



# La mobilité dans les Réseaux sans fils



Mobility in  
Wireless Network



# Sans Fil (1)

---

✱ Aucun câble ne relie les équipements  
-> utilisation des ondes ....

✱ Utilisation de bandes de fréquence

◆ 2 organismes en charge de la normalisation des fréquences

- USA : IEEE (Institute of Electrical and Electronics Engineers)
- Europe : ETSI (European Telecommunication Standards Institute)



# La fonction routage pour les réseaux sans fils

- 
- Quelques rappels sur le routage
  - Différents types de routage
    - routage proactif
    - routage réactif
    - routage géographique
-

# Routage

---

## ✧ Routage statique

- ✧ convient uniquement pour des sites de taille modeste.
- ✧ généralement, le routage est modifié après la découverte du problème.
- ✧ architecture géographique très peu flexible.

## ✧ Routage dynamique

- ✧ *indispensable dès que la topologie est complexe ou qu'elle change au cours du temps*
- ✧ algorithme en général auto-cicatrisant  
(si lien coupé, on recherche un autre passage)
- ✧ permet de maintenir des tables de routage cohérente

# Routage dynamique IGP

---

✧ But : mettre à jour dynamiquement les tables de routage lorsque :

- La topologie interne de « l'AS » est modifiée (câble coupé, routeur ajouté, modification de l'adressage, ...)

✧ Deux classes d'algorithmes existent :

les algorithmes *Vector-Distance (à vecteurs de distance)*

- ◆ chaque routeur ne connaît que ses voisins
- ◆ échange des tables de routage en utilisant un vecteur
- ◆ Utilisation et sélection des routes ayant le coût minimal
- ◆ Exemple : RIP, IGRP, EIGRP

les algorithmes *Link-State (à état de liens)*

- ◆ Chaque routeur reconstruit une carte complète de l'AS
- ◆ Chaque routeur recherche la meilleure route vers les autres routeurs
- ◆ Exemple : OSPF, IS-IS



*Et pour  
les réseaux sans fil ?*

# Les réseaux mobiles Ad Hoc (1)

---

## ✦ Définition:

Un réseau mobile ad hoc, appelé généralement **MANET** (**Mobile Ad hoc NETWORK**), consiste en une grande population, relativement dense, d'unités mobiles qui se déplacent dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée.

Un réseau ad hoc peut être modéliser par un graphe  $G = (V, E)$   
où - V représente l'ensemble des nœuds et  
- E l'ensemble des connections qui existent entre ces nœuds.

**But : Faire communiquer deux nœuds distants en passant par un ou plusieurs intermédiaires**

**Application : aéroport, sécurité routière, randonnée ...**



# Les réseaux mobiles Ad Hoc (2)

---

## ✧ Caractéristiques :

### ✧ Topologie dynamique

- ✧ c'est le but de ces réseaux, pouvoir bouger, donc changement continu de la topologie.
- ✧ impossible de savoir si un ordinateur sera encore joignable dans les minutes suivantes.

### ✧ Bande passante limitée /débit limité

- ✧ du fait du partage de la bande passante

### ✧ Contraintes énergétiques

- ✧ mouvement ➔ utilisation batterie...

### ✧ Absence d'infrastructure

### ✧ Sécurisation des données

- ✧ problème au niveau physique (inondation, dispersion,...)



# Routage dans réseaux MANET

---

## ✧ Objectifs :

### ✧ Minimiser la charge réseau

- Eviter les boucles
- Eviter la concentration du trafic en un point
- Eviter de surcharger les entêtes des trames ou des paquets

### ✧ Effectuer des communications multi-points fiables

### ✧ Assurer un routage optimal

### ✧ Optimiser le temps de latence

- De nombreux protocoles utilisent un timer pour gérer leurs envois de données → problème si latence trop longue

# Réseaux MANET

---

## ✧ Classification

### ✧ Mode infrastructure

- ✧ Routage classique filaire ( protocole statique/ dynamique)

### ✧ Réseau "mesh" sans fil ( Réseau maillé sans fil)

- ✧ Routage **dynamique classique** (OSPF, RIP,...)
- ✧ Les nœuds servant de routeurs ne bougent pas

### ✧ Réseau Ad hoc

- ✧ Création de nouveaux protocoles de routages dynamiques
- ✧ Chaque station est un routeur potentiel
- ✧ **Protocole proactifs, réactifs, mixtes**

# Routage, classification

---

## ✧ Protocoles proactifs

- ✧ échange périodique de paquets de contrôle
- ✧ **toutes les routes sont disponibles à tout instant**  
➡ mise à jour périodique des tables de routage
- ✧ exemple : DSDV, OLSR, FSR, WRP,...

