

## CHAPITRE 2

### LE ROUTAGE ET LES WMNS

#### 1 Introduction

Les protocoles de routage sont conçus essentiellement pour l'établissement et l'entretien des routes, pour que les messages soient correctement acheminés dans le réseau. Les caractéristiques des réseaux WMNs rendent l'utilisation des protocoles filaires habituels inadaptée. Ce chapitre introduit une généralité sur le routage récent et décrit les protocoles de routage les plus connus en réseau maillé sans fil

#### 2 Généralités sur le routage

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un chemin optimal des paquets à travers le réseau au sens d'un certain critère de performance (bande passante, délai etc.). Nous allons dans un premier temps faire un rapide rappel sur quelques notions essentielles de routage dans les réseaux avant de nous pencher plus particulièrement sur les protocoles de routage dans la section suivante.

##### 2.1 Classification des protocoles de routage

Les stratégies existantes utilisent une variété de techniques afin de résoudre ce problème. Suivant ces techniques, plusieurs classifications sont apparues, parmi lesquelles nous allons citer :

##### 2.1.1 Routage hiérarchique ou plat :

Le premier critère utilisé pour classer les protocoles de routage dans les réseaux ad hoc concerne le type de vision qu'ils ont du réseau et les rôles qu'ils accordent aux différents mobiles.

a-Les protocoles de routage à plat : considèrent que tous les nœuds sont égaux. La décision d'un nœud de router des paquets pour un autre dépendra de sa position. Parmi les protocoles utilisant cette technique, on cite l'AODV (Ad hoc On Demand Distance Vector)

b-Les protocoles de routage hiérarchique : fonctionnent en confiant aux mobiles des rôles qui varient de l'un à l'autre. Certains nœuds sont élus et assument des fonctions particulières qui conduisent à une vision en plusieurs niveaux de la topologie du réseau. Par exemple, un mobile pourra servir de passerelle pour un certain nombre de nœuds qui se seront attachés à lui. Le routage en sera simplifié, puisqu'il se fera de passerelle à passerelle, jusqu'à celle

directement attachée au destinataire. Dans ce type de protocole, les passerelles supportent la majeure partie de la charge du routage (les mobiles qui s'y rattachent savent que si le destinataire n'est pas dans leur voisinage direct, il suffit d'envoyer à la passerelle qui se débrouillera). Un exemple de protocole utilisant cette stratégie est l'OLSR (Optimized Link State Routing)

### **2.1.2 Le routage à la source et le routage saut par saut :**

a-Le routage à la source : le routage à la source ou << source routing >> consiste à indiquer dans le paquet routé l'intégralité du chemin que devra suivre le paquet pour atteindre sa destination. L'entête de paquet va donc contenir la liste des différents nœuds relayeur vers la destination. Le protocole le plus connu basant sur cette classe est : DSR (Dynamic Source Routing).

b-Le routage saut par saut : le routage saut par saut ou <<hop by hop>> consiste à donner uniquement à un paquet l'adresse du prochain nœud vers la destination. AODV fait partie des protocoles qui utilisent cette technique.

### **2.1.3 Etat de lien et Vecteur de distance :**

Autres classifications, hérité du monde filaire, est possible pour les protocoles de routage : les protocoles basés sur l'état des liens et sur le vecteur de distance. Les deux méthodes exigent une mise à jour périodique des données de routage qui doivent être diffusées par les différents nœuds de routage du réseau. Les algorithmes de routage basés sur ces deux méthodes, utilisent la même technique qui est la technique des plus courts chemins, et permettent à un hôte donné, de trouver le prochain hôte pour atteindre la destination en utilisant le trajet le plus court existant dans le réseau.

a-Les protocoles basés sur l'état de lien : La famille des protocoles à état de liens se base sur les informations rassemblées sur l'état des liens dans le réseau. Ces informations sont disséminées dans le réseau périodiquement ce qui permet ainsi aux nœuds de construire une carte complète du réseau. Un nœud qui reçoit les informations concernant l'état des liens, met à jour sa vision de la topologie du réseau et applique un algorithme de calcul des chemins optimaux afin de choisir le nœud suivant pour une destination donnée. En générale ces algorithmes se basent sur le principe de l'algorithme de Dijkstra pour calculer les chemins les plus courts entre un nœud source et les autres nœuds du réseau. Les principaux protocoles de routage dans les réseaux ad hoc qui appartiennent à cette classe sont les suivants : TORA, OLSR et TBRPF.

b-Les protocoles basés sur le vecteur de distance : Les protocoles à vecteur de distance se basent sur un échange, entre voisins, des informations de distances des destinations connues.

Chaque nœud envoie à ses voisins la liste des destinations qui lui sont accessibles et le coût correspondant. Le nœud récepteur met à jour sa liste locale des destinations avec les coûts minimums. Le processus de calcul se répète, s'il y a un changement de la distance minimale séparant deux nœuds, et cela jusqu'à ce que le réseau atteigne un état stable. Les calculs des routes se basent sur le principe de l'algorithme distribué de Bellman-Ford (DBF). Les protocoles de routage basés sur le vecteur de distance les plus connus pour les réseaux ad hoc sont : DSR, DSDV et AODV.

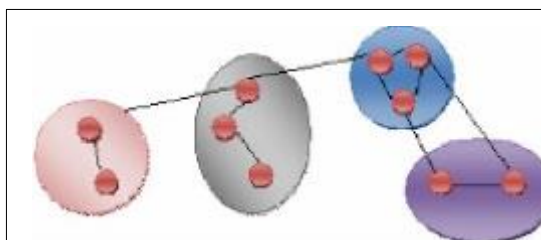
#### 2.1.4 L'inondation :

L'inondation ou la diffusion pure, consiste à répéter un message dans tous les réseaux. Un nœud qui initie l'inondation envoie le paquet à tous ses voisins directs, de même si un nœud quelconque de réseau reçoit le paquet pour la première fois, il le rediffuse à tous les voisins, Ainsi de proche en proche le paquet inonde le réseau.

Notons que les nœuds peuvent être anciens appliqués (durant l'inondation) certains traitements de contrôle dans le but d'éviter certains problèmes, tel que le bouclage et la duplication des messages, Le mécanisme d'inondation est utilisé généralement dans la première phase du routage plus exactement dans la procédure de découverte des routes, et cela dans le cas où le nœud source ne connaît pas la localisation exacte de la destination.

