IDLE/actif/dominant/dominé,
- degré : nombre de voisins,

¹⁴ *Time To Live* : nombre maximum de sauts à parcourir par un paquet. Ce nombre décroît à chaque saut.

¹⁵ Un paquet peut être de type *hello*, *unicast*, *data* etc.

- nœud dominant pour un nœud dominé,
- liste des pères pour un nœud dominé.

5.2.4 Temporisateur τ

Un temporisateur τ sert à ce qu'un nœud en mode actif ne se déclare pas dominant avant de vérifier le degré et l'état de tous les nouveaux nœuds actifs. Lorsqu'un nœud se déclare actif, il va diffuser le message de changement d'état à $(k-1)$ -sauts. De même, le dernier nœud qui recevra ce message de changement d'état va le retourner à $(k-1)$ -sauts. Le temporisateur τ doit alors être supérieur au temps aller-retour qu'un paquet va effectuer dans le trajet de $(k-1)$ -sauts. Soit Δ_t le délai maximum qu'un paquet subit pour le traitement, Δ_t est de l'ordre de 20~50 millisecondes.

$$\tau \geq 2 \cdot (k-1) \Delta_t$$

$$\text{pour } \Delta_t = 50ms, \text{ alors } \tau_{\min}(ms) = 100 \cdot (k-1)$$

Dans cette expression, nous ne prenons pas en considération le temps de propagation vu qu'il est négligeable comparé au temps de traitement d'un paquet.

5.3 Procédure de maintenance

Dans cette section, nous allons détailler le principe de fonctionnement de la procédure de maintenance proposée dans ce chapitre. Pour réaliser la maintenance, différentes tâches doivent être entreprises par le nœud qui désire se connecter à la dorsale. Tout d'abord, le nœud doit effectuer une découverte au voisinage. Ceci est réalisé en envoyant des messages *hello*. Par la suite, il se connecte à la dorsale. Suivant les messages réponses reçus de son voisinage, une procédure de connexion est alors employée en utilisant le diagramme d'état (figure 35). Deux autres tâches sont

nécessaires pour réaliser la maintenance, il s'agit de maintenir des informations pour chaque nœud et de maintenir les relais vers le dominant.

5.3.1 Découverte au voisinage

Un nœud dominé ou dominant qui se déplace (ou même un nouveau nœud qui désire se joindre au réseau) va donc tenter de se connecter à la dorsale, son état est IDLE. Un nœud envoie un paquet *hello* contenant sa propre adresse et un TTL égal à k . Le paquet est alors diffusé en *broadcast*. Nous avons choisi le $TTL = k$ pour que ce message *hello* puisse arriver à un dominant, s'il existe. Les nœuds voisins vont comprendre que ce nœud est IDLE puisque le champ dominant est nul, et vont donc l'ajouter dans leur table de voisinage. Chaque nœud va alors envoyer une réponse *hello* dans lequel les champs degré, état, dominant et la liste des pères sont bien identifiés. Pour un dominé, il doit indiquer dans son paquet de retour un TTL égal au nombre de pères dans sa liste, ceci afin d'informer son dominant.

Quoique cette méthode possède l'inconvénient de générer beaucoup de paquets (puisque chaque nœud envoie un paquet *hello* à chaque intervalle de *hello*), elle converge rapidement. Chaque nœud connaîtra son voisinage, s'il n'y a pas de perte de paquets de *hello*. Il existe une autre méthode où chaque nœud envoie la table de ses voisins à un saut. Ainsi, un nœud connaîtra de proche en proche son k -voisinage. La convergence de cette méthode est beaucoup plus lente, car la propagation se fait sur un saut malgré que la charge soit moins faible avec un seul paquet de taille plus grande.

5.3.2 Procédure de connexion

Un nœud dans l'état IDLE (et donc n'a pas de dominant) recevant une réponse d'un dominant (ce dernier peut alors couvrir le nœud) va changer son état à dominé et la procédure se termine. Si aucun dominant n'est à son voisinage et si un ou plusieurs

dominés lui répondent, il change son état à actif pour finalement devenir soit dominant soit dominé :

- un nœud actif va être dominant s'il est le père de k dominés y compris le dernier nœud qui cherche à se connecter. Ce nœud se trouve à un saut d'un dominant,
- un nœud actif va être dominé si un dominant peut l'accepter comme fils c'est-à-dire s'il se retrouve à une distance inférieure à k -sauts du dominant.

En recevant un message d'un nœud actif, un dominé d'ordre k_{CDS} va changer son état à actif parce qu'il pourrait devenir dominant si un nouveau nœud s'ajoute à la branche de k -dominés. Le dominé d'ordre 1 change son état à dominant pour pouvoir couvrir le nouveau nœud (dominé). Un dominé qui change son état à dominant se retrouve à un saut d'un dominant, et il sera donc connecté à la dorsale.

Reprenons le même exemple de la figure 34, où nous avons un réseau composé de 15 nœuds et la dorsale est constituée des nœuds 14, 8 et 9 ($k_{CDS} = 2$). Le nœud 13 va se déplacer et le nœud 2, qui était dominé avant la présence du nœud 13, devient un dominant afin de pouvoir couvrir le nœud 13 dans sa nouvelle position (figure 37-b).

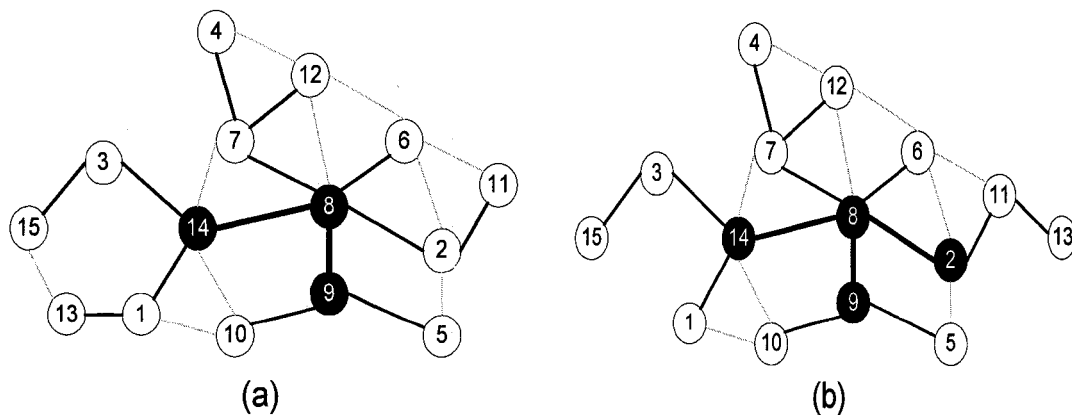


Figure 37 La structure s'auto organise suite à un déplacement d'un dominé

Le nœud 13 dans sa nouvelle position, en état IDLE, va tenter de découvrir le voisinage en envoyant des paquets *hello*. En recevant ces paquets, le nœud 11 va changer son état à actif et va faire savoir qu'il est dominé et qu'il possède un dominant à un saut. Le nœud 2 sachant qu'il y a un nœud qui tente de se connecter, va aussi changer son état à actif, en conséquence de l'action du nœud 11. Dans cet exemple, nous avons $k_{CDS}=2$, le nœud 2 devra donc changer son état à dominant pour avoir une dorsale connexe.

Considérons maintenant le cas où le nœud 14, dominant, va se déplacer. Les nœuds 3, 15, 13 et 1 qui étaient des dominés du nœud 14 vont changer leur état à IDLE et vont lancer le processus de découverte du voisinage. Dans un premier temps le nœud 10 va donc répondre au nœud 1 en lui indiquant qu'il peut se connecter. Une fois que le nœud 1 répond, le nœud 10 change son état à actif en même temps que le nœud 15. Étant donné que dans cet exemple $k_{CDS} = 2$, le nœud 10 doit changer d'état à dominant pour couvrir les nœuds 1 et 13. De la même façon, les nœuds 1 et 13 vont devenir dominants pour couvrir le nœud 3 (figure 38). Le nœud 14 qui se retrouve dans sa nouvelle position, dans un état actif, va lancer le processus de découverte de voisinage en envoyant des paquets *hello*. Le nœud 5 va répondre à ces paquets *hello* pour faire savoir qu'il est dominé et qu'il se trouve à un saut d'un dominant, le nœud 14 passe alors de l'état actif à dominé.