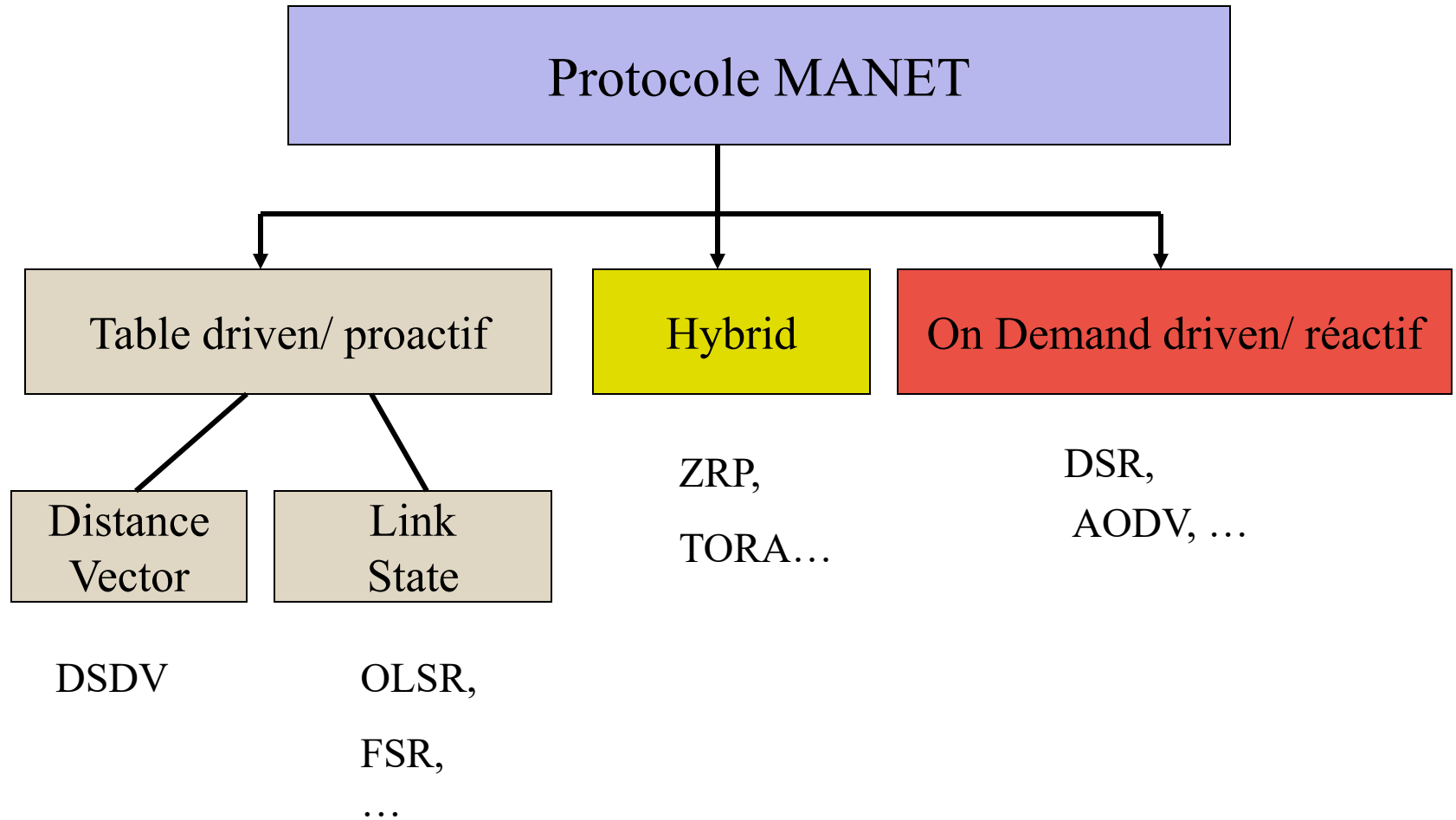
## ✧ Protocoles réactifs

- ✧ **les routes sont créées à la demande**
- ✧ minimisation du trafic, mais augmentation du temps de latence
- ✧ ouverture des routes par inondation
- ✧ exemple : DSR, AODV, ...

## ✧ Protocoles mixtes

- ✧ proactif local, réactif éloigné
- ✧ exemple : ZRP, TORA...

# Routage, classification (2)



# DSDV (1)

## ✧ Destination-Sequenced Distance Vector protocol (Vecteur de Distance à Destination Dynamique Séquencée)

- ✧ But :
- Garder la simplicité des algorithmes à vecteurs de distance
  - Eviter le problème de la formation des boucles
  - Autoriser les stations (nœuds) à ne pas bouger

✧ Implantation:

- ✧ Basé sur l'algorithme de Bellman-Ford  
    ➡ ( amélioration du RIP)
- ✧ Ajout d'un nouvel attribut dans la table de routage :  
    **le Numéro de Séquence** (SN : Sequence Number)
- ✧ Chaque nœud joue le rôle d'un routeur (si nécessaire)

