

Les réseaux ad hoc mobiles





Sommaire

1. Introduction
2. Routage dans le réseau **Ethernet**
3. Routage dans les **MANETs**
 - Protocole proactif **OSLR**
 - Protocole réactif **AODV**

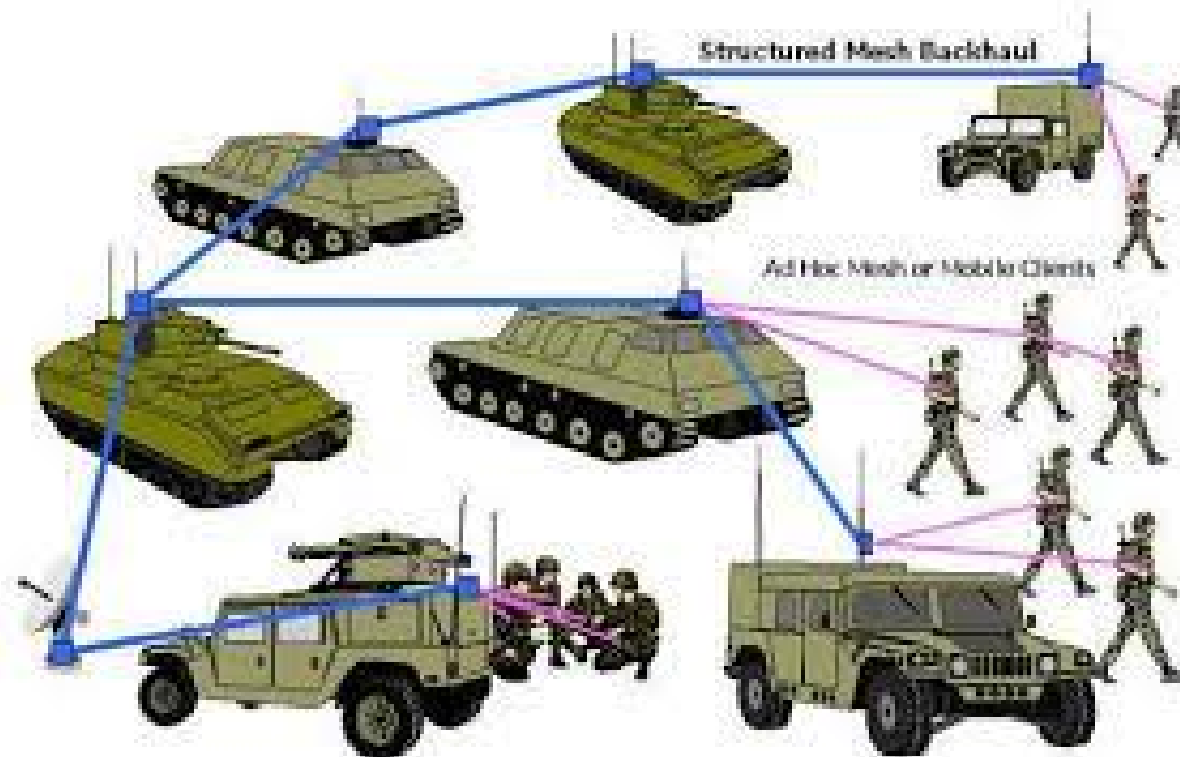
Introduction – Qu'est-ce que les réseaux ad hoc

❑ **MANET == Mobile Ad hoc Network.**

• «*ad hoc*» = «pour cela» = compétent, parfaitement qualifiée pour réaliser une tâche

❑ **Origine : application militaire**

• «Réseaux de combats»





Introduction – Qu'est-ce que les réseaux ad hoc

□ **Caractéristiques des MANETs:**

- ◆ Autonomes (**pas** de **routeurs** dédiés),
- ◆ Liaisons radios (bande passante **limitée**, taux d'**erreurs** élevé),
- ◆ Topologie **dynamique** (mouvement des nœud, entrée en sommeil),
- ◆ Utilisation de batteries pour l'énergie (Problème d'énergie),
- ◆ Sécurité physique limitée (vol),



Introduction – *Qu'est-ce que les réseaux ad hoc*

- Applications de collaborations
 - Meetings et échanger rapide des informations (*Bluetooth*)
- Acquisition de données dans des terrains inhospitaliers
 - Etude des espèces, météorologie, etc.
- Urgences et opération de secours
 - Catastrophes naturelles (infrastructures préexistantes sont non opérationnelles).
- Militaires
 - Interventions en milieu hostile (difficile d'utiliser un réseau à infrastructure).
- Etendre les réseaux
 - Assurer une couverture (zone où **il n'y a pas infrastructure**)

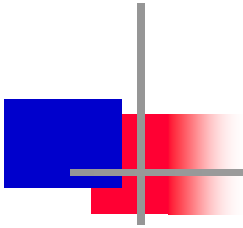
♦ **Avantages des réseaux ad hoc**

- Pas de câblage.
- Déploiement facile.
- Consommation énergétique faible.
- Mobilité et déplacement libre.
- Coût d'installation réduit.



Introduction – Qu'est-ce que les réseaux ad hoc

- ◆ Limitations typiques des réseaux ad hoc
 - Consommation d'énergie élevée
 - Bande passante faible
 - Taux d'erreur élevé
 - Mobilité élevée pour chaque nœud du réseau