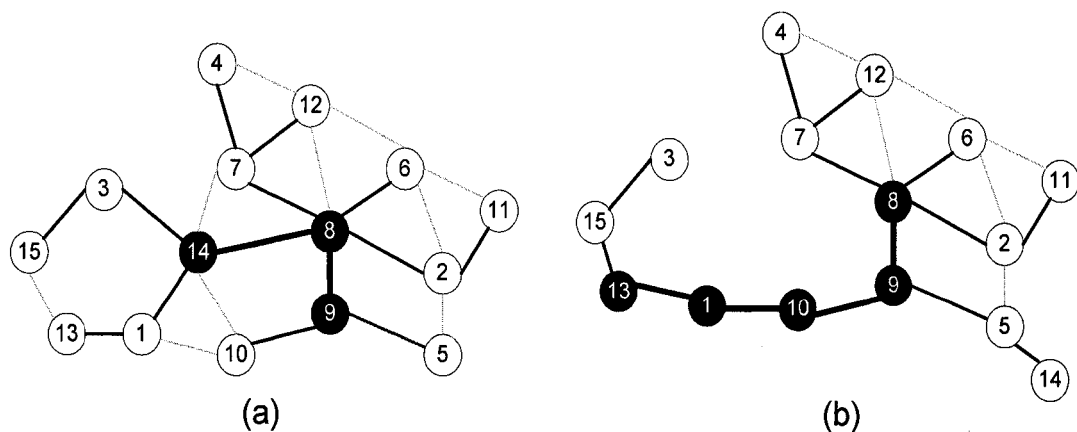


Figure 38 La structure s'auto organise suite à un déplacement d'un dominant

5.3.3 Maintien des informations

Nous avons introduit dans le paquet *hello* certaines informations propres à la dorsale telles que : état, degré et liste des pères. Nous pouvons alors utiliser ces données pour mettre à jour les informations sur la dorsale. Grâce au champ père, nous pouvons mettre à jour la liste des dominés pour chaque dominant. Ces informations peuvent servir à optimiser la structure de la dorsale après avoir effectué plusieurs maintenances successives.

5.3.4 Relais vers le dominant

Avec la structure d'une dorsale, un dominé a toujours besoin de dialoguer avec son dominant. Il doit donc posséder une route vers celui-ci. Un dominant envoie dans sa zone, de façon régulière, des paquets de contrôle ou des paquets *hello*, il suffit alors de noter l'identité du père précédent pour déterminer l'intermédiaire à contacter sur le chemin vers le dominant.

Il est important pour un nœud de pouvoir communiquer avec son dominant quand c'est nécessaire. Un nœud source qui désire envoyer des données vers un autre nœud destination va alors demander à son dominant de trouver un chemin et d'acheminer ce trafic de données. C'est le dominant qui va procéder à la recherche d'une route et à l'acheminement du trafic. Le dominant connaît a priori la destination et le dominant qui lui est associé. Une telle communication est obligatoire pour certaines procédures de maintenance. Nous stockons alors l'identification du nœud (ID) qui relaie les paquets de contrôle venant du dominant. Ce relais est donc mis à jour pour devenir dominé après la réception d'un paquet d'un dominant.

Remarque

Il est important de mentionner qu'il est possible qu'un ou plusieurs nœuds restent non connectés à la dorsale. Dans ce cas, le nœud isolé reste dans un état IDLE jusqu'à ce que la situation change et d'autres nœuds se rejoignent à lui. D'autres cas peuvent aussi se présenter. En effet, deux ou plusieurs sous-graphes peuvent se former après plusieurs maintenances. Dans cette situation, les sous-réseaux correspondants vont alors être indépendants jusqu'à ce qu'un nœud change de position et se trouve à la portée des nœuds qui n'appartiennent pas aux mêmes sous réseaux. Après l'échange périodique des paquets *hello*, le nœud va découvrir que ses voisins n'appartiennent pas à la même dorsale et par conséquent il change son état à dominant, ainsi que les nœuds dominés dans les chaînes vers les dominants, et les deux sous réseaux seront alors connectés. La figure suivante illustre ce cas particulier où nous avons $k_{CDS} = 2$ et à un instant donné deux sous-réseaux sont formés (figure 39-a). Ces deux sous-réseaux restent déconnectés jusqu'à ce que le nœud 4, par exemple, aille changer de position pour se retrouver à la portée des nœuds 11 (du sous-réseau 1) et 3 (du sous-réseau 2). Le nœud 4 ainsi que les nœuds dans les chaînes vers dominants (nœuds 2 et 11 dans le sous-réseau 1 et nœud 3 dans le sous-réseau 2) vont changer d'état à dominant (figure 39-b).

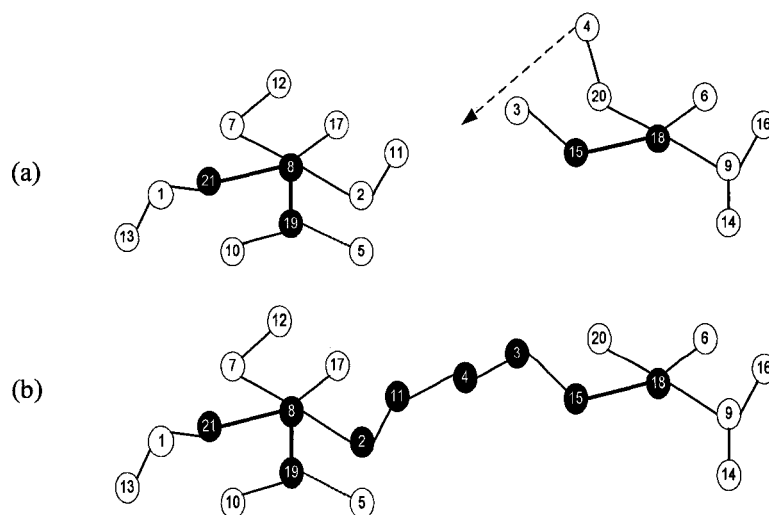


Figure 39 Cas où deux sous-réseaux se forment

Théorème 2 (maintenance)

En appliquant la procédure de maintenance sur une dorsale construite à partir de l'ensemble k_{CDS} , un dominé possède toujours un dominant, et il est d'au plus à une distance de k_{CDS} sauts de ce dominant.

Preuve : Nous supposons que la topologie est stable après quelques maintenances. Un dominé ayant un voisin dominant va choisir ce dernier comme dominant. Supposons qu'un ensemble de dominés se retrouvant à une distance de i sauts de leur dominant ait un père. Un dominé situé à $i+1$ sauts de son dominant peut choisir ce dernier comme dominant car il est situé à moins de k_{CDS} sauts via un autre dominé ayant choisi le même dominant, mais à i sauts, avec $i < k_{CDS}$. Ainsi, comme le père des dominés à i sauts d'un dominant existe, chaque dominé qui choisit un dominant possède un père.

Un dominé n'ayant aucun dominant possible dans sa table de voisinage (il n'existe aucun voisin qui pourra être choisi comme dominant situé d'au plus à $k_{CDS}-1$ sauts) il devient actif pour passer finalement à l'état dominant. D'après le diagramme d'état de la figure 35, un nœud actif va devenir dominé s'il existe un dominant valide, ou bien, il va devenir dominant et par la suite il va exécuter la procédure de maintenance d'un dominant. Ainsi, tout dominé possède au moins un dominant situé à moins de k_{CDS} sauts. ■

Théorème 3 (connectivité)

Soit un graphe connexe et une dorsale construite à partir de l'ensemble CDS. Suite à maintenance de la dorsale, l'ensemble des dominants forme un arbre et reste connecté quand les liens radio sont stables.