# DSDV (2)

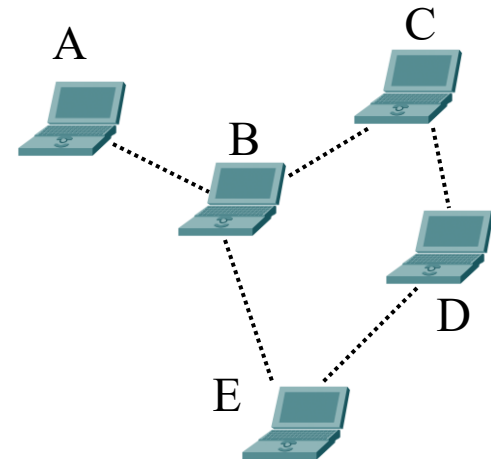
Chaque nœud a une table de routage qui contient :

- Pour toutes les destinations possibles

- Le nombre de sauts ( de nœuds) nécessaire pour atteindre la destination
- Le prochain saut pour chaque destination
- Le numéro de séquence qui correspond à un nœud destination

Exemple : table de routage de B

Dest	Saut	Metric	N° seq
A	A	1	A-212
B	/	0	B-24
C	C	1	C-146
D	C	2	D-378
E	E	1	E-156





# DSDV (3)

---

## ✧ Mise à jour

- Périodique ( comme dans RIP → 15s)
- Déclenchée  
    ➡ lorsqu'il y a de forts changements dans la topologie

## ✧ 2 façons de réaliser cette mise à jour

- ◆ soit mise à jour complète
  - on envoie la totalité de la table de routage
- ◆ soit mise à jour partielle ou incrémentale
  - on envoie que les informations ayant été modifiées depuis la dernière mise à jour complète

Si mise à jour incrémentale ne tient pas dans une NPDU, alors on fait une mise à jour **complète**.



# DSDV (4)

## ✧ Changement dans la table de routage

- ✧ Toute route avec un numéro de séquence supérieure à celui existant est mise à jour
- ✧ Si même numéro de séquence, alors on compare la métrique :
  - si métrique plus petite, mise à jour, sinon rien ( minimisation du chemin parcouru)
- ✧ Si route inexistante, alors rajout.

➡ Les changements sont diffusés aux autres nœuds.

# DSDV ( 4bis)

Dest	Saut	Metric	N° seq
A	A	0	A-130
B	B	1	B-56
C	E	2	C-212
D	E	3	D-156
E	E	1	E-40

paquet de maj venant de A

+

Dest	Saut	Metric	N° seq
A	A	1	A-126
B	/	0	B-56
C	A	4	C-210
D	E	2	D-156
E	E	1	E-40

Table de routage de B

=

Table de routage de B

Dest	Saut	Metric	N° seq
A	A	1	A-130
B	/	0	B-56
C	A	3	C-212
D	E	2	D-156
E	E	1	E-40

# DSDV (5)

## ✧ Perte d'un lien

- ✧ Incrémentation du numéro de séquence et diffusion
- ✧ Métrique mise à l'infini
- ✧ On évite la création de boucle grâce au numéro de séquence

## ✧ Nouveau lien

- ✧ apparition ou ré-apparition d'une station
- ✧ Incrémentation de 2 du numéro de séquence et diffusion

Si déplacement d'un nœud, on a perte d'un lien et ensuite création d'un nouveau lien, d'où numéro de séquence supérieur

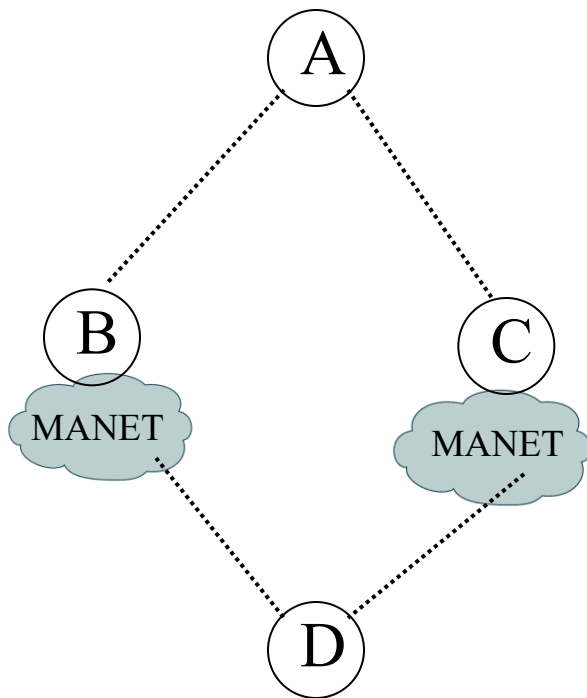
→ mise à jour correct de la table de routage

# DSDV (6)

## ✧ Quelques problèmes :

### ✧ **La fluctuation**

- ✧ route mise à jour : si  $n^{\circ} \text{ séquence reçue} > n^{\circ} \text{ séquence actuelle}$   
ou si même numéro de séquence, et métrique inférieure



On suppose :

- B a un chemin vers D en 10 sauts
- C a un chemin vers D en 8 sauts

Si D met un jour sa route vers A avec un nouveau numéro de séquence alors...