#### 2.1.5 Le concept de groupe :

Dans la communication de groupe, les messages sont transmis à des entités abstraites ou groupes, les émetteurs n'ont pas besoin de connaître les membres du groupe destinataire. La gestion des membres d'un groupe permet à un élément de se joindre à un groupe, de quitter ce groupe, se déplacer ailleurs puis rejoindre le même groupe. C'est en ce sens que la communication de groupe assure une indépendance de la localisation, ce qui la rend parfaitement basée sur les groupes. Le concept de groupe facilite les tâches de la gestion du routage (telles que les transmissions des paquets, l'allocation de la bande passante etc.) et cela en décomposant le réseau en un ensemble de groupes connectés.



**Figure 2.1** La décomposition du réseau en groupe

### 2.1.6 Protocoles uniformes et non-uniformes :

Certains protocoles de routage n'utilisent pas tous les nœuds d'un réseau pour faire transiter les messages, au contraire ils en sélectionnent certains, en fonction du voisinage ou pour former des cellules. Ces protocoles sont dits non-uniformes. Ceux qui utilisent tous les nœuds du réseau capables de router sont appelés protocoles uniformes.

### 2.1.7 La classification de MANET :

C'est la classification qui nous intéresse et qu'on maintient pour le prochain chapitre. Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en : **Proactif**, **Réactif** et **Hybride**.

#### a- Les protocoles de routage proactifs :

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 nœuds du réseau) au niveau de chaque nœud 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 de mise à jour des chemins. Le plus abouti de ces protocoles est OLSR.

#### Avantages et les inconvénients des protocoles proactifs :

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. Le problème est que, les changement de routes peuvent être plus fréquents que la demande de la route et le trafic induit par les messages de contrôle et de mise à jour des tables de routage peut être important et partiellement inutile, ce qui gaspille la capacité du réseau sans fil. De plus, la taille des tables de routage croît linéairement en fonction du nombre de nœud.

De ce fait, un nouveau type de protocole a apparu, il s'agit des protocoles de routage réactifs.

#### b- Les protocoles de routage réactifs :

Les protocoles de routage réactifs (dits aussi: 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 fils.

La majorité des solutions proposées pour résoudre le problème de routage dans les réseaux ad hoc, et qui sont évaluées actuellement par le groupe de travail MANET (Mobile Ad Hoc Networking working Groupe) de l'IETF (Internet Engineering Task Force), 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, le plus connu de ces protocoles est AODV.

Avantages et les inconvénients des protocoles réactifs:

A l'opposé des protocoles proactifs, dans le cas d'un protocole réactif, aucun message de contrôle ne charge le réseau pour 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 à une impossibilité de se connecter alors que le destinataire est bien là.

De ce fait, un nouveau type de protocole a apparu, il s'agit des protocoles de routage hybrides.

c- Les protocoles de routage hybrides :

Dans ce type de protocole, on peut garder la connaissance locale de la topologie jusqu'à un nombre prédéfini- a priori petit- de sauts par un échange périodique de trame de contrôle, autrement dit par une technique proactive. Les routes vers des nœuds plus lointains sont obtenues par schéma réactif, c'est-à-dire par l'utilisation de paquets de requête en diffusion. Un exemple de protocoles appartenant à cette famille est DSR (Dynamic Source Routing), qui est réactif à la base mais qui peut être optimisé s'il adopte un comportement proactif. Un autre exemple est le protocole ZRP (Zone Routing Protocol).

Avantages et inconvénient des protocoles hybrides :

Le protocole hybride est un protocole qui se veut comme une solution mettant en commun les avantages des deux approches précédentes en utilisant une notion de découpe du réseau.

Cependant, il rassemble toujours quelques inconvénients des deux approches proactives et réactives.

### **3 Les mécanismes dans WMNs de routage**

#### **3.1 OLSR (Optimized Link State Routing protocol)**

##### **3.1.1 Définition:**

C'est un protocole de routage proactif non uniforme dédié aux réseaux ad hoc inspiré de l'algorithme état des liens classique, développé dans le cadre du projet Hipercom de l'institut national de la recherche en informatique et algorithmique (INRIA) et proposé en tant que RFC (Request For Comment). Il a comme objectif de fournir des routes de plus court chemin

en termes de nombre de sauts. OLSR utilise les multipoints relais (MPR) pour retransmettre les messages diffusés au cours d'une inondation dans le but de réduire le nombre de messages envoyés, ce qui réduit par conséquent les frais. Tel que : lorsqu'un nœud MPR reçoit un message de diffusion, il traite et rediffuse le message. Par contre un nœud non MPR traite seulement le message. Par ailleurs, OLSR utilise le concept d'interface, tel qu'un nœud peut posséder plusieurs instances d'écoute, ce qui lui donne le comportement de plusieurs nœuds virtuels. C'est un protocole qui fonctionne mieux dans les réseaux denses et larges.

### 3.1.2 Notions de base :

- Nœud voisin à n-sauts : (w) est un voisin à n sauts de (u) si u peut communiquer avec (w) par l'intermédiaire de (n-1) nœuds.

Exemple : (u) ----- (v) : (v) est le voisin de (u) à un 1 saut.

(u) ----- (v) ----- (w) : (w) est le voisin de (u) à 2 sauts

(u) ----- (v) ----- (w) ----- (x) : (x) est le voisin de (u) à 3 sauts.

- Lien symétrique : (v) est un voisin de (u) par un lien symétrique si et seulement si (u) entend (v), et (v) entend (u).
- Lien asymétrique : (v) est un voisin de (u) par un lien asymétrique si et seulement si (u) entend (v), et (v) n'entend pas (u).
- L'ensemble de multipoints relais d'un nœud (u) : c'est un sous ensemble (idéalement le plus petit sous ensemble) de l'ensemble des nœuds voisin à 1 saut du nœud (u) élus pour atteindre la totalité des voisins à 2 sauts du nœud (u) par des liens symétriques.

L'exemple cité dans la Figure 2.2 montre les MPRs d'un nœud.