Preuve :

Supposons que les liens radios sont stables (topologie stable) après quelques maintenances. Chaque dominant reçoit alors les paquets *hello* maintenant ainsi la liste de ses dominés. Soit D_i l'ensemble des dominants qui forment la dorsale, l'ensemble D_i est supposé être connexe. L'indice i représente le nombre maximum de sauts entre deux dominants. Soit l'ensemble D_{i+1} tels que pour tout nœud de D_{i+1} on choisit s l'ensemble D_i , ainsi, l'ensemble D_{i+1} est connexe puisque ces nœuds reçoivent le paquet *hello* de leur père et ils sont alors à un saut de la dorsale.

Supposons que D_i ne contient pas de cycle; E_i est l'ensemble des arcs de D_i , et V_i est l'ensemble de ses nœuds. Nous avons alors $|E_i| = |V_i - 1| = |V_i| - 1$. Pour chaque nœud de l'ensemble $D_{i+1} - D_i$, on ajoute un nœud dans V_i et un arc dans E_i , alors :

$$|E_{i+1}| = |V_i| - 1 + [|V_{i+1}| - |V_i|] = |V_{i+1}| - 1$$

Donc D_{i+1} est connexe et ne contient pas de cycle. ■

5.4 Modélisation

Pour évaluer l'efficacité de la procédure de maintenance, *Opnet Modeler* est choisi, en utilisant à la couche 2 le protocole 802.11b qui est déjà intégré dans le simulateur. *Opnet* est un simulateur à événements discrets qui fournit un environnement de développement global permettant de modéliser et d'évaluer les performances des réseaux de communication. Il permet de spécifier le format des paquets, de définir des processus qui

seront représentés par des diagrammes d'états et enfin, de collecter et d'analyser les résultats. Un processus est représenté par un automate état/transition. Chaque état contient le code des tâches à exécuter à l'entrée et à la sortie de l'état. La transition spécifie la condition nécessaire pour passer d'un état à un autre.

La figure 40 illustre le modèle de simulation du plus haut niveau implémenté dans Opnet. Le réseau comprend 50 nœuds sont placés aléatoirement dans une zone carrée de $1,5 \text{ km} \times 1,5 \text{ km}$. Chaque terminal est représenté par un nœud, modélisé comme un émetteur/récepteur, une couche mac 802.11b (`wireless_lan_mac`) reliée au dispositif d'émission/réception et une couche MCDS au dessus de la couche 802.11b comme illustré dans la figure 41. Un processus indépendant est ajouté pour modéliser la mobilité d'un nœud et un processus d'interface pour établir l'interface entre la couche MAC et la couche MCDS.

Chaque processus est représenté par un diagramme d'état : lorsqu'on entre dans un état, une série d'instructions sont exécutées; d'autres instructions sont également à la sortie de l'état. Au moment où un événement se produit (interruption, arrivée d'un paquet, etc.), on sort de l'état et suite à une transition vraie pour entrer dans un autre état. Le diagramme d'état développé pour le processus MCDS est illustré à la figure 42.

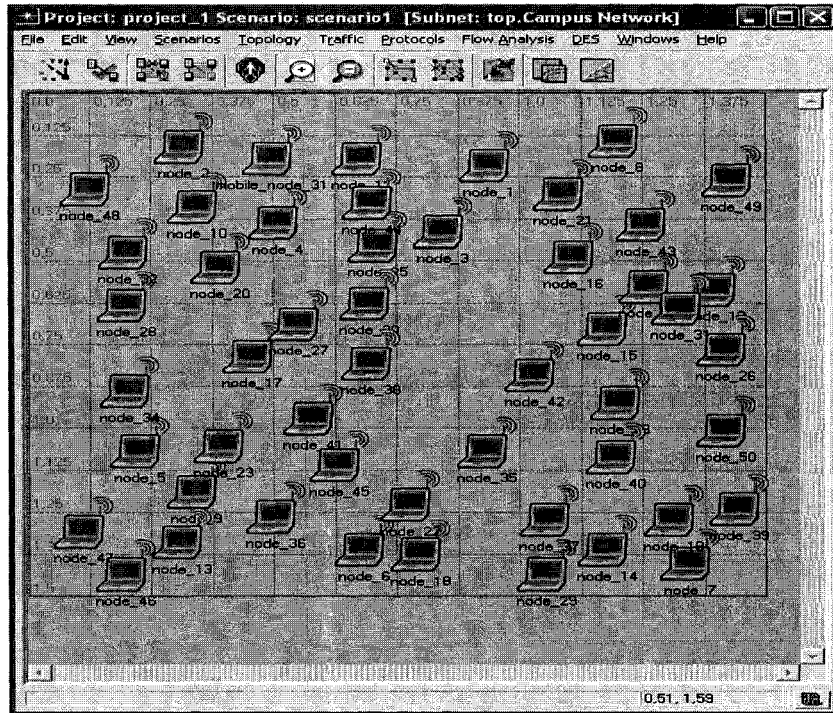


Figure 40 Vue d'ensemble du modèle de simulation ($N=50$)