2. Routage dans le réseau Ethernet filaire: *Rappelle*

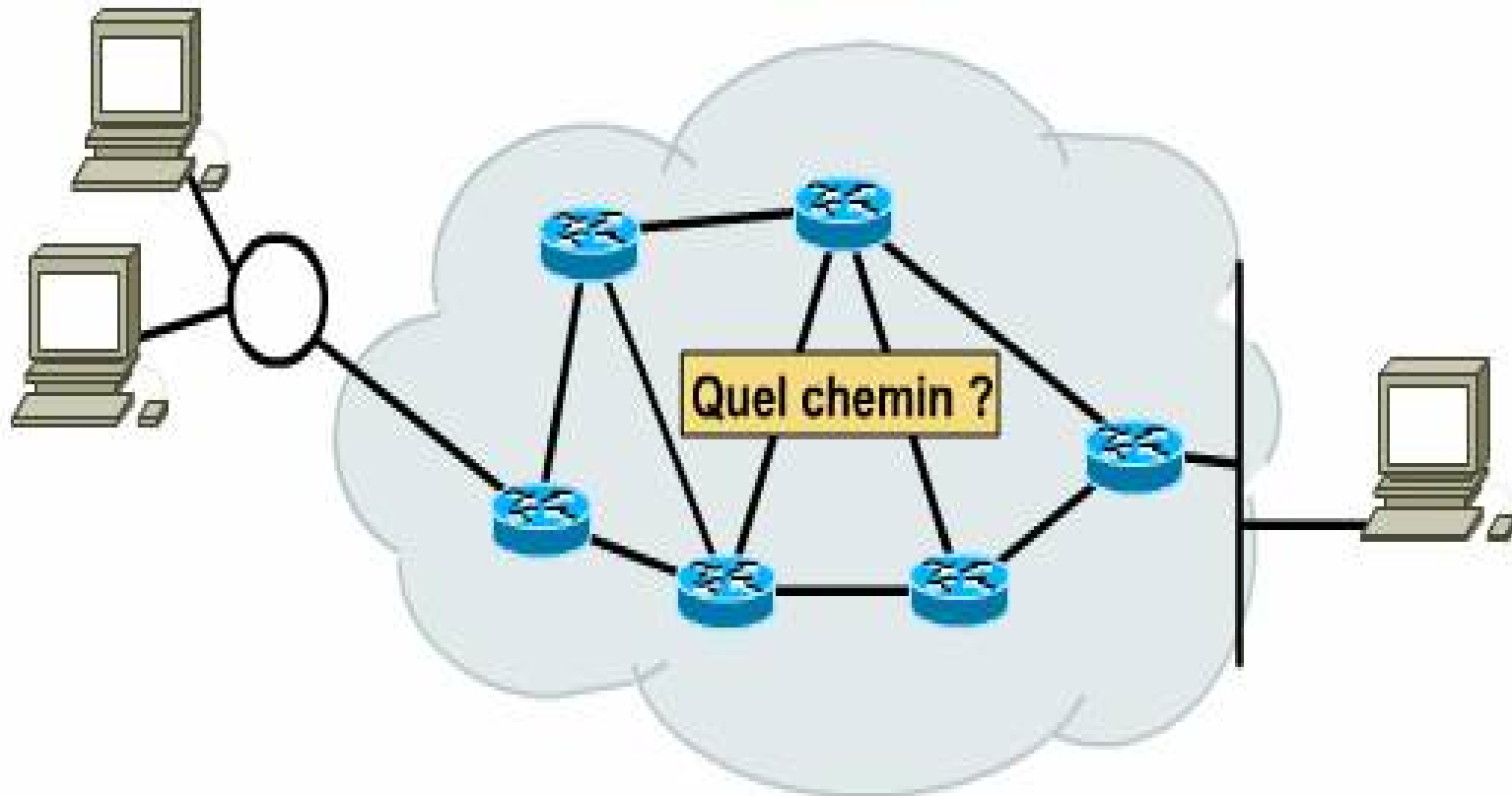


Routage dans le réseau Ethernet

Un routeur :

- Unité d'**interconnexion** de réseaux.
- **Achemine** les paquets de données entre les réseaux, en fonction des informations de la **couche 3**.
- Prend des décisions logiques d'optimisation pour **choisir** la meilleure voie d'**acheminement** des données.
- Les services de routage utilisent les informations de **topologie du réseau** dans l'évaluation des chemins.

Routage dans le réseau Ethernet



La couche 3 recherche le meilleur chemin dans l'interréseau.



Routage dans le réseau Ethernet

Protocole de routage:

- Permet aux *routeurs* de *communiquer* avec d'autres *routeurs* pour tenir à jour les *tables de routage*.
- Exemples de protocoles de routage *TCP/IP* :
 - ✓ Le protocole **RIP** (Routing Information Protocol)
 - ✓ Le protocole **IGRP** (Interior Gateway Routing Protocol)
 - ✓ Le protocole **EIGRP** (Enhanced Interior Gateway Routing Protocol)
 - ✓ Le protocole **OSPF**



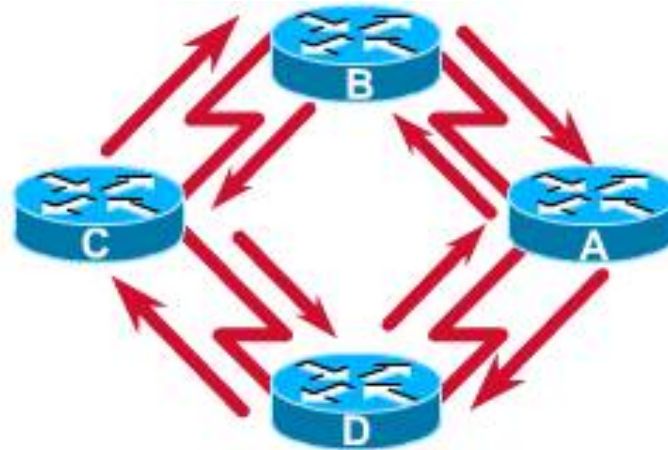
Types de routage dynamique

- ◆ Vecteur de distance « Distance Vector » (RIP)
 - ✓ Transmette des copies **périodiques** d'une table de routage d'un routeur à un autre.
 - ✓ Chaque routeur reçoit une table de routage des routeurs voisins auxquels il est directement connecté.
- ◆ État de lien « Link State » (OSPF)
 - ✓ Les algorithmes de routage à état de liens, également appelés algorithmes du *plus court chemin* d'abord, gèrent une base de données **topologiques** complexe.
 - ✓ Cette base de données topologique offre au routeur une **connaissance complète** sur les routeurs distants et leurs interconnexions.

Types de routage dynamique:

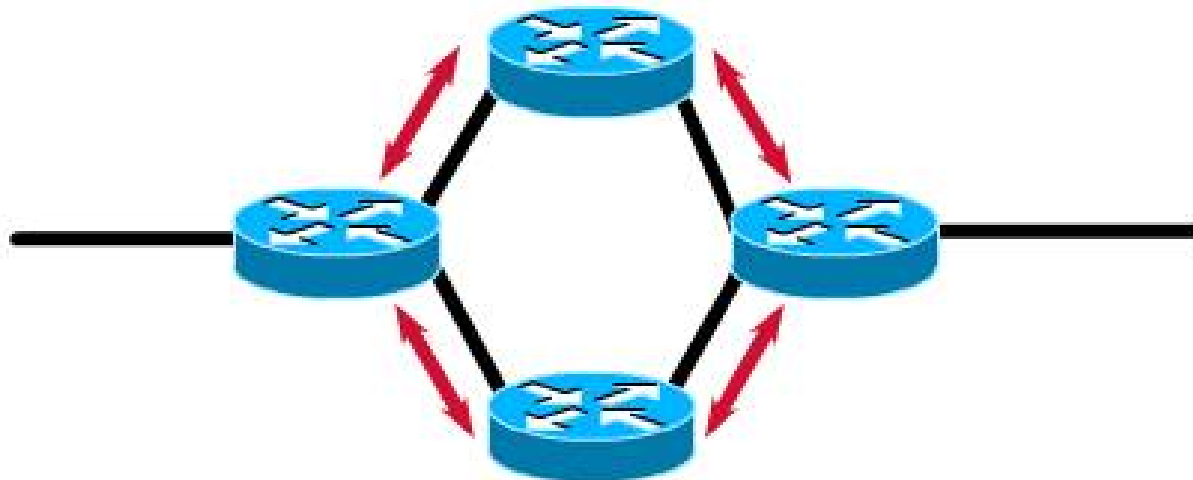
Vecteur de distance « Distance Vector » (*RIP*)