B envoie maj vers A → A change de route

→ C envoie maj vers A → A rechange de route

la route de A vers D va fluctuer entre B et C

# DSDV (7)

---

## ✦ Raisons

- ✦ irrégularité dans les mises à jour,
- ✦ trafic asynchrone,
- ✦ différentes vitesses de propagation,...

## ✦ Solutions

- ✦ **Maintenir deux tables**
  - une table contenant la table de routage
  - une table contenant le temps à attendre avant de diffuser un changement dans la table de routage

# DSDV (8)



## Conclusion

- ◆ DSDV est *assez lent à converger*  
(problème des algorithmes à vecteurs de distance)
- ◆ DSDV ne peut converger si les nœuds ne s'arrêtent pas de bouger pendant 300 secondes
- ◆ Le taux de transmission réussit est de 70% à 90% lors de mouvement non rapide
- ◆ DSDV reste assez instable lors de changement assez brusque de topologie
- ◆ DSDV nécessite de temps en temps des mises à jour complète pour un bon fonctionnement
- ◆ Utilisation de deux tables : une de routage, une pour les "updates"

# OLSR (1)

---

## ✦ Optimized Link State Routing

- ✦ Basé sur un algorithme d'état de lien "optimisé"
  - Gestion des relais multipoints ( MPR : MultiPoint Relay)
- ✦ Protocole ratifié par l'IETF  
RFC 3626, 7181
- ✦ Utilise plusieurs tables :
  - Table de voisinage
  - Table de topologie du réseau
  - Table de routage



# OLSR (2)

## ✧ Découverte du voisinage

- ✧ Envoie de trames HELLO périodique

- ✧ 4 sortes de lien

- lien symétrique
- lien asymétrique
- lien MPR
- lien perdu

—————> plusieurs trames "hello" sans réponse

➡ Un lien sera considéré comme fiable s'il est symétrique

4 échanges pour avoir lien symétrique :

- A envoie une trame Hello vide ( on découvre)
- B répond avec une trame Hello asymétrique ( B entend A)
- A répond avec une trame Hello symétrique
- B répond avec une trame Hello symétrique

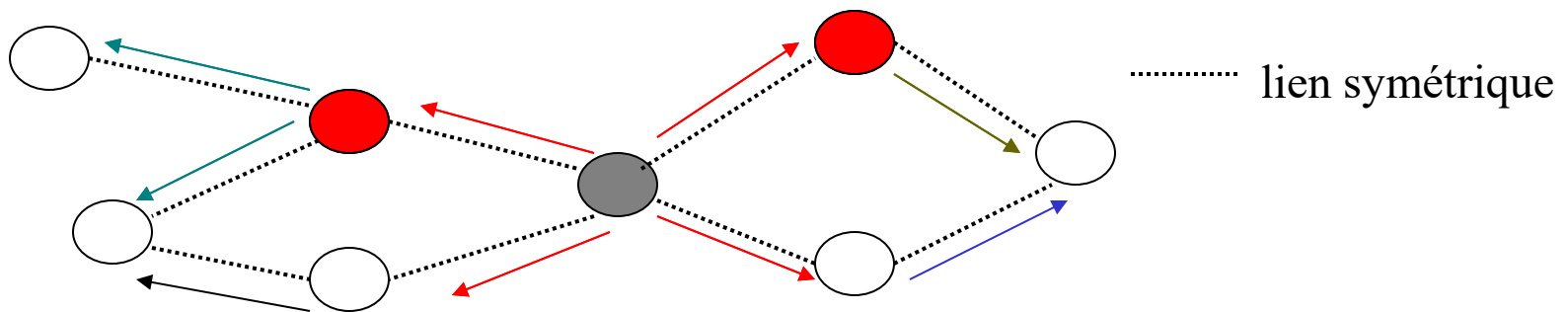
# OLSR (3)

## ✦ Les relais multipoints (MPR)

Chaque nœud envoie à tous ces voisins sa table de voisinage.