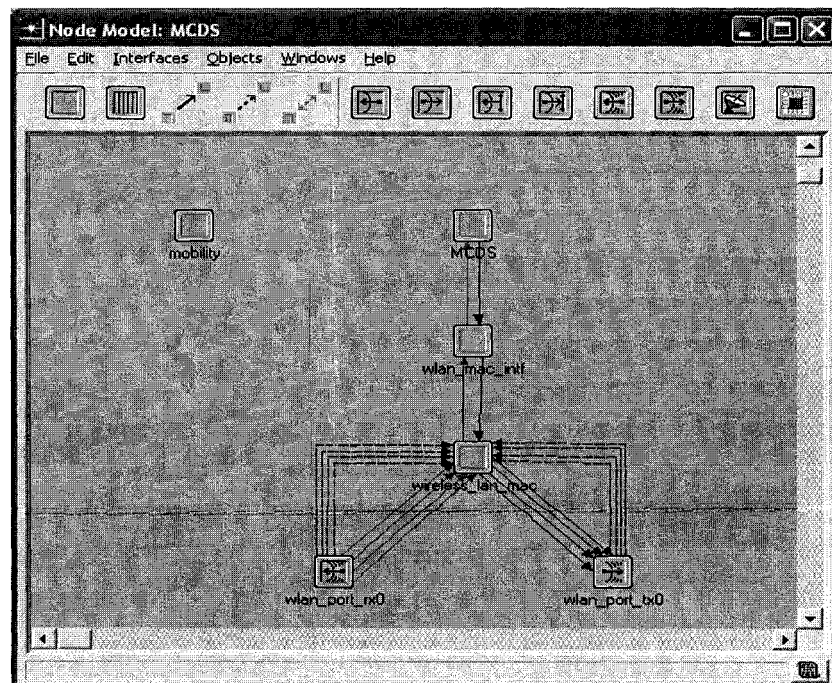


Figure 41 Modélisation d'un terminal *ad hoc*

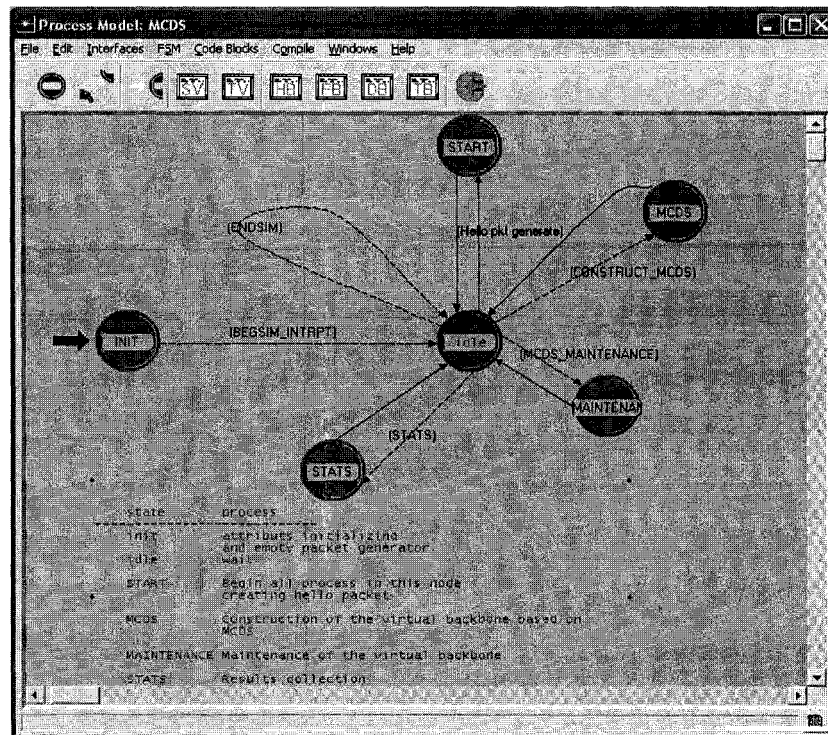


Figure 42 Diagramme d'état pour le processus MCDS

5.4.1 Modèle de mobilité

Pour le modèle de mobilité, nous avons utilisé le modèle le plus fréquent pour modéliser le mouvement des nœuds dans le réseau *ad hoc* soit le modèle *Random WayPoint Model* (RWP). Le modèle RWP a été proposé pour la première fois par Johnson et Maltz dans leur simulation de DSR [Johnson et Maltz (1996)]. Cinq paramètres décrivent l'environnement de simulation pour le modèle RWP original :

- taille et forme de la région de déploiement Q ,
- distribution des nœuds dans la région Q ,
- un temps de repos (*pause time*) constant pour chaque nœud, et
- vitesse minimale et vitesse maximale $0 \leq v_{min} \leq v_{max}$.

Dans le modèle RWP, chaque nœud choisit une position destination et une vitesse suivant une distribution uniforme et se dirige vers la destination. Quand le nœud arrive à

la position destination, il s'arrête pour une certaine durée de temps (temps de repos). Le temps de repos est choisi aléatoirement suivant une distribution uniforme. Différents travaux de recherche ont été publiés pour montrer les insuffisances de ce modèle [Bettstetter et al. (2003)] et [Yoon et al. (2003)]. En effet, ces travaux montrent que ce modèle ne permet pas d'atteindre un état d'équilibre (*steady state*) en termes de :

- distribution des nœuds,
- vitesse moyenne par nœud.

Bettstetter *et al.* ont étudié le problème de distribution des nœuds comme étant l'effet de bordure et ceci vient du fait qu'un nœud choisira sa prochaine position de destination uniquement dans la zone du réseau prédéfinie et ne doit pas aller en dehors de cette zone. Un nœud proche de la bordure de la zone va choisir une position de destination proche du centre de cette zone. Ils ont démontré que la distribution des positions n'est pas uniforme et que la distance entre le bord de la zone et le centre a un effet sur le résultat. Afin de résoudre ce problème, nous suivons la recommandation des auteurs [Bettstetter et al. (2003)]. Cette recommandation consiste à introduire un paramètre de stabilité, p_s , avec $0 \leq p_s \leq 1$. Ce paramètre représente la probabilité qu'un nœud reste statique pendant toute la durée de la simulation. Bettstetter *et al.* ont montré qu'avec l'introduction de ce paramètre, le système pourra atteindre son régime permanent plus rapidement.

L'autre problème avec le modèle RWP est que la vitesse moyenne d'un nœud tend vers zéro et non pas vers une valeur moyenne entre la valeur maximale et la valeur minimale. Ceci se produit surtout quand la valeur minimale de la vitesse prend la valeur zéro et que la vitesse maximale est relativement faible. Pour remédier à ce problème,

nous prenons alors :

- $v_{min} \neq 0$,
- une vitesse qui varie uniformément entre v_{min} et v_{max} pour avoir une moyenne autour de $\frac{v_{min} + v_{max}}{2}$
- un temps de repos variable compris entre $[t_{min}, t_{max}]$, avec $t_{min} \neq 0$.

5.4.2 Évaluation de la procédure de maintenance

Comme nous l'avons mentionné dans notre méthodologie, la construction et la maintenance de la dorsale se font avec deux algorithmes différents. De ce fait, nous supposons que la phase d'établissement est déjà réalisée et à l'instant $t=0$ de la simulation une dorsale est déjà formée en utilisant l'algorithme proposé dans le chapitre 3. En d'autres termes, avant de lancer la simulation et les nœuds commencent à se déplacer un certain nombre de nœuds ont été sélectionnés comme nœuds dominants et les autres nœuds sont alors des dominés.

Nous avons effectué un certain nombre de simulations pour apprécier le comportement et la performance de notre solution de maintenance vis-à-vis différents paramètres tels que la variation du nombre de dominants dans le temps, la variation de pourcentage de connexions à la dorsale dans le temps, l'effet de la mobilité et l'effet du nombre total de nœuds dans le réseau.

Dans un premier temps, nous nous intéressons au comportement de cette procédure pour une durée de vie d'un réseau *ad hoc*. La taille du réseau est $N = 50$ nœuds. Un nœud se déplace avec une vitesse moyenne de $10m/s$. Le temps moyen de repos (t_{moy}) est égal à 50 secondes. Un nœud possède une portée radio $R = 300m$, paramètre par défaut du modèle 802.11b d'*Opnet*. La durée de vie d'un réseau est choisie égale à une heure (3600s). Nous déterminons le nombre de dominants dans tout le réseau durant la période de simulation ; les valeurs sont prises toutes les 10 secondes.

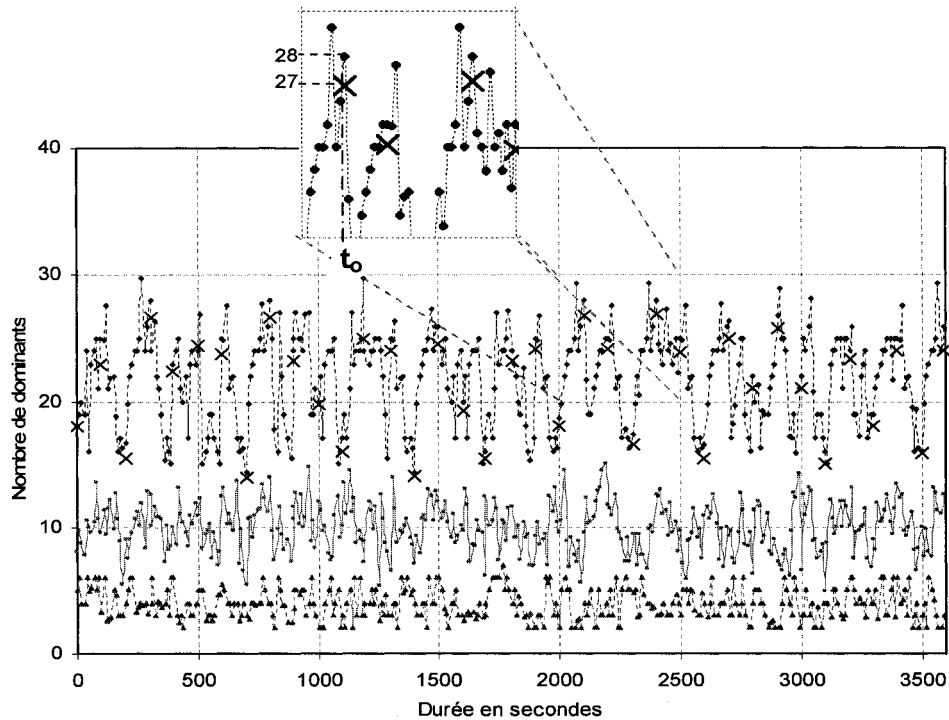


Figure 43 Variation du nombre de dominants en fonction du temps pour $k_{cds} = 1, 3$ et 5