- Chaque routeur reçoit une table de routage des routeurs **voisins** auxquels il est directement **connecté**

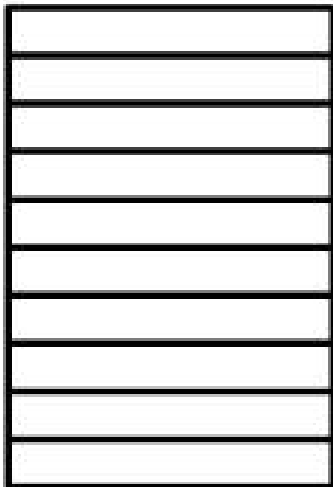


- ◆ Un routeur transmet périodiquement des copies de sa table de routage aux routeurs voisins et cumule les vecteurs de distance

Types de routage dynamique: État de lien « Link State » (OSPF)



Base de données
topologiques



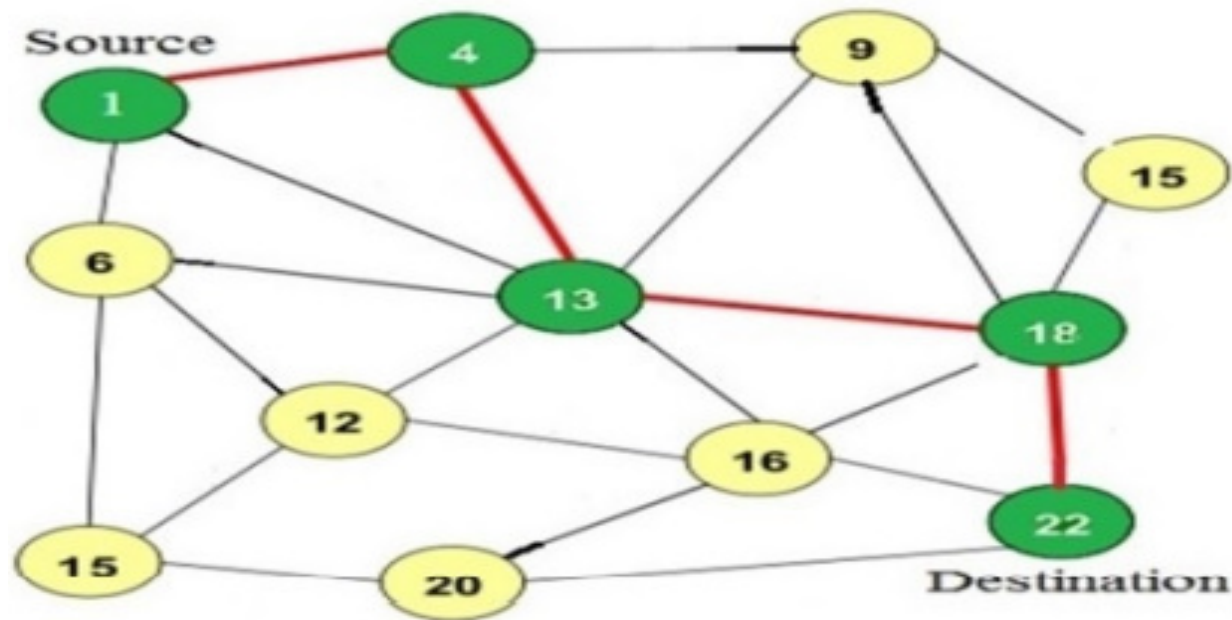
Paquets de mise à jour de
routage à état de liens

Algorithme
du plus court
chemin d'abord

Table de
routage

Arbre du plus court
chemin d'abord

3. Routage dans les MANETs



Exemple de Route ad hoc



Routage dans les MANETs

- ♦ Topologie Dynamique
 - Changements **fréquents** de liens, **qualité** des liens,
- ♦ Performances limitées des systèmes mobiles
 - Les mises à jour périodiques des tables de routage ont besoin de **l'énergie** et parfois **inutiles**.
 - Les échanges des informations de routage réduisent de plus en plus la **bande passante**.
- ♦ **Problème !**: Les protocoles conçus pour les **réseaux fixes** ne conviennent pas:
 - Changements peu fréquents
 - Les liens **symétriques**



Routage dans les MANETs

♦ Solution

- Concevoir des algorithmes **spécialement** pour les réseaux ad hoc

→ Léger (Énergie)

→ Optimal (précision des routes)

→ Scalable (mobilité / nœuds)

→ Sécurisé, ...