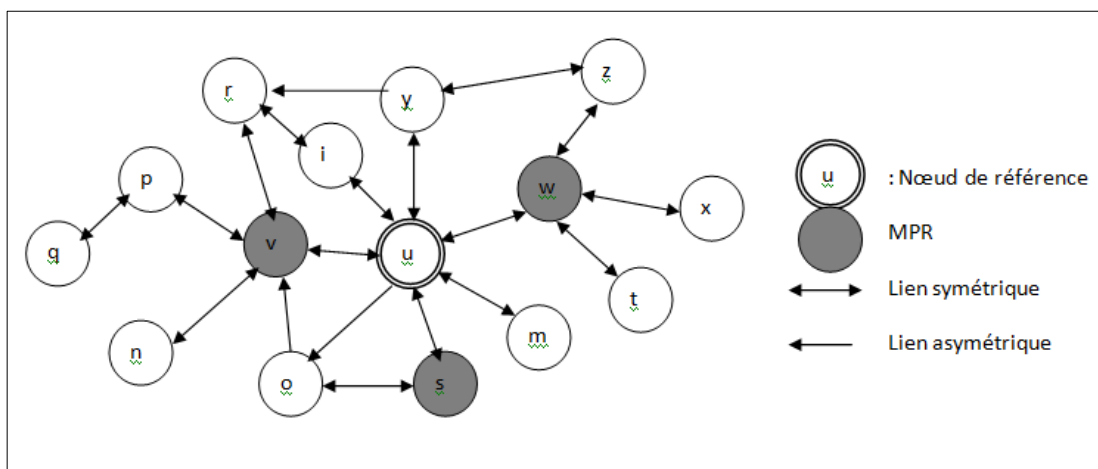


Figure 2.2 Ensemble des MPRs d'un nœud

Le choix des MPRs est un problème difficile, car cela revient à trouver un ensemble dominant dans un graphe, en plus le contexte dynamique qui implique le calcul de cet ensemble assez souvent. Donc il faut un algorithme efficace et rapide.

### Algorithme utilisé pour sélectionner les MPR d'un noeud

Soient :

(u) un noeud quelconque,  $MPR(u)$  : l'ensemble des MPR de u

$N(u)$  : l'ensemble des voisins de u ayant –au moins- un lien avec les voisins de deux sauts de u,

$N2(u)$  : l'ensemble des voisins à deux sauts de u

1- Initialiser  $MPR(u) = \{ \}$

2- Identifier l'ensemble  $N(u)$ ,

3- Tant que  $N2(u) \neq \{ \}$  faire

- Calculer le degré de chaque élément de  $N(u)$ , sachant que le degré d'un nœud v appartenant à  $N(u)$  = nombre de ses voisins appartenant à  $N2(u)$ .

- Choisir le noeud qui a un degré maximal et l'ajouter à  $MPR(u)$  puis éliminer ses voisins dans l'ensemble  $N2(u)$ .

❖ L'ensemble des sélecteurs multipoint relais d'un nœud

L'ensemble des sélecteurs multipoint relais  $MPR\ selector(u)$  = l'ensemble des voisins à un saut de u qui l'ont choisi comme MPR. Dans notre exemple :  $MPR\ selector(u) = \{v, s, m, w, y, i\}$  ;  $MPR\ selector(v) = \{u, p, n, r\}$  .

❖ Format d'un paquet OLSR

Dans OLSR un paquet est structuré comme suit :

Longueur du paquet		N° de séquence du paquet	} Entête du paquet
Type de message	Temps de validité	Taille du message	
Adresse de l'expéditeur			} Entête du message
TTL : nbre maximal de sauts	Nombre de sauts atteints	Numéro de séquence du message	
Message			
Type de message	Temps de validité	Taille du message	
Adresse de l'expéditeur			
TTL : nombre maximal de sauts	Nombre de sauts atteints	Numéro de séquence du message	
Message			
.			
.			

**Figure 2.3** Format d'un paquet OLSR

### 3.1.3 Messages échangés

Les bases d'informations sont obtenues et calculées sur la base de l'échange de quatre types de messages.

- Multiple Interface Déclaration (MID) :

Chaque nœud équipé de multiples interfaces, doit informer l'ensemble des nœuds du réseau de sa configuration par l'envoi périodique du message 'MID' qui contient l'ensemble de ses adresses interfaces (voir la Figure 2.4). Cette diffusion sert à l'établissement de la table d'association des interfaces multiples.

Longueur du paquet		Numéro de séquence du paquet
MID-MESSAGE	MID_HOLD_TIME	Taille du message
Adresse de l'expéditeur		
TTL=255	Nombre de sauts atteints	Numéro de séquence du message
Adresse interface		
Adresse interface		
...		

**Figure 2.4** Format d'un message MID

- Message Hello :

Chaque nœud informe ses voisins à un saut, l'état de son voisinage par l'envoi périodique du message 'Hello' qui contient la liste de ses liens sauf le lien du nœud destination (voir la Figure 2.5). L'échange périodique du message 'Hello' et les informations de la table d'association des adresses multiples servent à mettre à jour les tables : Ensemble des voisins, Table de voisinage et l'ensemble de voisinage à deux sauts, qui servent à leur tour de calculer l'ensemble des MPRs et l'ensemble des MPRs Selectors.



Longueur du paquet		Numéro de séquence du paquet	
HELLO_MESSAGE	NEIGHB_HOLD_TIME	Taille du message	
Adresse de l'expéditeur			
TTL =1	Nombre de sauts atteints	Numéro de séquence de message	
Réservé		Intervalle d'émission	Volonté
Code lien	Réservé	Taille du message lien	
Adresse de l'interface du nœud voisin1			
Adresse de l'interface du nœud voisin2			
Adresse de l'interface du nœud voisin2			
...			
Code lien	Réservé	Taille du message lien	
Adresse de l'interface du nœud voisin			
Adresse de l'interface du nœud voisin			
....			
.....			

**Figure 2.5** Format d'un message Hello

- Message Topology Control (TC) :

Les messages 'TC' sont diffusés dans la totalité du réseau par l'inondation optimisée par les MPRs. Ils servent à calculer les routes vers n'importe quelle destination. Un message de contrôle contient dans sa partie données un numéro de séquence incrémenté à chaque changement de topologie, il indique la fraîcheur des informations diffusées, ainsi que la liste des voisins du nœud émetteur identifiés par leur adresse principale, cette ensemble inclus son ensemble des MPR selectors (voir la Figure 2.6).

Longueur du paquet		Numéro de séquence du paquet	
TC_MESSAGE	TOP_HOLD_TIME	Taille du message	
Adresse de l'expéditeur			
TTL=255	Nombre de sauts atteints	Numéro de séquence du message	
ANSN		Réservé	
Adresse principale voisin 1			
Adresse principale voisin 2			
Adresse principale voisin 3			
.....			

**Figure 2.6** Format d'un message TC

- Message HNA (Host and Network Association) :

Chaque nœud possédant une interface d'un autre réseau n'implémentant pas le protocole OLSR (joue le rôle d'une passerelle) génère un message 'HNA' qui sera diffusé dans tout le réseau, pour annoncer sa configuration.

Longueur du paquet		Numéro de séquence du paquet
HNA_MESSAGE	HNA_HOLD_TIME	Taille du message
Adresse l'expéditeur		
TTL=255	Nombre de sauts atteints	Numéro de séquence du message
Adresse réseau		
Masque réseau		
Adresse réseau		
Masque réseau		
...		