La figure 43 représente les résultats de simulation pour $k_{cds} = 1, 3$ et 5 . Ces résultats montrent que plus la valeur de k augmente plus la structure est stable dans le sens où le nombre de dominants ne change pas rapidement; absence des pics comme pour $k = 1$. En ce qui concerne la taille de la dorsale après l'application successive de la procédure de maintenance, nous pourrions vérifier que la solution a tendance de converger vers la solution donnée par la construction. Sur la partie zoomée de la figure 41, nous voyons clairement que la taille de la dorsale est très proche de la taille de l'ensemble MCDS. La taille de l'ensemble MCDS est représentée par des "x", et il est calculé en utilisant l'algorithme de construction tout en considérant qu'il s'agit d'un système statique à des instants bien déterminés. Par exemple à l'instant $t_o = 2100$ s, la taille de la dorsale après la maintenance est égale à 28 et la taille de l'ensemble MCDS est égale à 27. Ces résultats démontrent que la procédure de maintenance s'adapte bien avec la variation de

la topologie du réseau et a tendance à converger vers la solution donnée par l'algorithme de construction de la dorsale. L'écart entre la solution donnée après la maintenance et la solution donnée par la construction est de l'ordre de 6%. Dans le cas où cette solution s'éloigne de la solution optimale une initialisation du réseau est alors nécessaire et le coût engendré par cette action sera le temps nécessaire pour exécuter l'algorithme de construction proposé dans le chapitre 3.

Une deuxième série de simulations consiste à évaluer la robustesse de la procédure de maintenance dans un environnement dynamique. Pour cela, nous allons vérifier la connectivité de la structure formée par la dorsale dans le temps; en d'autres termes, nous allons déterminer (en %) le nombre de nœuds qui restent connectés à la structure globale formée par la dorsale en présence de la procédure de maintenance. Avec les mêmes paramètres que précédemment : $N = 50$, $v_{moy} = 10m/s$, $t_{moy} = 50s$, $R = 300m$ et une durée de simulation de 3600 secondes, soit la durée de vie du réseau. Nous illustrons uniquement le cas pour $k = 1$ étant donné que c'est le cas où la structure est la plus critique, d'après les résultats de simulations précédentes (figure 44). La structure globale du réseau est peu sensible à la mobilité; la variation du pourcentage de connexions dans le temps. En moyenne, les nœuds restent connectés 92,6 % du temps. Cette valeur est plus spécifiquement de 94,2 % et 95,3 % pour $k = 3$ et $k = 5$ respectivement. Les creux, là où le pourcentage chute au-dessous de cette moyenne, correspondent à une variation brusque et importante du nombre de dominants ce qui fait qu'un nombre important de nœuds entrent dans un processus de changement d'état.

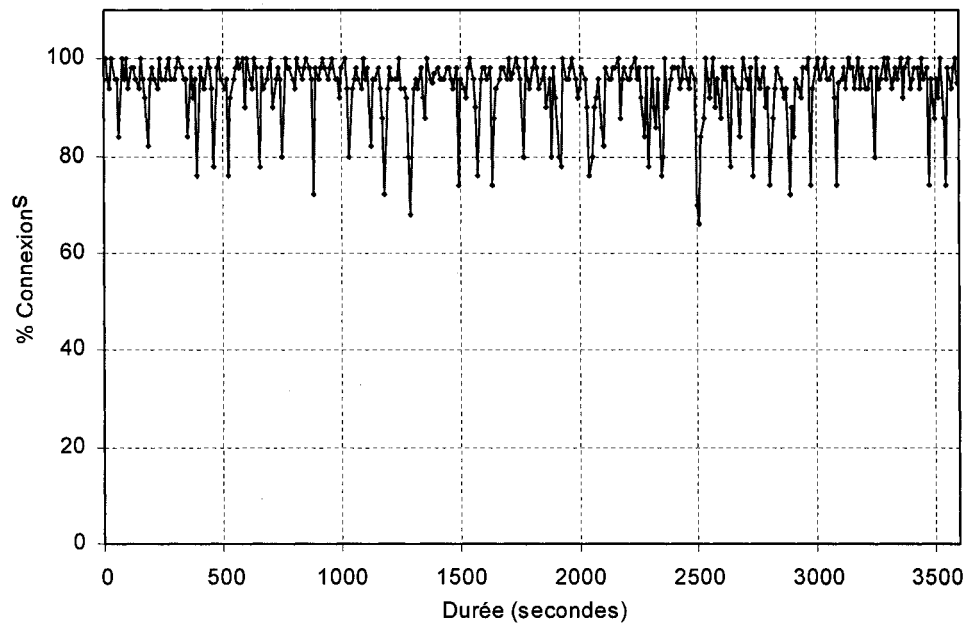


Figure 44 Variation du pourcentage de nœuds connectés en fonction du temps pour $k_{cds}=1$

Nous avons aussi collecté des valeurs reliées à la performance de la procédure de maintenance telles que :

- la durée moyenne qu'un nœud reste dans l'état dominant,
- la durée moyenne qu'un dominé reste avec le même dominant,
- le temps moyen nécessaire pour qu'un nœud (dans l'état IDLE) se connecte,
- le nombre moyen de dominés par dominant.

Tableau V
 Résultats pour différentes valeurs de k_{mcds}

	$k_{mcds} = 1$	$k_{mcds} = 3$	$k_{mcds} = 5$
durée moyenne qu'un nœud reste dans l'état dominant (s)	110 (7.4) [♦]	118 (8.5)	105 (6.4)
durée moyenne qu'un dominé reste avec le même dominant (s)	64 (4.2)	86 (7.5)	94 (6.7)
temps moyen nécessaire pour qu'un nœud se connecte (ms)	185 (12)	254 (18)	312 (27)
nombre moyen de dominés par dominant	2.7 (0.7)	5.2 (1.3)	9.2 (1.8)

[♦]La valeur entre () représente l'écart type.

D'après les résultats trouvés (tableau V), nous avons pu remarquer que les durées (la durée moyenne qu'un nœud reste dans l'état dominant ainsi que la durée moyenne qu'un dominé reste avec le même dominant) possèdent le même ordre de grandeur pour différents valeurs de k_{cds} . Par contre, ce paramètre k_{cds} a un impact direct sur le temps moyen nécessaire pour qu'un nœud se connecte. Ceci est dû au fait qu'un nœud qui va tenter de se connecter et qui se retrouve à k_{cds} sauts d'un dominant doit attendre que le message *hello* envoyé atteigne le dominant s'il existe. D'autre part, plus la valeur de k_{cds} augmente, plus le nombre de dominés attachés au même dominant augmente. Ceci s'explique par le fait qu'il y a moins de dominants pour le même nombre de terminaux dans le réseau.

5.4.2.1 Effet de la mobilité

Nous voulons évaluer l'influence de la mobilité sur la performance de la procédure de maintenance. Pour ce faire, nous allons faire varier la vitesse de 1 à 30 m/s. Les autres paramètres de simulation sont les mêmes que précédemment. Pour chaque valeur de k_{cds} , nous déterminons la valeur moyenne (en %) des nœuds qui restent connectés à la dorsale ainsi que le nombre de dominants dans la dorsale. La figure 35 illustre les résultats de

simulations pour $k_{c_{ds}}=1, 3$ et 5 . Ces résultats montrent que la proportion moyenne de nœuds connectés à un dominant, à un temps donné, diminue légèrement avec la mobilité. Mais elle reste au delà de 90 %, ce qui représente un taux de connectivité élevé même quand la vitesse augmente. La variation du pourcentage de connexion est pratiquement identique quand $k_{c_{ds}}$ varie.