Exemple : Table de Routage d'un réseaux ad hoc

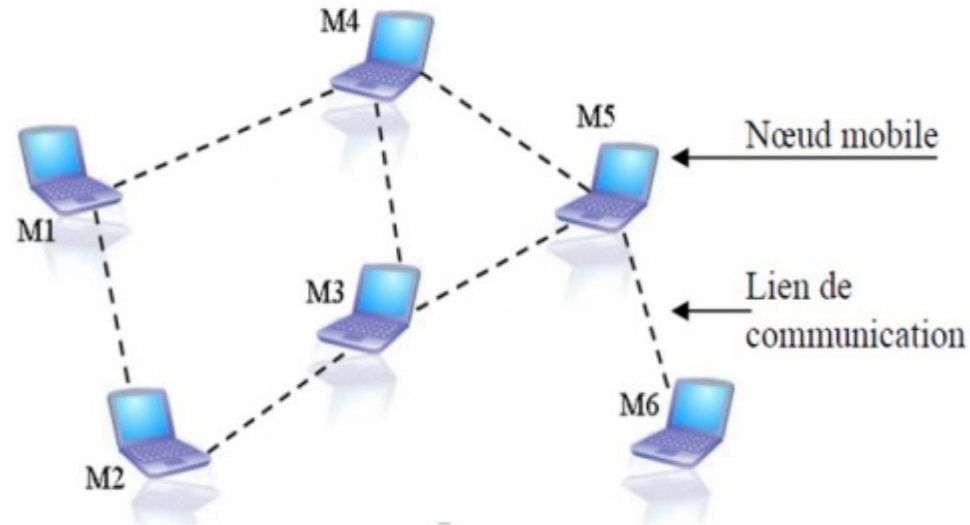
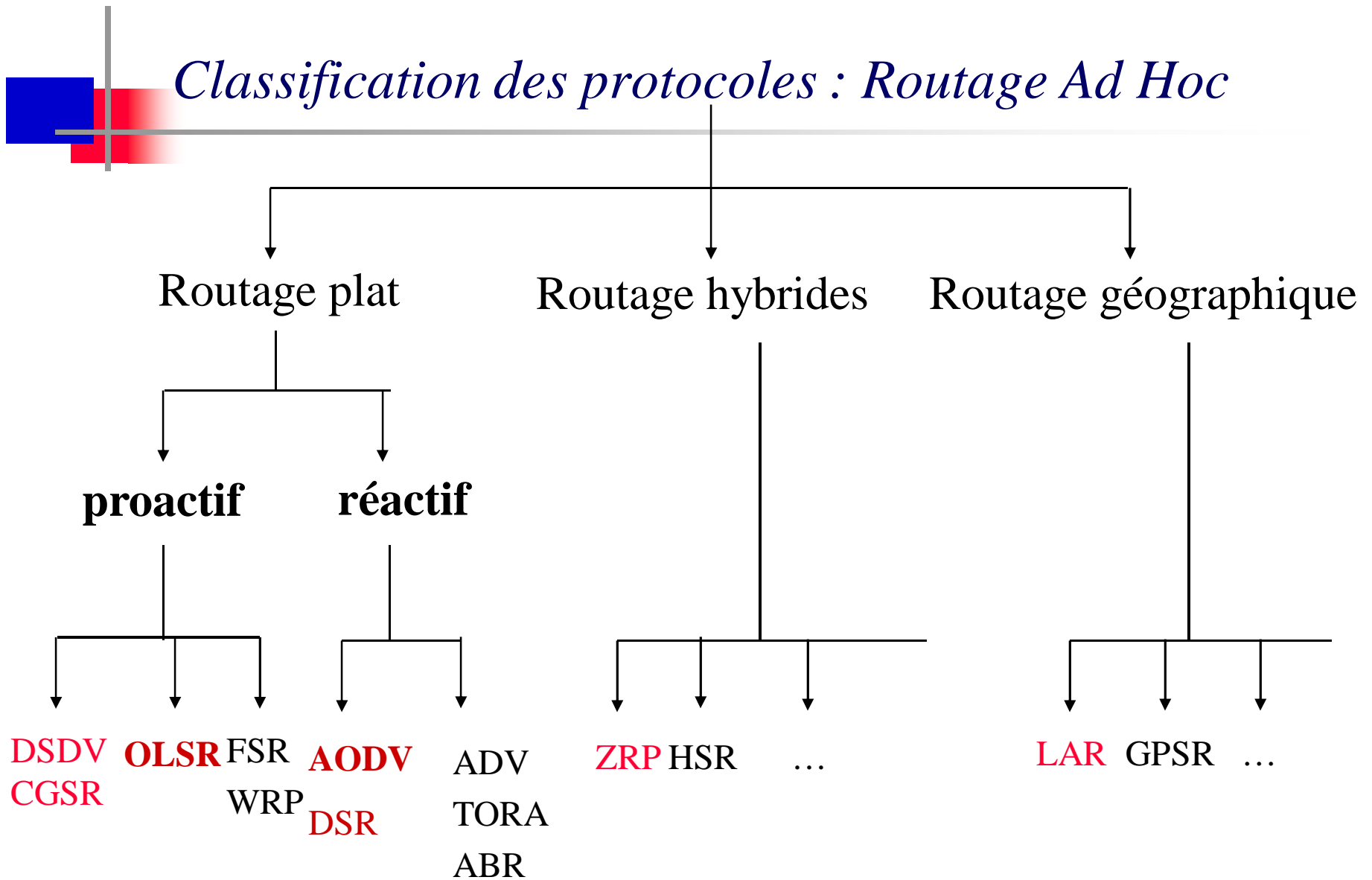


FIGURE 1.18 – Exemple d'un réseau Ad Hoc

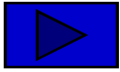
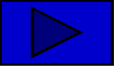
TABLE 1.4 – Table de routage du noeud M1 de la figure 1.18

Destination	Nombre de sauts	Prochain noeud	Numéro de séquence
M1	0	M1	NS1
M2	1	M2	NS2
M3	2	M2	NS3
M4	1	M4	NS4
M5	2	M4	NS5
M6	3	M4	NS6

Classification des protocoles : Routage Ad Hoc



Routage dans les réseaux ad hoc

- ◆ Deux types de routage :
 - Basé-table  = Table-driven / **proactif**
 - Basé-source initialisation (à la demande) 
= on-demand / **reactif**



Routage dans les réseaux ad hoc

Proactif vs Réactif

- Proactif: Basé-table
 - Chaque nœud **maintient** une ou plusieurs tables pour **stocker** des information de routage.
 - Chaque nœud **réagit** aux changements de la **topologie** du réseau en propageant des **misés à jour**.
- Réactif: Basé-source initialisation (à la demande)
 - Crée les routes à la **demande** du nœud source.
 - Une route est **maintenue** jusqu'à ce que la destination devient **inaccessible** depuis la source ou lorsque la route n'est plus **désirée**.