**Figure 2.7** Format d'un message HNA

### 3.1.4 Les bases d'informations

OLSR maintient dans chaque nœud une image locale et globale de la topologie du réseau, ceci nécessite le stockage et la mise à jour plusieurs bases d'informations dans la mémoire de chaque nœud.

- Table d'association des interfaces multiples :

Un nœud peut être équipé de plusieurs interfaces, chaque interface est référenciée par une adresse, donc un nœud peut posséder plusieurs adresses. Mais il est identifié par une seule adresse appelée **adresse principale**. La table d'association des interfaces multiples fait l'association entre les interfaces et l'adresse principale pour chaque destination.

Adresse de l'interface du nœud	Adresse principale de ce nœud	Heure d'expiration

**Table 2.1** Table d'association des interfaces multiples

- L'ensemble des voisins :

Cette table contient pour chaque interface du nœud, la liste des interfaces voisines avec une information sur l'état du lien. A l'expiration de l'enregistrement, il doit être supprimé.

Adresse de l'interface du nœud local	Adresse de l'interface du nœud voisin	Temps ou le lien est considéré asymétrique	Temps d'expiration de l'enregistrement

**Table 2.2** Ensemble des voisins

- Table de voisinage :

Cette table contient la liste des voisins du nœud local identifiés par leur adresse principale avec une information sur l'état du lien et leur volonté de participation dans le routage, celle-ci est mesurée par un entier compris entre 0 et 7.

Adresse principale d'un voisin	Etat de lien	volonté

**Table 2.3** Table de voisinage

- Ensemble de voisinage à deux sauts :

Cette table contient la liste des voisins symétriques du nœud local à deux sauts. Ainsi que le voisin intermédiaire (tous identifiés par leur adresse principale). Le temps d'expiration précise la date et l'heure à laquelle expire l'enregistrement et doit être enlevé.

Adresse principale du voisin à 1 saut	Adresse principale du voisin à 2 sauts	Temps d'expiration

**Table 2.4** Ensemble de voisinage à deux sauts

- Ensemble des MPRs :

Chaque nœud maintient l'ensemble de ses MPRs identifiés par leur adresse principale.

MPR

**Table 2.5** Ensemble des MPRs

- Ensemble des MPR Selectors :

Chaque nœud enregistre une série d'enregistrements MPR selectors décrivant les voisins qui l'ont choisi comme MPR.

Adresse principale du nœud MPR selector	Temps d'expiration

**Table 2.6** Ensemble des MPR Selectors

- La table topologique :

Cette table donne une vision globale du réseau, elle contient pour chaque destination au moins un enregistrement contenant l'adresse principale de la destination, un MPR qui permet l'accès à cette destination, plus un numéro de séquence et un temps d'expiration.

Adresse principale d'une destination	Dernier MPR de la destination	Numéro de séquence	Temps d'expiration

**Table 2.7** La table topologique

- La table de routage :

Chaque nœud maintient une table de routage qui permet de router les paquets pour une communication entre deux nœuds du réseau. La table est recalculée pour mettre à jour son contenu par rapport aux changements produits dans le réseau.

Destination	Suivant	Distance	Interface

**Table 2.8** La table de routage OLSR

Un enregistrement de cette table est interprété comme suit : Un nœud "**Destination**" est atteint par un itinéraire composé d'un nombre de sauts mesuré à "**Distance**", tel que le nœud du saut suivant dans l'itinéraire est "**Suivant**", qui est un voisin symétrique du nœud local accessible par l'interface "**Interface**".

- Ensemble des associations :

Chaque nœud enregistre une table contient des informations sur les passerelles du réseau. Un enregistrement de cette table est composé de : l'adresse de la passerelle, l'adresse du réseau, le masque du réseau et le temps d'expiration. Cette table est établie sur la base de l'échange périodique du message 'HNA'.

Adresse de la passerelle	Adresse du réseau	Masque du réseau	Temps d'expiration

**Table 2.9** Ensemble des associations

### 3.1.5 Principe de fonctionnement :

OLSR est un protocole Proactif qui utilise la méthode état des liens, basé sur l'inondation effectuée par le mécanisme de diffusion optimisé par les MPRs. OLSR est composé d'un noyau qui assure le fonctionnement de base (fournir des routes de plus court chemin), ainsi d'un ensemble de fonctions auxiliaires.

- Fonctionnement de base :

Le module fonctionnement de base consiste à assurer le routage dans les réseaux ad hoc. Le routage passe par les étapes suivantes :

- a. Annonce des interfaces multiples :

Chaque nœud doté des multiples interfaces annonce périodiquement des informations concernant sa configuration par l'inondation du message 'MID', cette inondation est optimisée par le mécanisme des MPR. Les informations diffusées sont enregistrées dans la table d'association des interfaces multiples de chaque nœud du réseau.

- b. Découverte de voisinage :

Chaque nœud envoie régulièrement un message 'Hello' vers ses voisins à un saut (TTL=1) sur les interfaces par lesquelles la connexion est activée, contenant la liste de ses voisins qu'il entend, et aussi la liste des nœuds entendus par ceux-ci, également ce nœud reçoit les messages 'Hello' de ses voisins à un saut. Cela permet de savoir l'état des liens de ses voisins à un saut (symétrique ou asymétrique), ainsi d'avoir une vision à deux sauts. En plus, les informations concernant l'ensemble des voisins à un saut identifiés par adresse interface sont enregistrées dans la table ensemble des voisins. Les informations concernant le voisinage à un saut identifiées par l'adresse principale sont enregistrées dans la table de voisinage. Les messages 'Hello' échangés permet de savoir l'état des liens des voisins à deux sauts, ces informations sont stockées dans la table Ensemble de voisinage à deux sauts.

c. Sélection des MPR :

Sur la base de la table Ensemble de voisinage à deux sauts, chaque nœud calcule et élite la liste de ses MPRs, en utilisant l'algorithme de calcul des MPRs cité à l'avance. La liste des MPRs est enregistrée dans la table Ensemble des MPRs. Cette liste est annoncée aux voisins grâce à des messages 'Hello'. Donc chaque nœud peut savoir les nœuds qui l'ont choisi comme MPR, et enregistre cet ensemble dans la table Ensemble des MPR selectors.

d. Découverte de la topologie

Chaque nœud MPR diffuse régulièrement des messages 'TC' (topologie control) dans tout le réseau, en utilisant l'inondation par multipoint relais. Le message 'TC' contient pour un nœud MPR donné la liste des MPR selectors. Sur la base des messages 'Hello' et des messages 'TC', chaque peut maintenir sa table topologique.

e. Calcul de la table de routage

Chaque nœud possède une table de routage contient pour chaque destination du réseau le tuple suivant : (destination, suivant, distance, interface) indiquant pour chaque destination le prochain saut à effectuer et la distance entre le nœud local et la destination mesurée en nombre de sauts. La table de routage est établie sur la base des informations contenants dans la table de voisinage et la table topologique, en exécutant l'algorithme suivant :

1- Détruire toutes les entrées ultérieures de la table de routage

2- Insertion dans la table de routage tous les voisins symétriques à un saut (la distance=1)

3- Boucle pour  $h$  initialisé à 1 et incrémenté de 1 à chaque étape, jusqu'à ce qu'il n'y ait plus de nouvelles entrées dans la table de routage. A chaque étape, on insère les destinations à  $(h+1)$  sauts.