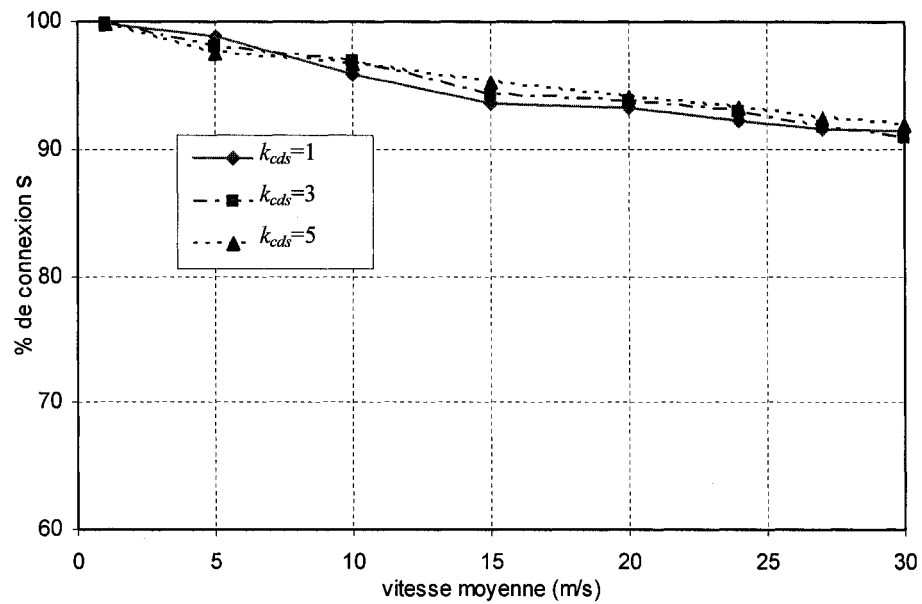


Figure 45 Pourcentage de connexions en fonction de la vitesse moyenne

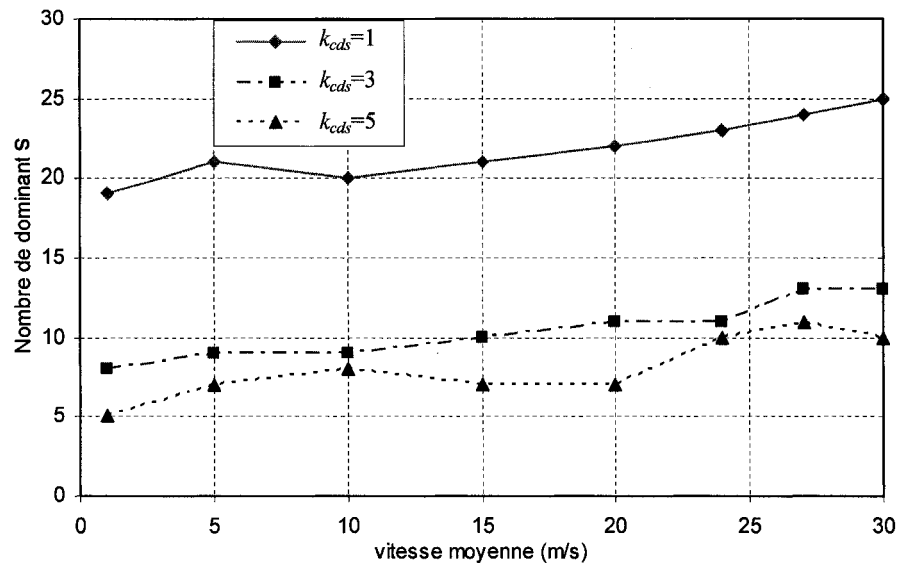


Figure 46 Variation du nombre de dominants en fonction de la vitesse moyenne

Les résultats de la figure 46 montrent la variation de la valeur moyenne du nombre de dominants quand la mobilité varie. Le nombre de dominants a tendance à croître légèrement quand la mobilité augmente. Ce nombre ne dépasse pas la moitié du nombre total des nœuds pour une vitesse moyenne de 30 m/s et $k_{c_{ds}} = 1$. Ces résultats témoignent bien de l'efficacité de la procédure de maintenance conçue dans ce chapitre. Elle est capable de bien s'adapter même quand la mobilité augmente.

5.4.2.2 Effet de la taille du réseau

Le problème de mise à l'échelle (*scalability*) est largement considéré dans les réseaux *ad hoc* et représente un critère primordial pour évaluer la performance des mécanismes de routage. Nous examinerons l'effet du nombre de nœuds dans le réseau sur l'efficacité de notre procédure de maintenance. Nous prenons alors des réseaux dont le nombre de nœuds varie de 20 à 80, la vitesse moyenne est de 10m/s et les autres paramètres de simulation sont les mêmes que précédemment.

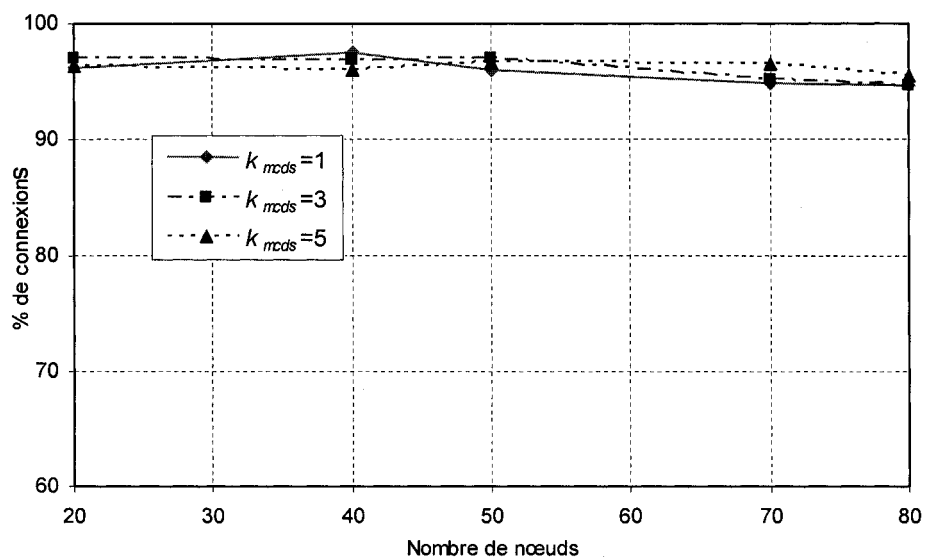


Figure 47 Le pourcentage de connexion en fonction de la taille du réseau

D'après la figure 47, nous observons que le pourcentage de nœuds connectés est peu sensible à la taille du réseau, bien que la taille de la dorsale soit proportionnelle à la taille du réseau (chapitre 4). Le nombre de nœuds dans le réseau n'a pas d'effet sur l'efficacité de notre procédure de maintenance étant donné quelle est distribuée. Notre procédure de maintenance permet le passage à l'échelle.