Protocoles de routage basé-table

Proactifs

- ◆ **OLSR**
 - **(Optimized Link State Routing)**



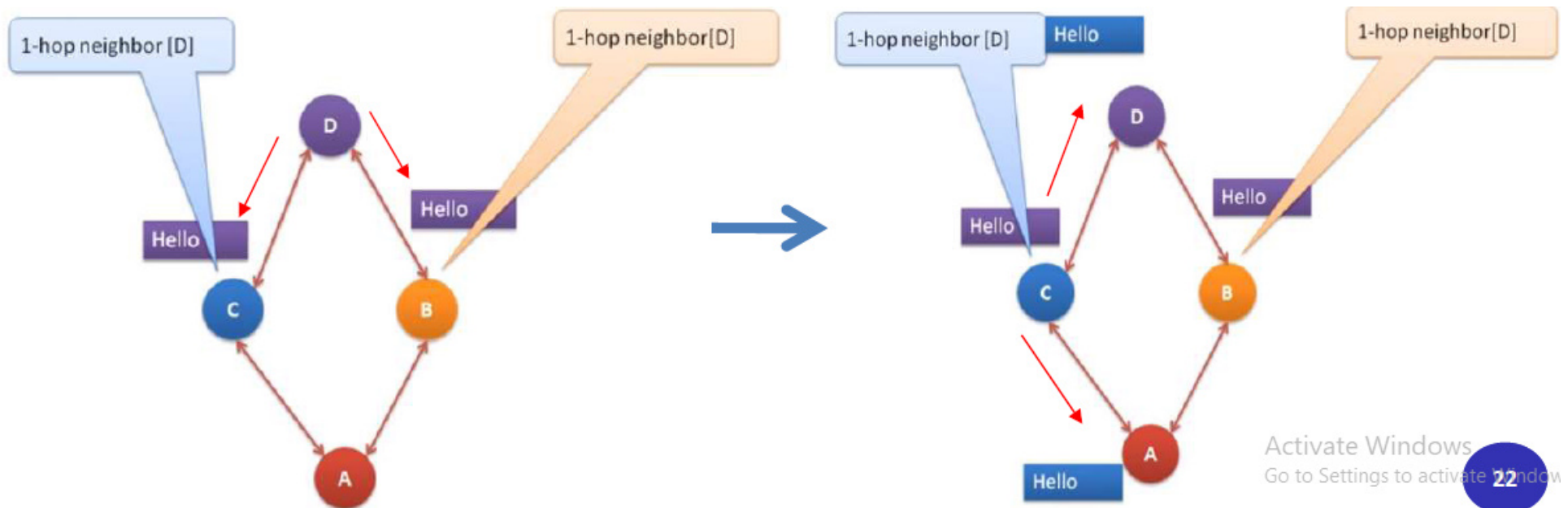
OLSR : idée de base

- ◆ Le concept principal est de sélectionner un sous-ensemble de nœuds appelés **relais multi-points** (**MPRs** : Multi-Point Relays)
- ◆ Les **MPRs** sont des nœuds choisis qui **expédient** des messages de diffusion pendant le processus d'*inondation*.
- ◆ **But : Réduire la surcharge.**
 - L'information d'état de lien est **produite** seulement par des nœuds élus comme **MPRs**.
 - Un nœud **MPR** doit rapporter seulement des liens entre lui-même et ses *sélecteurs*.
- ◆ fonctionnalités principales
 - **Détection de voisinage.**
 - **Gestion de la topologie.**
 - **Calcul des routes**

OLSR : Principe de fonctionnement

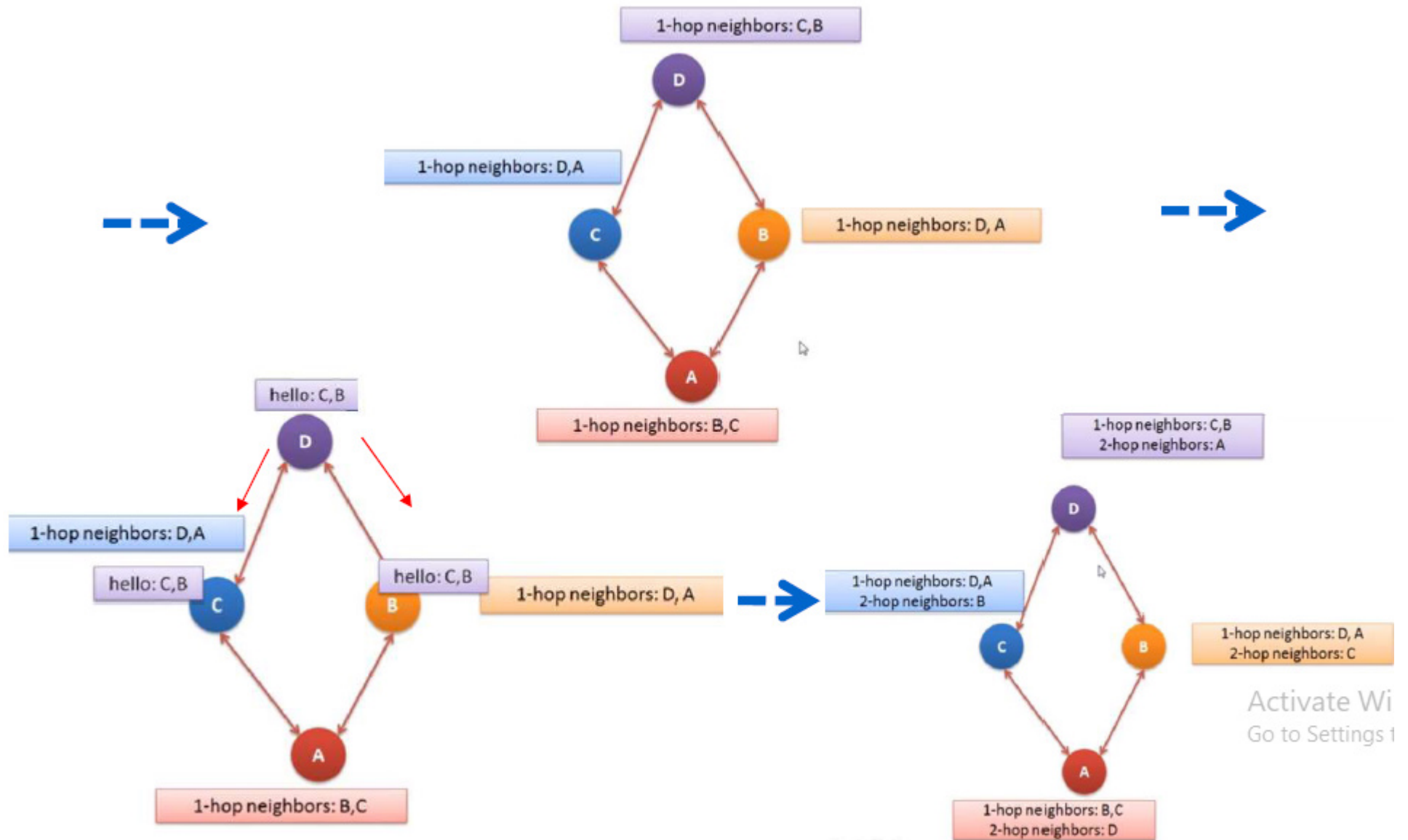
Détection de voisinage

- Chaque noeud diffuse **périodiquement** un message « **Hello** » dans son voisinage **1-hop** et **2-hops**.
- Le message contient des informations sur les interfaces voisines: adresse, **état de lien**.
- Les voisins traitent le message (ne le relaient pas)