- Permet de remplir la table de topologie
- Permet à un nœud de connaître la liste des stations se trouvant à 2 sauts

- ✦ But des relais multipoints : minimiser le nombre de transmission pour que tous les voisins de second niveau reçoivent l'information.  
(une station peut-être visible par plusieurs nœuds)



SANS MPR → 5 retransmissions

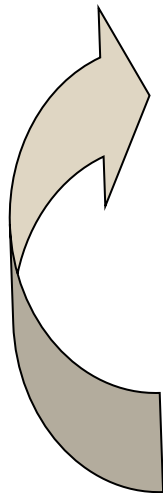
AVEC MPR → 3 retransmissions

# OLSR (4)

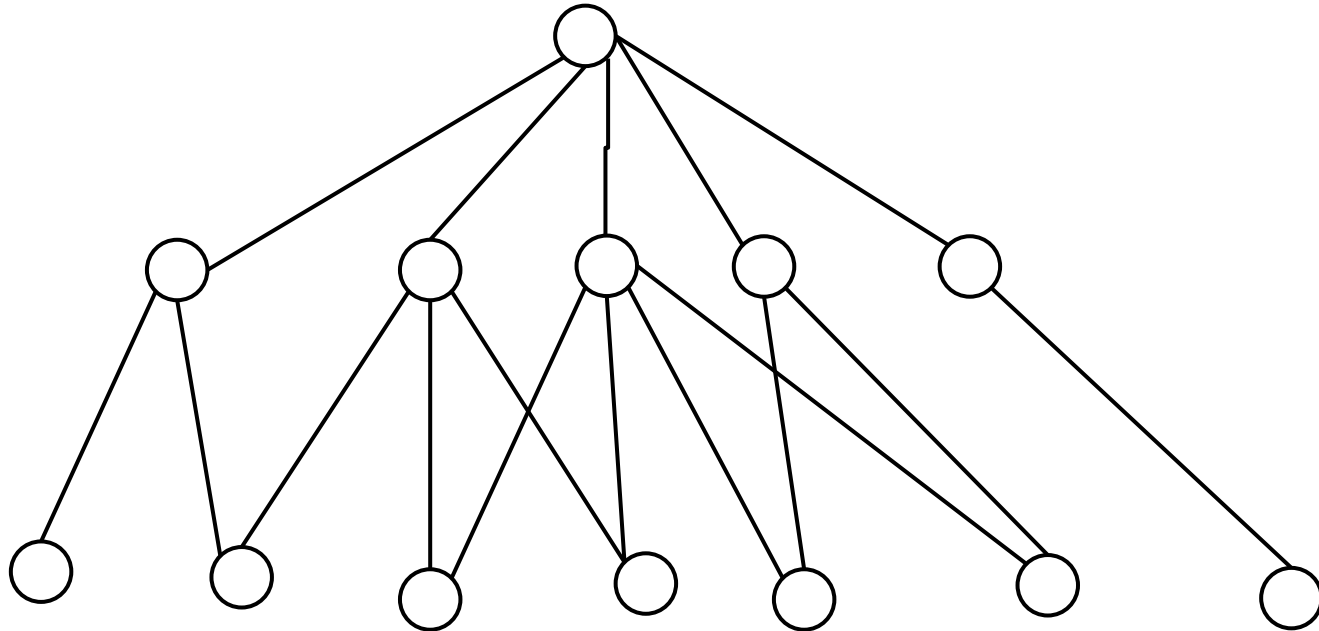
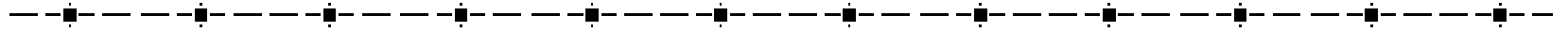


## Election des MPR

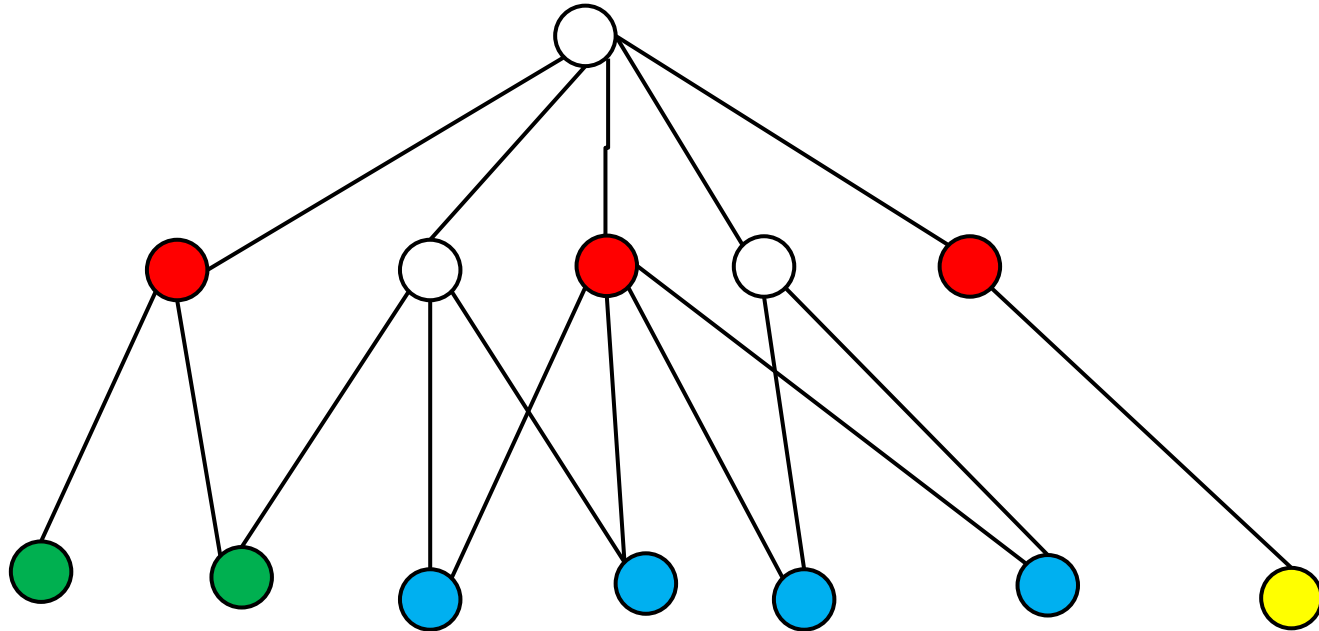
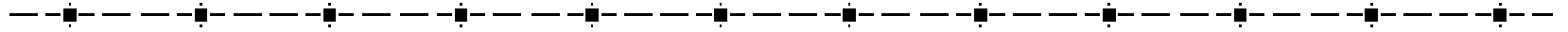
- Algorithme np-complets → impossible car trop lent
  - utilisation d'un heuristique
1. On recherche les nœuds du second degré isolé  
(lien unique avec un nœud du 1er degré)  
→ on étiquette alors les nœuds du 1er degré les desservant
  2. On élimine tous les nœuds accessibles par les MPRs trouvés
  3. On recherche le nœud du 1er degré permettant d'accéder aux plus grands nombres de nœuds du second degré
  4. On étiquette ce nœud alors comme MPR, et on revient à l'étape 2 tant qu'il reste des nœuds du second degré.