- Pour chaque tuple (destination, dernier, n° de séquence, temps d'expiration) dans la table topologique, si 'destination' n'est pas une entrée dans la table de routage, et si 'dernier' correspond à une entrée de la table de routage (dernier, suivant, h) alors on insère une nouvelle entrée dans la table de routage (destination, suivant,  $h+1$ , interface).

- Fonctionnement auxiliaire :

Le module fonctionnement auxiliaire est composé d'un ensemble de fonctions assurant des fonctionnalités supplémentaires qui peuvent être applicables dans des scénarios précis, exemple :

- La connectivité avec des domaines de routage non OLSR.
- La détection avancée des liens.
- La propagation de données topologiques redondantes.

- L'inondation redondante,...

## 3.2 Le protocole Ad Hoc On Demand Distance Vector AODV :

### 3.2.1 Définition:

AODV est un protocole de routage réactif multi sauts basé sur l'algorithme de **Bellman-Ford**, destiné à être utilisé par des nœuds mobiles dans un réseau ad hoc. Proposé par les membres du groupe de travail MANET, spécifié dans le RFC 3561, inspiré du protocole DSDV par l'amélioration de son algorithme de routage. C'est un protocole qui découvre les routes demandées par la technique des numéros de séquence pour éviter les boucles de routage et le comptage à l'infini, ainsi que pour maintenir la consistance des informations et de garder les routes les plus fraîches, en conservant les chemins d'une façon distribuée dans une table de routage stockée dans chaque nœud.

### 3.2.2 Les messages échangés

L'ensemble des flux échangés dans AODV sont de type UDP, envoyés sur le port 654, un flux peut être l'un des messages suivants :

- Demande de route (Route REQuest RREQ)

C'est un message diffusé en mode broadcast lors de la recherche d'une nouvelle route, ou de la maintenance d'une route cassée.

Type=1	J	R	G	D	U	Réservé	Nombre de sauts
ID RREQ							
Adresse IP destination							
Numéro de séquence destination							
Adresse IP source							
Numéro de séquence source							

**Figure 2.8** Format d'un message RREQ

Tel que :

J : joignez le pavillon, réservé pour le multicast.

R : réparation du pavillon, réservé pour le multicast.

G : indique si une Réponse de route 'RREP' générée par un nœud intermédiaire devrait être envoyée vers la destination spécifiée dans 'RREQ'.

D : indique que seule la destination peut répondre au message 'RREQ'.

Nombre de sauts : le nombre de sauts entre le nœud source et le nœud qui traite le message 'RREQ'.

ID RREQ : identifiant de la demande, il est obtenu par la combinaison de l'adresse IP de la source et d'un numéro de séquence.

Numéro de séquence destination : le dernier numéro de séquence reçu dans le passé du nœud source pour toute route vers la destination.

Numéro de séquence source : le numéro de séquence en cours pour être utilisé sur la route désirée.

- Réponse de route (Route REPlY RREP) :

C'est un message envoyé vers la source pour l'informer qu'il existe un chemin vers la destination en question.

Type=2	R	A	Réservé	Taille préfixe	Nombre de sauts
Adresse IP destination					
Numéro de séquence destination					
Adresse IP source					
Durée de vie					

**Figure 2.9** Format d'un message RREP

Tel que :

R : réparation du pavillon, utilisé pour le multicast.

A : une reconnaissance nécessaire par un accusé de réception.

- Erreur de route (Route ERRor RERR) :

C'est un message envoyé vers la source par le nœud qui détecte un lien brisé dans une route active.

Type =3	N	Réservé	Nombre de dest
Adresse IP de destination inaccessible			
Numéro de séquence de destination inaccessible			
Adresse IP de destination inaccessible additionnelle			
Numéro de séquence de destination inaccessible additionnel			

**Figure 2.10** Format d'un message RERR

Tel que

N : indique que la route a été réparée localement.

Nombre de dest : c'est le nombre de destinations concernées par le lien brisé.

- Accusé de réception de réponse de route (Route REPlY ACKnowledgment RREP-ACK) :



C'est un message envoyé comme accusé de réception du message 'RREP' avec la présence du bit A. Cela se fait habituellement lorsqu'il y a un danger de liens unidirectionnels.

<b>Type=4</b>	<b>Réservé</b>
---------------	----------------

### 3.2.3 Les bases d'informations :

Malgré que la recherche des routes soit réactive dans AODV, chaque nœud doit stocker et maintenir des informations pour le fonctionnement du protocole AODV.

- La table de routage

C'est la table de base, une entrée de cette table contient essentiellement :

@IP dest : l'adresse IP de la destination.

N°seq dest : le numéro de séquence destination qui permet d'éviter les boucles de routage.

Nbre de sauts : c'est la distance entre le nœud local et la destination mesurée en nombre de sauts.

Suivant : l'adresse du nœud qui représente le premier pas dans l'itinéraire.

Liste des précurseurs actifs : liste des nœuds originaires des transmissions vers la destination qui passent par le nœud local.

Temps d'expiration : temps au bout duquel l'entrée est expirée et doit être supprimée.

Tampon de requête : utilisé pour assurer qu'une seule réponse 'RREP' est envoyée par requête 'RREQ'.

<b>@ IP dest</b>	<b>N° seq dest</b>	<b>Nbre de sauts</b>	<b>Suivant</b>	<b>Liste précurseurs actifs</b>	<b>Temps d'expiration</b>	<b>Tampon de requête</b>

**Table 2.10** La table de routage AODV

- La table d'historique :

C'est une table qui enregistre le traitement des 'RREQ'. Une entrée de cette table identifie d'une manière unique le paquet 'RREQ' pour savoir si cette requête a déjà été vue et traitée ou non.