OLSR : Principe de fonctionnement

Détection de voisinage

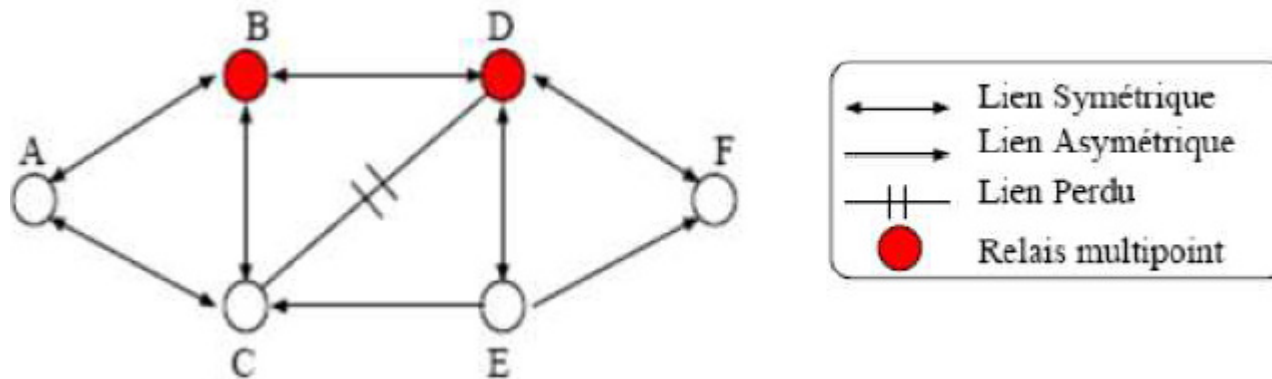


Activate Wi
Go to Settings 1

OLSR : Principe de fonctionnement

Détection de voisinage

- Les informations de voisinage sont *maintenues* dans une **table de voisinage** contenant: les voisins **directs**, les voisins à **deux sauts** les **MPRs** les *sélecteurs* de MPR



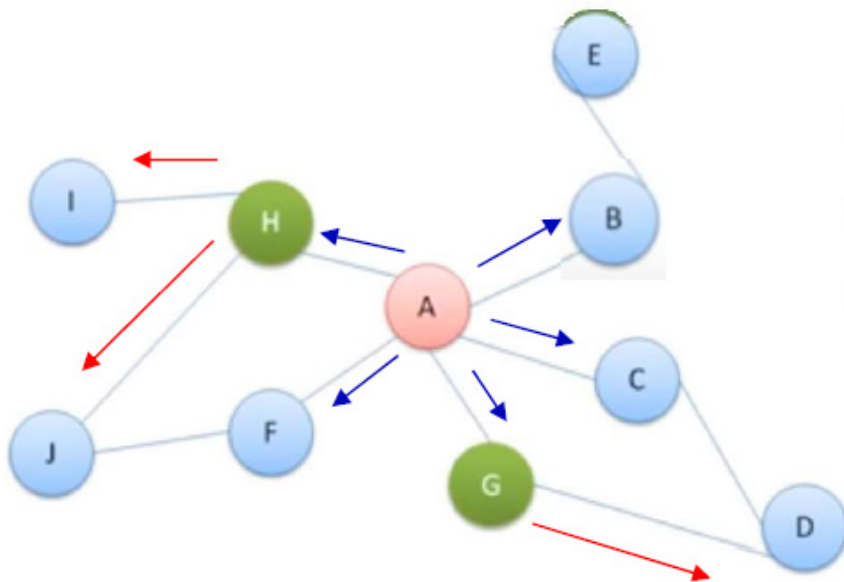
Base de A	Base de B	Base de C	Base de D	Base de E	Base de F
Voisins à 1 saut SYM: B, C ASY: — MPR: B Perdu: — Voisins à 2 sauts D via B Selecteurs de MPR —	Voisins à 1 saut SYM: A, C, D ASY: — MPR: D Perdu: — Voisins à 2 sauts E, F via D Selecteurs de MPR A, C, D	Voisins à 1 saut SYM: A, B ASY: C MPR: B Perdu: D Voisins à 2 sauts D via B Selecteurs de MPR —	Voisins à 1 saut SYM: B, E, F ASY: — MPR: B Perdu: C Voisins à 2 sauts A, C via B Selecteurs de MPR B, E, F	Voisins à 1 saut SYM: D ASY: — MPR: D Perdu: — Voisins à 2 sauts B, F via D Selecteurs de MPR —	Voisins à 1 saut SYM: D ASY: E MPR: D Perdu: — Voisins à 2 sauts B, E via D Selecteurs de MPR —

OLSR : Principe de fonctionnement

Gestion de la topologie

Principe (OLSR) *Eviter les transmissions non nécessaires, comment!:*

- Chaque noeud sélectionne, parmi ses **voisins**, ceux qui *diffusent* ses messages d'état: appelés **MPR** – *Multi Point Relay*
- Seulement les **MPR** peuvent transmettre des messages
 - Les voisins non MPR *traitent* les messages reçus mais ne les retransmettent pas
- Le noeud **MPR** ne retransmet que les paquets reçu de ses *sélecteurs*

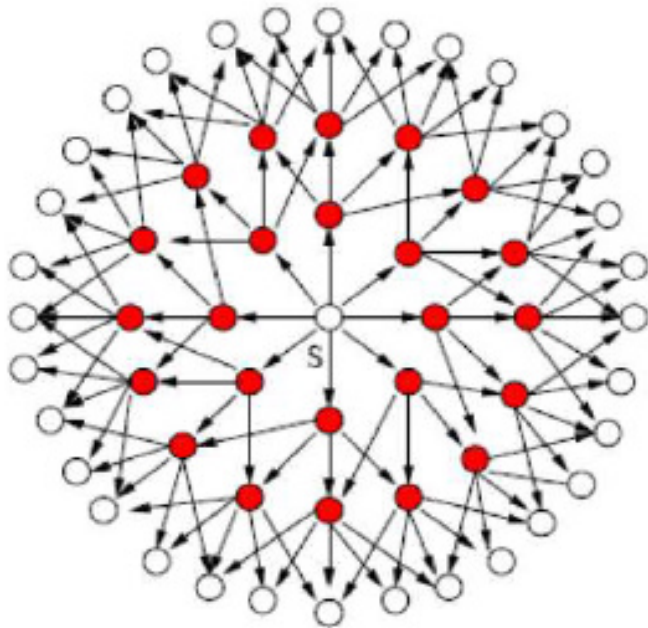


- A selecte **H** et **G** ses relais multipoints
- A diffuse un message à **tous ses voisins** (1-saut)
- Seuls H et G vont **diffuser** le message à leurs voisins