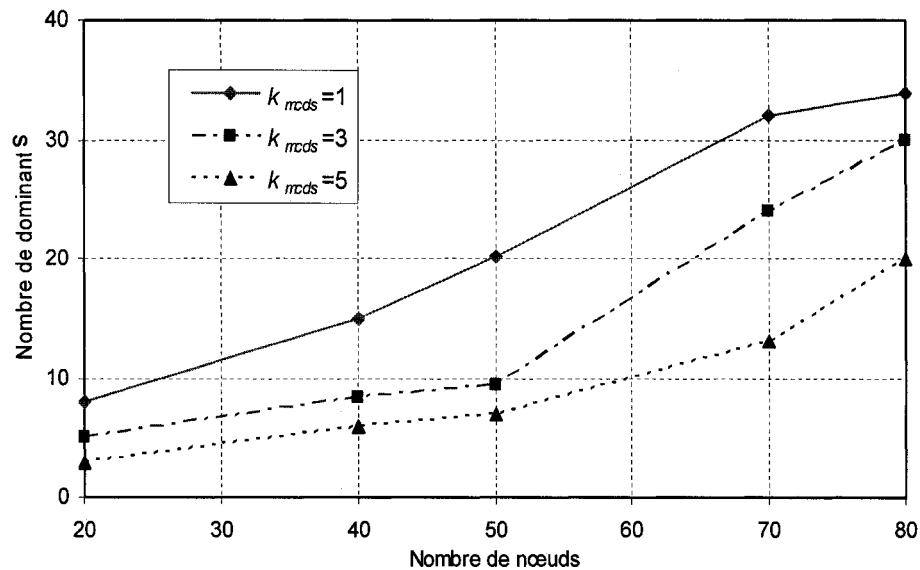


Figure 48 Nombre de dominants en fonction de la taille du réseau

D'après les courbes de la figure 48, nous remarquons que la taille de la dorsale, représentée par le nombre de dominants, croît proportionnellement avec la taille du réseau ce qui est tout à fait raisonnable. L'allure de cette croissance est presque identique à celle retrouvée dans le Chapitre 4, où nous n'avons ni mobilité ni maintenance.

5.4.2.3 Occurrence à l'état dominant

Dans cette section, nous voulons vérifier l'occurrence pour chaque nœud dans l'état dominant. Pour cela, nous allons déterminer le nombre de fois qu'un nœud passe à l'état dominant durant toute la simulation. Les paramètres de simulation sont les suivants : le réseau comprend 50 nœuds, $k_{c ds} = 1$, la vitesse moyenne pour chaque nœud est égale à $10m/s$ et la durée de simulation est d'une heure.

La figure 49 illustre le nombre de fois qu'un nœud devient dominant au cours de la simulation. Plus de 70 % des nœuds passent à l'état dominant entre 20 et 28 fois.. Ces résultats reflètent en quelque sorte qu'il y a un équilibre de la charge dans le réseau en

termes de consommation d'énergie car les nœuds dominants auront à consommer plus d'énergie. Rappelons que nous sommes partis avec l'hypothèse que tous les nœuds sont homogènes et qu'au départ ils possèdent le même niveau d'énergie. De plus, seulement les nœuds qui forment la dorsale (dominants) sont chargés de retransmettre le trafic de contrôle. Une grande partie de la bande passante dans un réseau *ad hoc* est utilisée par le trafic de contrôle [Corson et Macker (1999)].

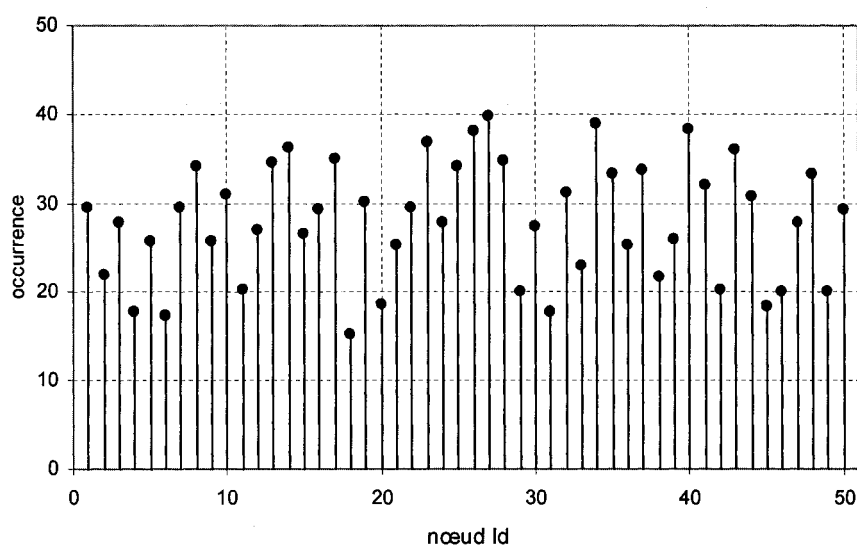


Figure 49 Nombre de fois qu'un nœud devient dominant

5.5 Comparaison avec d'autres algorithmes

Dans cette section, nous allons évaluer et comparer la performance de la procédure de maintenance proposée dans ce chapitre avec d'autres algorithmes proposés dans la littérature. Comme nous avons mentionné dans l'introduction, toutes les propositions utilisent un seul algorithme pour la construction et la maintenance. Nous allons reprendre les algorithmes décrits dans la section 2.5.3 (WCDS, CDS-based et B-CDS). Par la suite nous noterons MW, MC et MB pour désigner les algorithmes WCDS, CDS-based et B-CDS respectivement. La procédure proposée dans ce chapitre sera identifiée par M1. Pour mieux comparer notre procédure avec ces algorithmes en ce qui concerne la maintenance, nous ne considérons pas la phase d'établissement du réseau. En d'autres

termes, pour chaque simulation les statistiques ne seront prises qu'après la construction de la dorsale.

Dans le but de faire cette comparaison nous allons définir les deux critères d'évaluation suivants :

- **Taille de la dorsale** : c'est le nombre moyen des nœuds dominants. Nous examinons la variation de ce paramètre en fonction de la taille du réseau et la mobilité des nœuds. Rappelons que nous avons intérêt à garder une taille réduite: plus la taille de la dorsale est minimale meilleur est le résultat,
- **Durée de la maintenance** : c'est le temps moyen nécessaire pour effectuer la maintenance. Pour la procédure de maintenance proposée, ce paramètre est défini comme étant le délai moyen entre le moment où le nœud change de position (état IDLE) et le moment où le dernier nœud concerné par ce mouvement change son état à dominant ou dominé. Pour les autres algorithmes, c'est le temps moyen nécessaire pour qu'un nœud se connecte à la dorsale. Nous évaluons ce paramètre en fonction de la mobilité des nœuds,

5.5.1 Taille de la dorsale

La figure 50 illustre la variation de la taille de la dorsale (en %) en fonction de la taille du réseau. Dans cette simulation, la vitesse moyenne est égale à 15 *m/s*. Les résultats de simulations montrent bien que la procédure de maintenance proposée (M1) ainsi que les trois algorithmes (MC, MB et MW) possèdent un comportement similaire. En d'autres termes, la taille de la dorsale varie proportionnellement avec la taille du réseau. En effet elle est de l'ordre de 47%, 44%, 43% et 40% pour MC, MB, MW et M1 respectivement. Ces résultats montrent la nature distribuée des ces algorithmes. La maintenance se fait localement là où le changement de la topologie s'est produit.