@source	ID requête

Table 2.11 Table d'historique

### 3.2.4 Principe de fonctionnement :

AODV utilise le principe des numéros de séquence afin d'éviter les boucles infinies et de maintenir la consistance des informations de routage. Le fonctionnement du protocole AODV est basé sur deux mécanismes : Découverte de routes et Maintenance de routes.

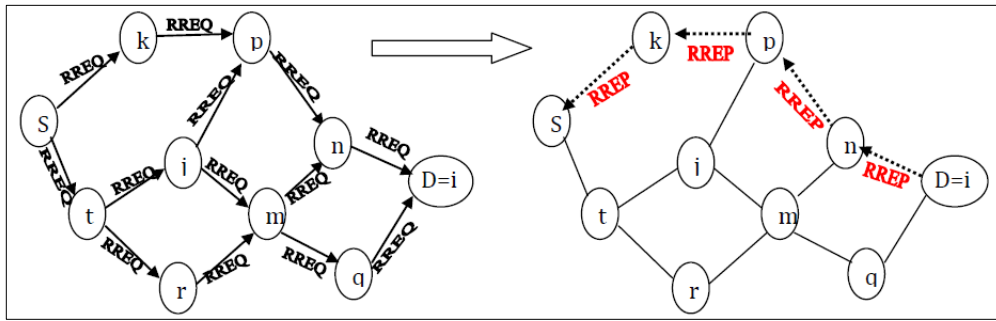
#### ❖ Découverte de routes :

Lorsqu'un nœud (S) a besoin de connaître la route qui mène vers une destination donnée (D), il consulte sa table de routage :

- Si le chemin est connu (existence d'une route fraîche) : il commence l'envoi des paquets directement.

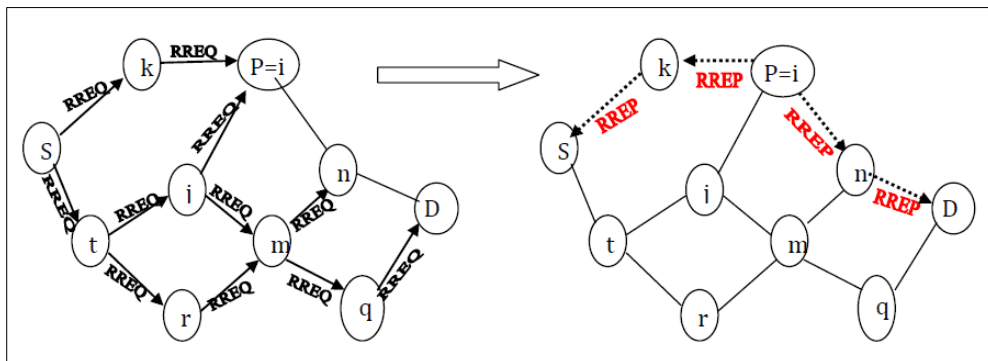
- Sinon : destination inconnue ou chemin expiré : dans ce cas le nœud source (S) diffuse un message demande route 'RREQ', ensuite il attend la réponse. Le nœud source utilise une technique d'expansion d'anneau afin d'éviter les déplacements inutiles des paquets 'RREQ' dans la totalité du réseau. Cette technique consiste à commencer la recherche avec un nombre de sauts initial et un délai correspond. Si la recherche est infructueuse, l'opération sera relancée avec augmentation du nombre de sauts (respectivement le délai d'attente). Cette opération est répétée jusqu'à obtention d'une réponse de route 'RREP' ou dépassement du nombre de sauts maximum, c'est le diamètre du réseau, dans ce cas : la destination est déclarée injoignable. Lorsqu'un nœud intermédiaire (i) reçoit un message 'RREQ', il vérifie dans sa table historique si la requête existe (déjà traitée), dans ce cas la requête sera ignorée. Sinon il inscrit l'identifiant de la requête dans sa table historique, ensuite il met à jour sa table de routage par l'enregistrement des paramètres de routage de la route qui mène vers la source (S), cette route est utilisée comme chemin inverse pour la réponse de route 'RREP', puis il passe au traitement de la requête 'RREQ', on distingue deux cas :

**Cas1** : le nœud (i) est lui-même la destination désirée, dans ce cas il compare son propre numéro de séquence avec celui du 'RREQ' : s'ils sont égaux alors il incrémente son numéro de séquence, sinon il ne le change pas. Ensuite il envoie la réponse de route 'RREP'. Remarque : ce cas ne doit se produire que très rarement puisque le voisin de la destination appartenant à la route connaît le chemin vers la destination, donc c'est lui qui génère la 'RREP'.



**Figure 2.11** Exemple de découverte de route avec RREP générée par la destination

**Cas2 :** le nœud (i) vérifie dans sa table de routage, l'existence d'un chemin fraîche vers la destination. Si oui alors il génère une réponse de route 'RREP', puis il copie un numéro de séquence connu de la destination dans le champ "numéro de séquence destination" de la 'RREP' avant de l'envoyer vers la source. De plus, le nœud intermédiaire (i) informe le nœud destinataire qu'il est sollicité par un nœud (S), à l'aide d'une 'RREP' car celui-ci ne connaît pas la route pour aller vers le nœud source(S), (en réalité ceci est réalisé uniquement si le bit 'G' pour gratuitous est présent dans la 'RREQ').



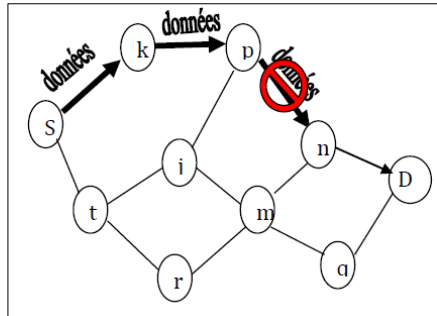
**Figure 2.12** Exemple de découverte de route avec RREP générée par un nœud intermédiaire

Sinon, (le chemin vers la destination est encore inconnu ou expiré) alors le nœud (i) compare le numéro de séquence enregistré dans sa table de routage avec celui de la 'RREQ', puis il garde le maximum, puis il incrémente le nombre de sauts de la 'RREQ' ensuite il rediffuse de sa part le paquet 'RREQ'.