# OLSR (4 bis)



# OLSR (4 bis)



# OLSR (5)

## ✧ Base de données topologique

✧ Pour chaque destination, on garde :

- l'adresse de destination
- le dernier point relais (MPR) avant l'adresse de destination
- un numéro de séquence
- le temps avant destruction si aucune remise à jour

Pour construire cette base, les MPRs diffusent de **temps en temps** leur table de voisinage (Topology Control –TC-) qui sont relayés par les autres MPRs.

Si la trame est trop grande, plusieurs trames sont envoyées.

Chaque TC a un numéro de séquence (cf DSDV)

**Attention** : un nœud non MPR n'envoie que des trames "hello" à ces voisins et donc ne diffuse rien au reste du réseau



# OLSR (6)

---

## ✧ Table de routage

- ✧ création identique à un algorithme d'état de lien
  - utilisation table de voisinage, bd topologique et SPF.
  - utilisation des liens stables

## ✧ Autres

- ✧ 4 sortes de messages de contrôle / UDP port 698
  - trame Hello
  - trame TC
  - trame MID
    - message contenant les différentes interfaces d'un nœud
  - trame HNA
    - message permettant de déclarer des adresses réseaux en dehors du MANET ( utilisé par les passerelles)



# OLSR (7)

---

## ◆ Conservation de l'énergie

- Une station mobile peut se mettre en attente

Si nœud normal, aucun problème

Si MPR → problème la diffusion des informations

### Solution :

- le nœud ne répond plus aux messages "hello", mais continue de diffuser les messages TC
- le nœud va être considéré comme perdu, donc modification de la topologie
- recalcule pour un nœud lointain d'un nouveau MPR, le nœud redevient un nœud normal.

# OLSR (8)

---

## ✦ Conclusion

- ✦ Protocole avec la technique à état de liens
  - Pas de boucle
- ✦ Minimisation de la bande passante
  - en minimisant le nombre d'envoie
  - en utilisant la périodicité
- ✦ Beaucoup de travail sur ce protocole  
exemple : NOLSR, NOA-OLSR, EE-OLSR, BATMAN...

# Protocole réactif

---

## ✧ Protocole réactif

- ✧ apprentissage des routes à la demande
- ✧ permet d'éviter le stockage de routes inutiles
- ✧ diminution des messages de contrôle si la topologie ne bouge pas
- ✧ ouverture des routes par inondation
  - Sélection de la route la plus courte renvoyée
  - en cas de rupture, récupération de la route par inondation
- ✧ autorise
  - lien symétrique
  - lien asymétrique

# AODV (1)

---

## ✧ Ad Hoc On –Demand Distance Vector

- ✧ Basé sur l'algorithme de Bellman-Ford

- vecteur de distance, mais en évitant les boucles
- assez proche de DSDV.