OLSR : Principe de fonctionnement

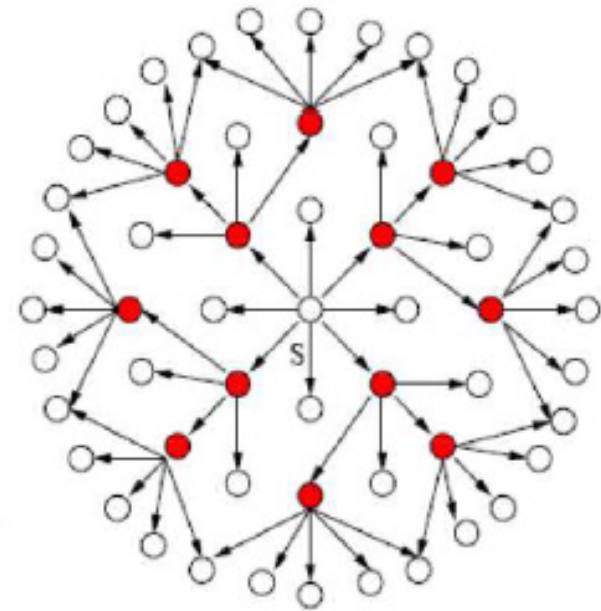
2- Gestion de la topologie

Inondation pure vs Inondation avec relais multipoints



Inondation pure

Diffusé au voisinage à trois sauts
par 24 retransmissions.



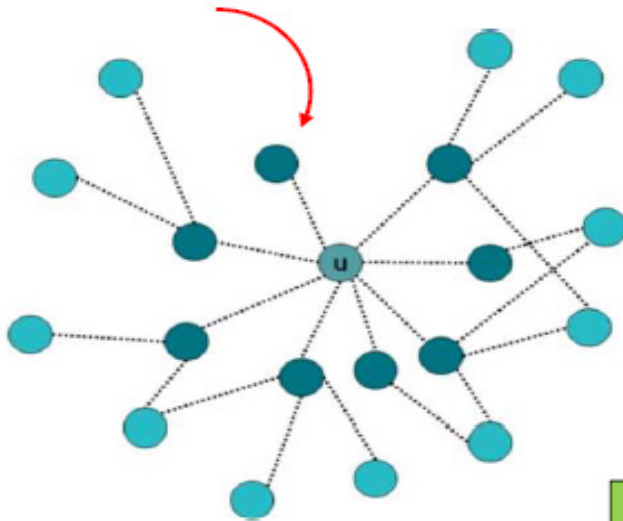
Inondation avec des relais multipoints

Diffusion au voisinage à trois sauts
par 12 retransmissions.