#### ❖ Maintenance des routes

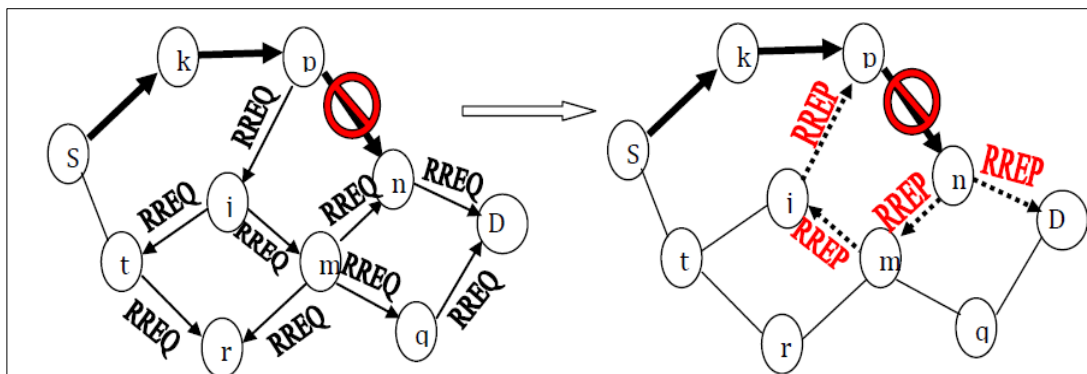
Le mouvement des nœuds appartenant à une route active donnée peut causer une défaillance dans les liens composants la dite route. Pour cela AODV offre une maintenance des routes actives aussi longtemps possible, cette maintenance est réalisée par les étapes suivantes :

a- Détection des défaillances : chaque nœud appartenant à un itinéraire actif vérifie la connectivité de ses voisins appartenant aussi à un itinéraire actif par l'envoi périodique du message 'Hello'. S'il compte trois messages 'Hello' consécutifs sans aucune réponse, alors le lien est considéré défaillant.



**Figure 2.13** Exemple de détection d'une rupture d'un lien actif

b- Tentative de réparation locale : le nœud qui détecte une rupture dans un itinéraire actif essaie de réparer le lien localement, pour cela il incrémente le numéro de séquence de la destination puis il diffuse une 'RREQ' pour cette destination, puis il attend une 'RREP'. Pendant la tentative de réparation, les paquets de données doivent être tamponnés. Si le nœud de réparation réussit sa mission, il achève l'envoi des paquets sans informer le nœud source, sinon il procède à l'étape (c).

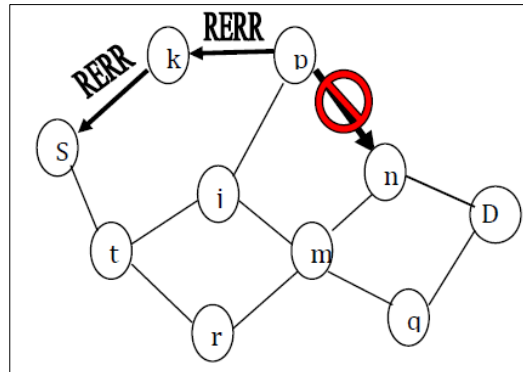


**Figure 2.14** Exemple de réparation locale avec succès

c- Annonce d'une erreur dans la route :

Le nœud qui détecte la rupture, commence par l'établissement de la liste des destinations inaccessibles par le lien perdu, à l'aide de sa table de routage, puis il diffuse un message

d'erreur 'RERR' vers les nœuds sources qui ont sollicités ces destinations, la diffusion est débutée par l'envoi du 'RERR' vers les précurseurs actifs pour chaque destination



**Figure 2.15** Exemple d'annonce d'une erreur de route

### 3.3 BATMAN (Better Approach TO Mobile Ad-hoc Networks)

BATMAN est né d'une réponse aux lacunes du protocole OLSR. La communauté Freifunk à Berlin basé sur le protocole OLSR a remarqué que OLSR avait beaucoup lacunes de performance lorsque le réseau a grandi. Ces lacunes sont en raison des routes inutiles dans les tables de routage et qui ont causé des boucles de routage.

Il y a eu une prise de conscience qu'un algorithme de routage pour un réseau maillé grand et statique doit être développé à partir des premiers principes et en conséquence le BATMAN projet a été lancé.

#### 3.3.1 Principe de fonctionnement :

Dans BATMAN tous les nœuds diffuse périodiquement des paquets hello, aussi connu comme originator messages OGMs, à ses voisins. Chaque OGM se compose d'une adresse de l'expéditeur, l'adresse de nœud qui l'envoi et un numéro de séquence unique. Chaque voisin change l'adresse d'envoi à sa propre adresse et rediffuse le message. En recevant son propre message l'originator fait une vérification de la liaison bidirectionnelle pour vérifier que le lien détecté peut être utilisé dans les deux sens. Le numéro de séquence est utilisé pour vérifier la fraîcheur du message. BATMAN ne maintient pas le chemin complet jusqu'à la destination, chaque nœud depuis le début de chemin ne maintient que les informations sur le lien suivant par lequel vous pouvez trouver le meilleur route.

#### 3.3.2 Modèle de système

Un réseau est modélisé en tant que  $G = (N, E)$ , où  $N$  représente un ensemble de nœuds et  $E$  représente un ensemble de liens entre les nœuds paires. Pour chaque nœud  $i \in N$  dans Batman, il existe un ensemble de voisins d'un seul saut,  $K$ . Le message provenant d'une

source  $s \in N$  vers une destination  $d$  est transmis le long d'une liaison  $(s, d) \in E$  si  $d$  est également un élément de  $K$  sinon il est transmis le long d'un route multi-hop composé d'un lien  $(s, i)$  et une route  $[i, j]$ , où  $i$  est un nœud dans  $K$  et  $(s, i)$  est un lien dans  $E$ . Le parcours  $[i, j]$  représente un itinéraire à partir du nœud  $i$  au nœud  $d$  via un sous-réseau  $S = (N - \{s\}, E - \{(s, i): i \in K\})$ .

### 3.3.3 Objectif de routage

L'objectif est de maximiser la probabilité à délivrer un message. BATMAN ne cherche pas à vérifier la qualité des chaque lien, il vérifie simplement son existence. Les liens sont comparées en termes de nombre de messages qui ont été princps reçue à l'intérieur de la fenêtre glissante courante.

### 3.3.4 Algorithme

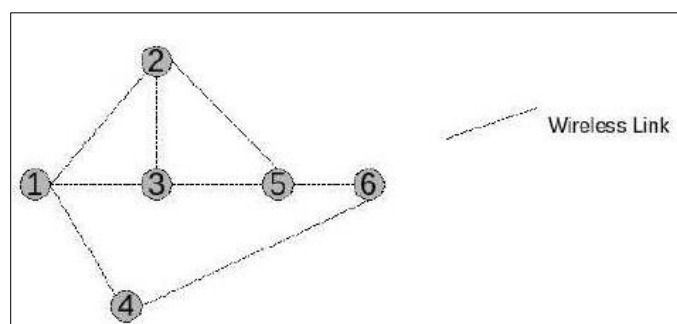
Étape 1 Considérons le routage de messages  $m$  de  $s$  à  $d$  sur le réseau  $G$ . Éliminer tous les liens  $(s, i) \forall i \neq K$  pour réduire le graphe.