- ✧ Protocole normalisée par l'IETF  
RFC 3561

- ✧ Protocole réactif

- supporte l'unicast et le multicast
- Plutôt développé pour les liens symétriques
- 2 fonctions : la découverte des routes  
la maintenance des routes

# AODV (2)

---

## ✧ Table de routage

✧ Elle contient :

- l'adresse de destination
- Le nombre de sauts ( de nœuds) nécessaire pour atteindre la destination
- Le prochain saut
- Le numéro de séquence qui correspond à cette destination
- Le temps d'expiration de l'entrée de la table
- La liste des précurseurs

Une destination D est rajoutée dans la table de routage seulement lorsque la station veut envoyer un message à D

➡ temps d'expiration arrive à 0 → élimination de la destination

# AODV (3)

---

## ✧ Découverte d'une route

### ✧ Broadcast d'un paquet **ROUTE REQUEST** (*requête de route*) avec la destination D

- Ce paquet contient :
  - un numéro de broadcast (RREQ ID)
  - l'adresse de la source (initiateur de la demande)
  - l'adresse destination
  - le n° séquence correspondant à la dernière fois que cette destination était dans la table de routage  
( si jamais, n° séquence = 0 et flag = U)
  - le n° séquence à utiliser actuellement pour le retour

Si pas de retour avant NET\_TRAVERSAL\_TIME, n° de broadcast est incrémenté de 1, et une RREQ est relancée

un seul n°broadcast par station ( pour toutes les destinations)



# AODV (4)

## ✧ Réception d'une RREQ

### ✧ Si destination,

- Vérification si (n° RREQ, id\_source) n'a pas été déjà reçu
- mise à jour de sa table de routage
  - insertion du chemin vers le destinataire
  - mise à jour du n° séquence,
  - mise à jour du temps d'expiration, ...
- Renvoie d'une Route Reply (RREP) en unicast

### ✧ Si nœud intermédiaire,

- vérification si (n° RREQ, id\_source) n'a pas été déjà reçu
- cas 1 : Pas de route disponible vers destination D
  - ✧ Mise à jour de la table de routage pour le retour  
(on suppose les liens symétriques)
  - ✧ Diffusion de la demande en ajoutant 1 au nombre de saut



# AODV (5)

- cas 2: route disponible vers la destination D
  - ♦ Vérification du n°séquence de l'ancienne route (stocké dans le paquet RREQ) < n° séquence de la route trouvée dans la table de routage
  - ♦ Renvoi d'une Route Reply en unicast avec le nombre de saut correspondant à la distance jusqu'à destination
  - ♦ Mise à jour de la liste des prédécesseurs pour cette destination

**➡ Mais, il faut avertir la destination qu'un nœud cherche à le joindre ( car autrement table de routage incomplète)**

Emission d'une RREQ "gratuite"  
vers la destination (flag G)

(champ de la trame RREQ)

# AODV (6)

## ✧ Mise à jour table de routage

- ◆ Lors de la réception d'une RREQ
- ◆ Lors de la réception d'une RREP
  - lorsqu'un n° séquence est plus grand que celui de la table
  - Même numéro de séquence, mais nombre de saut plus petit
- ◆ A chaque réception d'un paquet, remise à jour du délai d'expiration de la route (**temps restant + ACTIVE\_ROUTE\_TIMEOUT**)
- ◆ Lors de la réception d'une RERR
- ◆ Lorsqu'un timeout de validité de la route expire
  - La route est éliminée ou marquée inutilisable

Un chemin figurant dans la table de routage est considéré comme :  
*un chemin actif*

# AODV (7)

## ✦ Route reply

- ✦ Lorsqu'une station Z reçoit une route reply, on renvoie l'information à un nœud stocké dans la table de routage  
( du fait du passage précédent de RREQ)
  - Mais si lien unidirectionnel ➡ ECHEC

### Problème :

- Si le nœud source renvoie une RREQ, forte probabilité de se retrouver dans le même cas
- Si le nœud Z reçoit après le lien unidirectionnel une autre route valide ➔ information peut être effacée si chemin plus long

**Solution :** Si un nœud s'aperçoit d'un lien unidirectionnel, ce lien est mis en **blacklist**, c'est à dire que l'on ignorera par la suite toutes les RREQ qui en proviennent.

# AODV (8)

## ✧ Limitation :

### ✧ Chercher une route prend du temps

- maximum RREQ\_RATELIMIT début de connexion par secondes
- Tous les paquets sont bufferisés
- Lorsqu'une route est trouvée, tous les paquets sont envoyés  
sinon, tous les paquets sont effacés et la couche transport est prévenu
- Par défaut, RREQ\_RATELIMIT = 10

# AODV (9)

---

## ✧ Optimisation

- ✧ Minimisation du chemin parcouru par les paquets RREQ
  - le TTL au niveau IP de la 1er RREQ est mis à TTL\_START  
( généralement TTL\_START = 1)
  - Si pas de réponse RREP, alors  $TTL = TTL + TTL\_INCREMENT$   
( généralement TTL\_INCREMENT = 2)
  - Le TTL augmente jusqu'à TTL\_THRESHOLD (7)
  - 2 essais maximum avec le TTL max avant de déclarer la route inaccessible
- Si perte d'un chemin dans la table de routage, alors  
 $TTL = \text{distance connue dans la table} + TTL\_INCREMENT$   
(on autorise une recherche de chemin plus long)