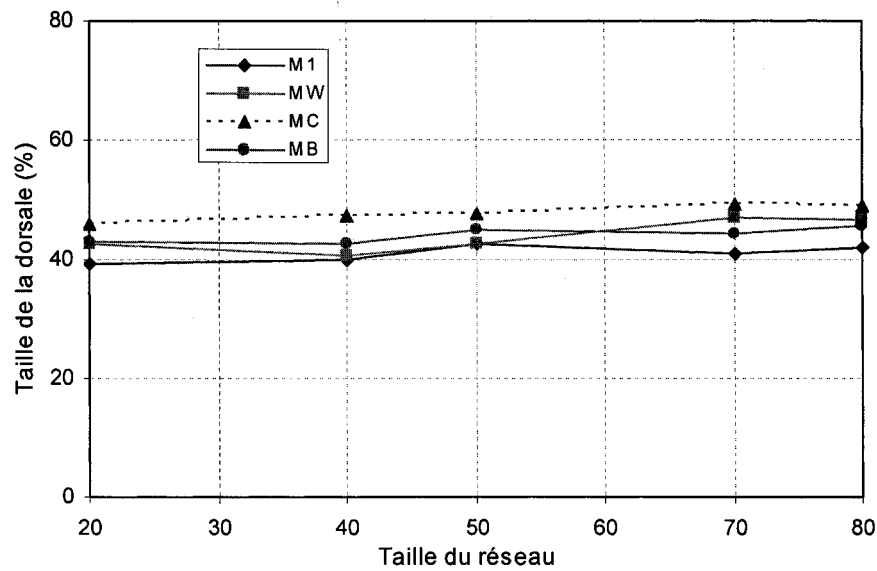


Figure 50 Taille de la dorsale en fonction de la taille du réseau

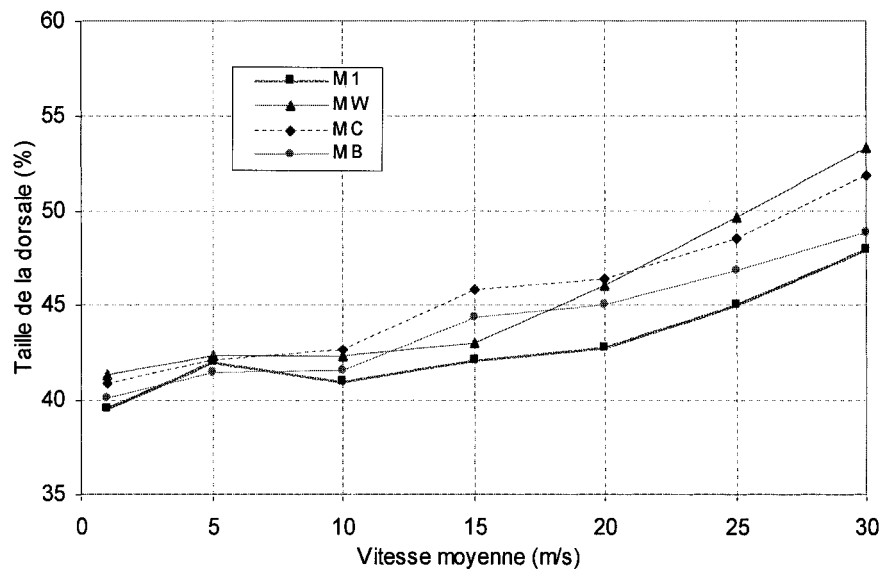


Figure 51 Taille de la dorsale en fonction de la vitesse moyenne des nœuds

La figure 51 illustre la variation de la taille de la dorsale (en %) en fonction de la vitesse moyenne des nœuds. Dans cette simulation, le nombre de nœuds dans le réseau est égal à 50 nœuds. La taille de la dorsale croît quand la mobilité des nœuds augmente. Lorsque

la dynamique augmente, le nombre de nœuds concernés par le changement de la topologie augmente également et par la suite la taille de la dorsale augmente. Pour une mobilité élevée, la procédure de maintenance M1 présente de meilleurs résultats. En effet, pour une vitesse moyenne de 30m/s , la dorsale possède une taille nettement inférieure avec la procédure M1 comparée aux autres algorithmes.

5.5.2 Durée de la maintenance

D'après les résultats de simulation, figure 52, le temps nécessaire pour effectuer la maintenance croît proportionnellement quand la vitesse moyenne des nœuds augmente. Ces résultats confirment ceux retrouvés dans la section précédente (figure 51). Plus la vitesse des nœuds augmente, plus le nombre de nœuds qui sont concernés, ayant subis un changement d'état, augmente ce qui augmente également le temps de maintenance augmente. L'algorithme M1 réalise la maintenance plus rapidement que les algorithmes MW, MC et MB.

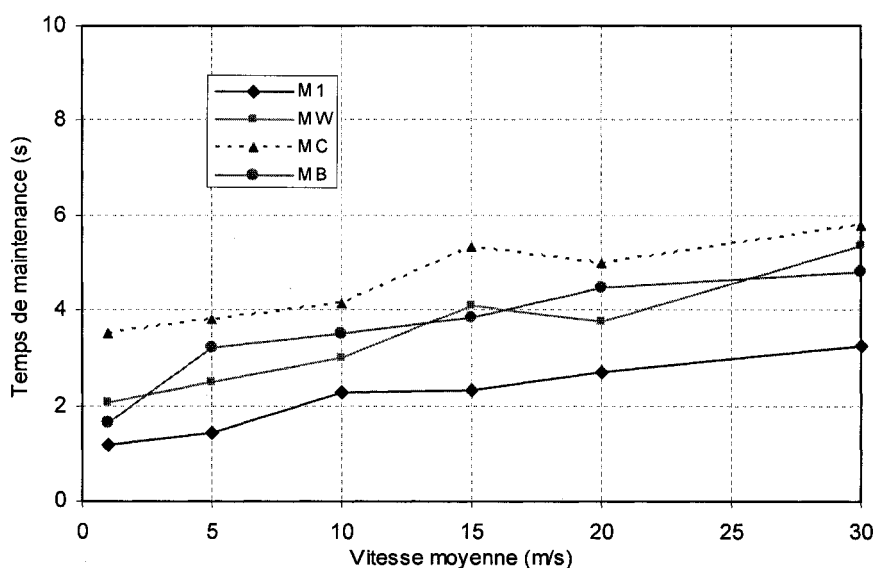


Figure 52 Temps de maintenance en fonction de la vitesse moyenne des nœuds

5.6 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle procédure de maintenance pour la dorsale virtuelle dans un réseau *ad hoc* mobile. Ensuite, nous avons procédé à l'évaluation de sa performance en fonction de la mobilité et de la taille du réseau. Cette procédure est distribuée et elle ne sera appliquée que s'il y a un changement dans la topologie. En d'autres termes, quand un terminal change de position il va tenter de se connecter au nœud appartenant à la dorsale le plus proche en appliquant la procédure de maintenance. Rappelons qu'une dorsale est construite dans la phase d'établissement en utilisant l'algorithme développé dans le chapitre 4. Cet algorithme garantit une dorsale de taille petite et par la suite la maintenance sera plus efficace.

Nous avons traité le cas général où une dorsale est construite à partir de k_{cds} . Dans ce cas, les résultats de simulation nous indiquent qu'il n'y a aucun avantage à utiliser un k_{cds} élevé. En effet, nous avons constaté que, quand k augmente, le temps nécessaire pour qu'un terminal se connecte à la dorsale augmente aussi. Toutefois, les performances en termes de pourcentage des nœuds connectés en fonction de la mobilité et de la taille du réseau sont invariables avec k_{cds} .

De nature distribuée, la procédure de maintenance proposée a démontré son efficacité non seulement au niveau de la variation de pourcentage de connexion en tenant compte de la mobilité, mais également au niveau de la mise à l'échelle (la variation du nombre des nœuds dans le réseau). En effet, cette procédure s'adapte bien avec la dynamique de la topologie au cours du temps et le pourcentage de connexions décroît légèrement quand la vitesse moyenne des terminaux augmente (92 % des terminaux restent connectés pour une vitesse moyenne de 30m/s), alors que ce pourcentage est quasiment constant quand le nombre de nœuds varie dans le réseau. Ces résultats montrent que la procédure de maintenance proposée assure une certaine stabilité de la structure globale du réseau quand la mobilité des nœuds augmente. De plus, nous avons vérifié qu'à long terme les nœuds auront quasiment la même chance de passer en état dominant ce qui

donne un certain équilibre de la charge et de l'énergie dissipé dans le réseau étant donné qu'un nœud dominant aura à relayer plus de trafic (contrôle et donné) qu'un nœud dominé.

La dernière partie de ce chapitre a été consacrée pour comparer la procédure de maintenance proposée dans ce chapitre avec d'autres algorithmes proposés dans la littérature. Les résultats de simulation montrent bien que notre procédure de maintenance proposée a une meilleure performance par rapport aux autres algorithmes. En effet, en utilisant notre procédure, nous obtenons toujours une dorsale de taille réduite dans un environnement dynamique tout en réduisant le temps de maintenance.

MCours.com