Étape 2 associer chaque lien avec le poids  $W_{si}$  où  $W_{si}$  est le nombre de OGMs reçus à partir de la destination à travers le nœud voisin  $i$  au sein du la fenêtre glissante en cours.

Étape 3 Trouver le lien avec la plus grande  $W_{si}$  dans le sous-graphe et envoyer  $m$  sur du lien  $(s, i)$ .

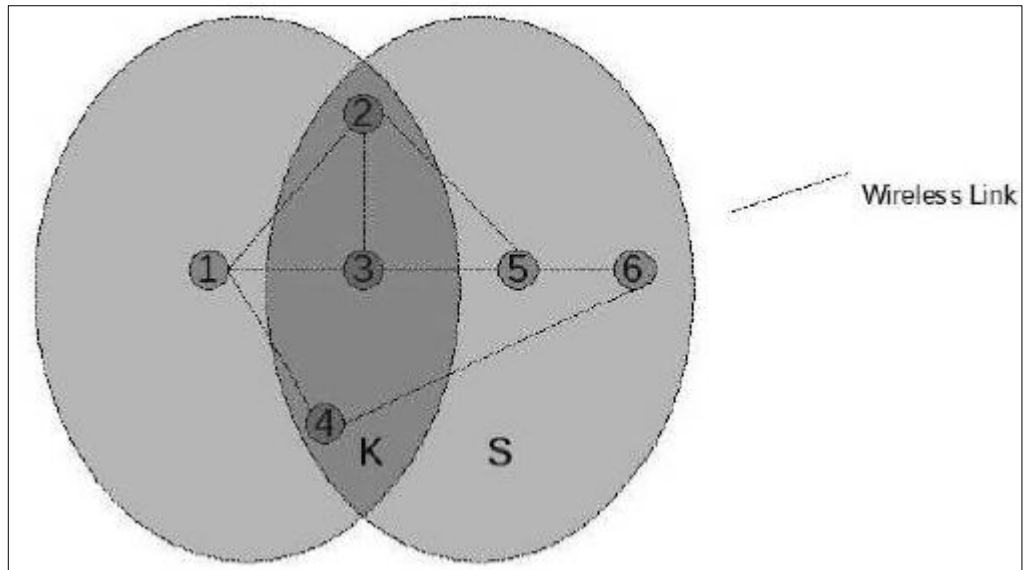
Étape 4 Si  $i \neq d$  Répétez les étapes 1 à 4 pour router le message à partir de  $i$  à  $d$  dans le sous-graphe  $S$ .

Les figures suivantes illustres le fonctionnement de BATMAN algorithme pour le scénario suivant:



**Figure 2.16** graphe initial  $G$

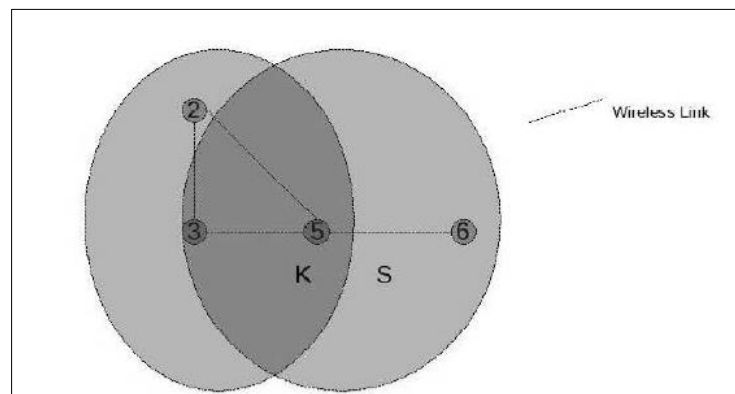
- Le nœud 1 veut envoyer un message à noeud 6. Il ne prend en compte que cet ensemble de liens  $\{(1, 2), (1, 3), (1, 4)\}$  à ses voisins  $\{2, 3, 4\}$ . Les ensembles correspondants sont illustrés à la Figure 2.17



**Figure 2.17** Sous-ensembles de nœuds formés par algorithme Batman dans la 1ère itération.

Il montre la relation entre les trois sous-ensembles qui sont visées à l'algorithme au-dessus

- Déterminer le meilleur lien qui a le plus grand nombre de OGMs reçus du nœud 6.
- Supposons que (1, 2) est le meilleur lien, alors envoyer le message suivant ce lien.
- Comme le nœud 2 n'est pas la destination, réduire le graphe N au graphe S et répétez les étapes 1 à 4 de l'algorithme. Cette est illustré à la Figure 2.18.



**Figure 2.18** Sous-ensembles de nœuds formés par algorithme BATMAN dans la 2<sup>e</sup> itération. Il

montre la relation entre les trois sous-ensembles qui sont visées à l'algorithme au-dessus

- Le Noeud 2 ne considère que cet ensemble de liens  $\{(2, 3), (2, 5)\}$  pour ses voisins  $\{3, 5\}$ .
- Déterminer le meilleur lien qui a le plus grand nombre de OGMs reçus du nœud 6

- Supposons que (2, 5) est le meilleur lien, puis envoyer le message suivant ce lien.
- Depuis Node 5 n'est pas la destination, de réduire le graphe N au graphe S et répétez les étapes 1 à 4 de l'algorithme.
- Nœud 5 ne considère que cet ensemble de liens {(5, 6), (5, 3)} pour ses voisins {6, 3}.
- Déterminer le meilleur lien qui a le plus grand nombre de OGMs reçus du nœud 6
- Supposons que (5, 6) est le meilleur lien, puis envoyer le message suivant ce lien.
- Nœud 6 est la destination.

#### **4 Conclusion :**

Après avoir défini l'environnement mobile maillé et décrit ses principales applications et caractéristiques, nous avons parlé dans ce chapitre du problème d'acheminement des paquets dans ce type de réseaux, c'est à dire le problème de routage. Le routage est un service très important dans les environnements mobiles, surtout quand il n'y a pas d'infrastructure qui s'occupe de l'acheminement des données. Dans le chapitre qui convient nous essayerons de mettre des mesures et un environnement pour comparer ces mécanismes.