# AODV (10)

---

## ✧ La maintenance d'une route

### ◆ Emission

- Pas de communication réelle : envoie de paquets "hello" vers les nœuds appartenant à des chemins actifs  
( TTL =1 pour ne pas aller loin)

### ◆ Réception

- Remise à jour du timer d'expiration quand :
  - ◆ réception d'un paquet hello
  - ◆ réception d'un paquet de données venant d'un nœud en direction d'un autre(temps restant + ACTIVE\_ROUTE\_TIMEOUT)



# AODV (11)

## ✧ Route Error

- ✧ En cas de disparition d'un nœud,
  - on recherche l'ensemble des destinations qui étaient accessibles par ce nœud
  - on incrémente le n° séquence de chaque destination devenue invalide
  - on envoie une RERR à tous les prédécesseurs de ces destinations
  - Mise à jour de la table de routage
- ✧ Réception d'une RERR
  - on met à jour la table de routage
  - on envoie l'information aux prédécesseurs

On met en place un timer  
DELETE\_PERIOD. Si  
RAS, alors effacement

# AODV (12)

---

## ✦ Sauvetage !!

- ✦ Le nœud le plus proche du nœud qui disparaît peut essayer de trouver une nouvelle route

➡ émission d'une RREQ

- Mise en attente des paquets dans un buffer

Si une réponse valide, envoie d'une RERR avec le bit N pour indiquer un changement dans la route, mais que celle-ci reste valide

➡ remise à jour des routes et du nombre de saut pour toutes les stations

# AODV (13)

---

## ✦ Conclusion

- ✦ protocole réactif utilisant les vecteurs de distance
  - Convergence assez rapide
- ✦ n° séquence permet d'éviter les boucles
- ✦ un mécanisme existe pour traiter le multicast
- ✦ Permet de gérer l'IPv6 (AODV6)
  
- ✦ Routage utilise trames UDP sur le port 654

# Protocole hybride

---

## ✧ 2 façons de gérer dynamiquement les routes

- soit d'une manière proactive
  - ✧ routes sont disponibles immédiatement
  - ✧ mais beaucoup de trafics de mises à jour
- soit d'une manière réactive
  - ✧ routes sont longues à être trouvées
  - ✧ beaucoup de trafic pour trouver la route (inondation)
  - ✧ autrement, peu de trafic

## ✧ Hybride

- ✧ mélange de réactif et proactif
- ✧ Pour les nœuds proches, connaissance des routes immédiates  
→ proactif
- ✧ Pour des nœuds lointains, utilisation d'un protocole réactif

# ZRP (1)

## ✧ Zone Routing Protocol

✧ Comme son nom l'indique, basé sur des zones

- Une zone contient l'ensemble des stations autour d'un nœud de référence dont la distance (nombre de sauts) est inférieure ou égale à x.
- La distance x est appelé "zone radius"
- Chaque nœud a sa zone individuelle. Les stations situées par le chemin le plus court à la distance x du nœud de référence sont appelées les nœuds frontaliers (peripheral ou bordercast node)
- A l'intérieur de la zone, utilisation d'un protocole proactif, à l'extérieur utilisation d'un protocole réactif

# ZRP (2)

---

## ✧ Le protocole IARP (Intrazone Routing Protocol)

- ✧ Routage à l'intérieure de la Zone
- ✧ Utilisation d'un protocole proactif (choix libre ex :OLSR, DSDV)
- ✧ Quelques spécificités :
  - Chaque nœud envoie sa table de voisinage à tous ces voisins  
(utilisation de trame hello)
  - Détection de la perte de voisinage
  - Une table de routage semi-complète...  
(seule les stations appartenant à la zone sont répertoriées)



# ZRP (3)

---

## ✧ Le protocole IERP ( Interzone Routing Protocol)

- ✧ Routage à l'extérieure de la Zone
- ✧ Utilisation d'un protocole réactif (choix libre ex :AODV)
- ✧ Quelques spécificités :
  - Désactivation de la table de voisinage
  - Utilisation d'un numéro de séquence pour éviter les boucles
  - Utilisation du protocole BRP

BRP : Border Resolution Protocol

# ZRP (4)

## ✦ BRP ( à l'intérieur de IERP)

- ✦ Lorsqu'un nœud frontalier reçoit une demande pour une destination, *d'après les protocoles réactifs* → *diffusion*
- ✦ Mais, il a la connaissance de toutes les destinations dans sa zone,
  - si destination existe, retour de l'information
  - **sinon diffusion que vers ses nœuds frontaliers.**
- Minimisation du nombre de requêtes envoyées.

### Conclusion :

- Protocole essayant de prendre le meilleur des 2 grandes catégories
- Chemin proche dans la table de routage
- minimisation de l'inondation