OLSR : Principe de fonctionnement

Gestion de la topologie : Sélection des Relais multipoint

Chaque nœud choisit ses MPR parmi ses voisins 1-hope

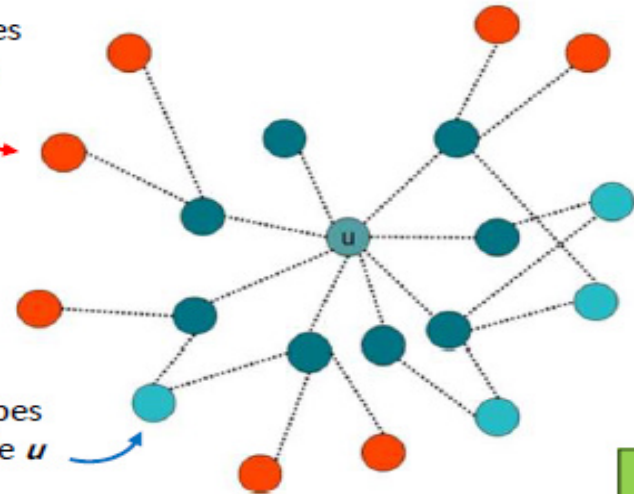


1

Etp1. Choisir les voisins 1-hope qui couvrent les voisins 2-hopes isolés

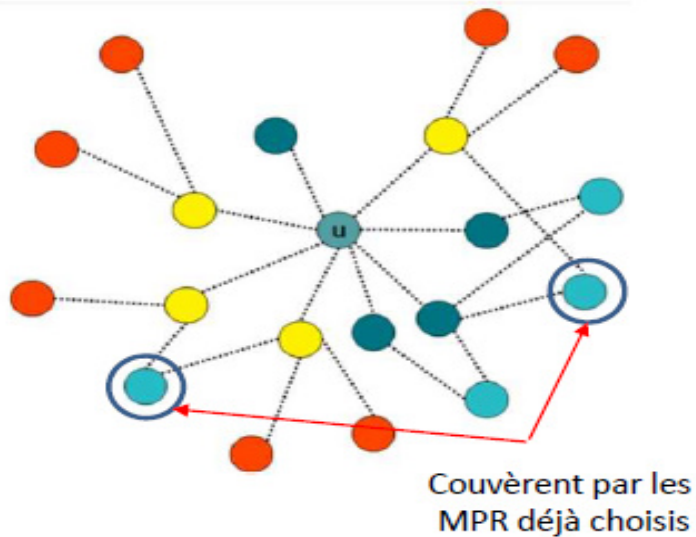
voisin 2-hopes isolé de u

voisin 2-hopes non isolé de u



2

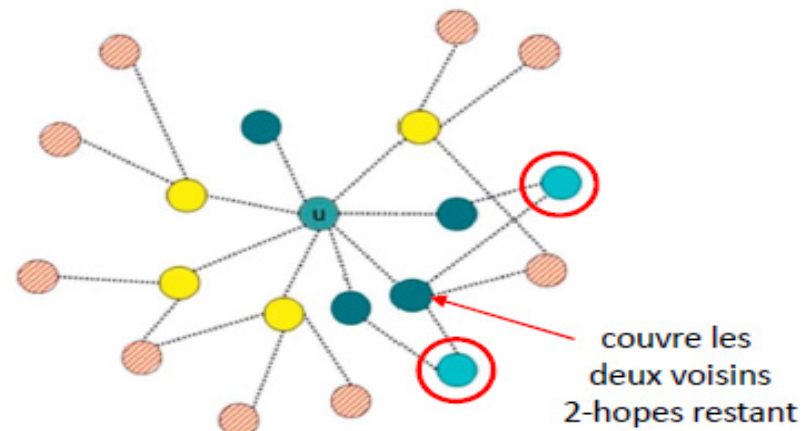
Etp1. les MPR sont en jaune



Couvèrent par les MPR déjà choisis

3

Etp2. Couvrir le reste des voisins 2-hopes



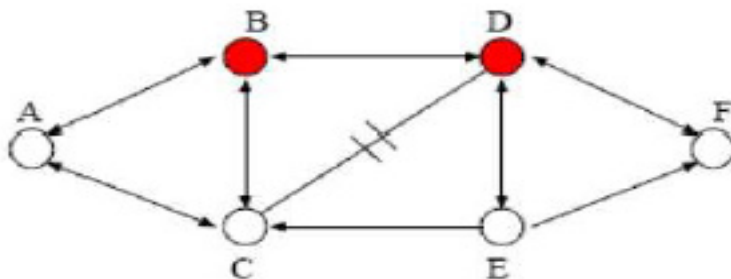
couvre les deux voisins 2-hopes restant

4

OLSR : Principe de fonctionnement

Gestion de la topologie

- Chaque **MPR** déclare ses nœuds *sélecteurs* aux autres *nœuds*
 - Il diffuse **périodiquement** à tous les nœuds des messages **TC- Topology Control**.
 - Le message **TC** contient *< les sélecteurs, les numéros de séquence >*
- A l'aide des **TC**, chaque nœud **maintient** une base qui donne une **vision** globale de la **topologie** du réseau.
- Cette base est constitué d'un ensemble de tuples [**T_dest**, **T_last**, **T_seq**, **T_time**]:
 - **T_dest**: l'@ principale de la destination accessible à un saut du nœud dont l'@T_last
 - **T_last**: un relais multipoint de *T_dest*.
 - **T_seq**: un numéro de séquence
 - **T-time**: le lien est temporaire et peut changer à cause d'un mouvement.



Base topologique de tous les nœuds

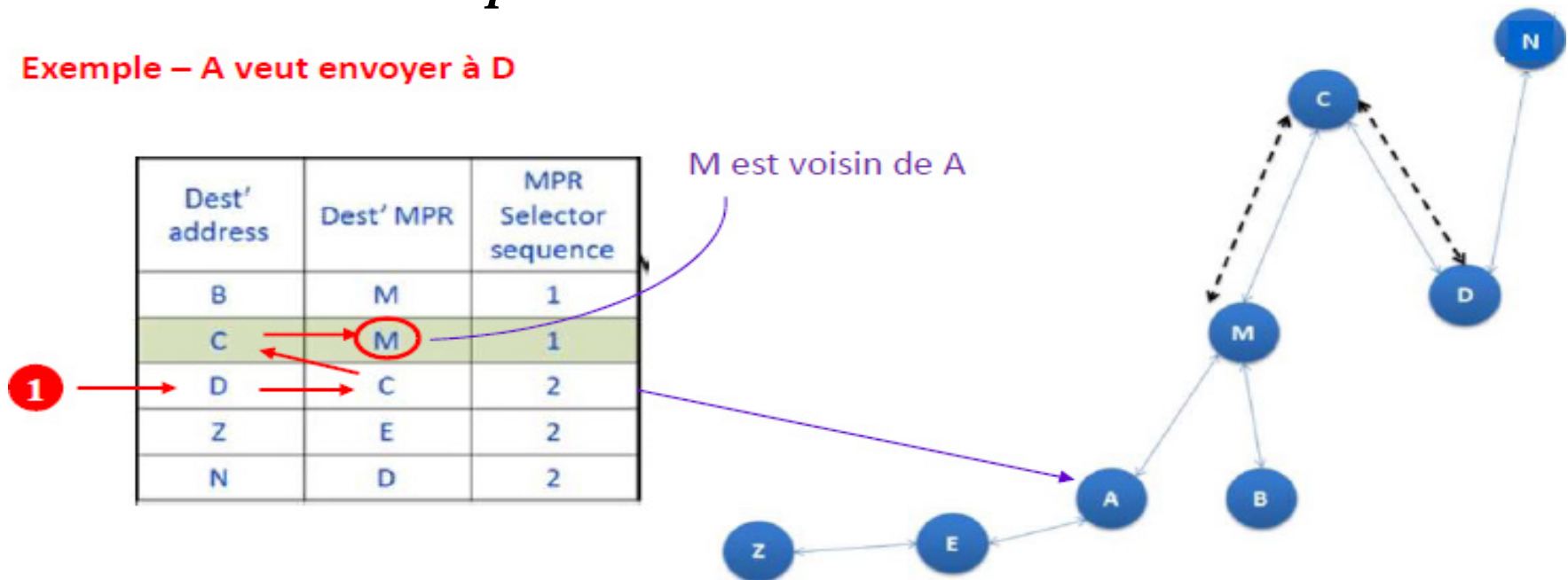
T_dest	T_last	T_seq
A	B	seq-num
B	D	seq-num
C	B	seq-num
D	B	seq-num
E	D	seq-num
F	D	seq-num

OLSR : Principe de fonctionnement

Calcul des routes

- Chaque noeud utilise l'algorithme de *Dijkstra* sur le graphe de topologie partiel construit des:
 - ✓ Liens (m, v): de la **table de voisinage**, v est un voisin symétrique de m.
 - ✓ Liens (last, dest): de la **table de topologie**.
- Les routes sont optimales, sans boucles et les noeuds intermédiaires sont des *relais multipoint*.

Exemple – A veut envoyer à D





OLSR

Avantages

- ◆ La technique des **MPR** réduit sensiblement la **surcharge** due aux messages par rapport à un mécanisme classique d'inondation.
- ◆ Un noeud de la liste des **MPR** doit rapporter seulement les liens qui le relie avec ses sélecteurs.

Inconvénients

- ◆ Une quantité d'information **immense** stockée au niveau de chaque noeud.
 - chaque noeud sauvegarde dans des tables la liste de ses **voisins** MPR, la liste de ses MPRS, la table de **topologie** et la table de **routage**.
- ◆ Le calcul engendré par chaque modification de **topologie** ou du voisinage d'un noeud.



Protocoles de routage basé-source initialisation

Réactifs

- ◆ AODV
 - (Ad Hoc On demand Distance Vector Routing)



AODV (Ad hoc On demand-DV)

- ◆ Établissement des routes **à la demande**
 - Phase (**RREQ/RREP**) pour la recherche d'une nouvelle route
- ◆ Gestion des **pannes** de route (**RRER**)
- ◆ **Pas de boucle** (Loop-free)
 - Utilisation des numéros de séquence **SeqN°** pour chaque nœud.
 - Chaque nœud **incrémente** son **SeqN°** quand la topologie change (changement de ses voisins)
- ◆ Utilisation des liens **bidirectionnels**
- ◆ Utilise deux processus: **Découverte de route** et **maintenance de routes**.



AODV (Ad hoc On demand-DV)

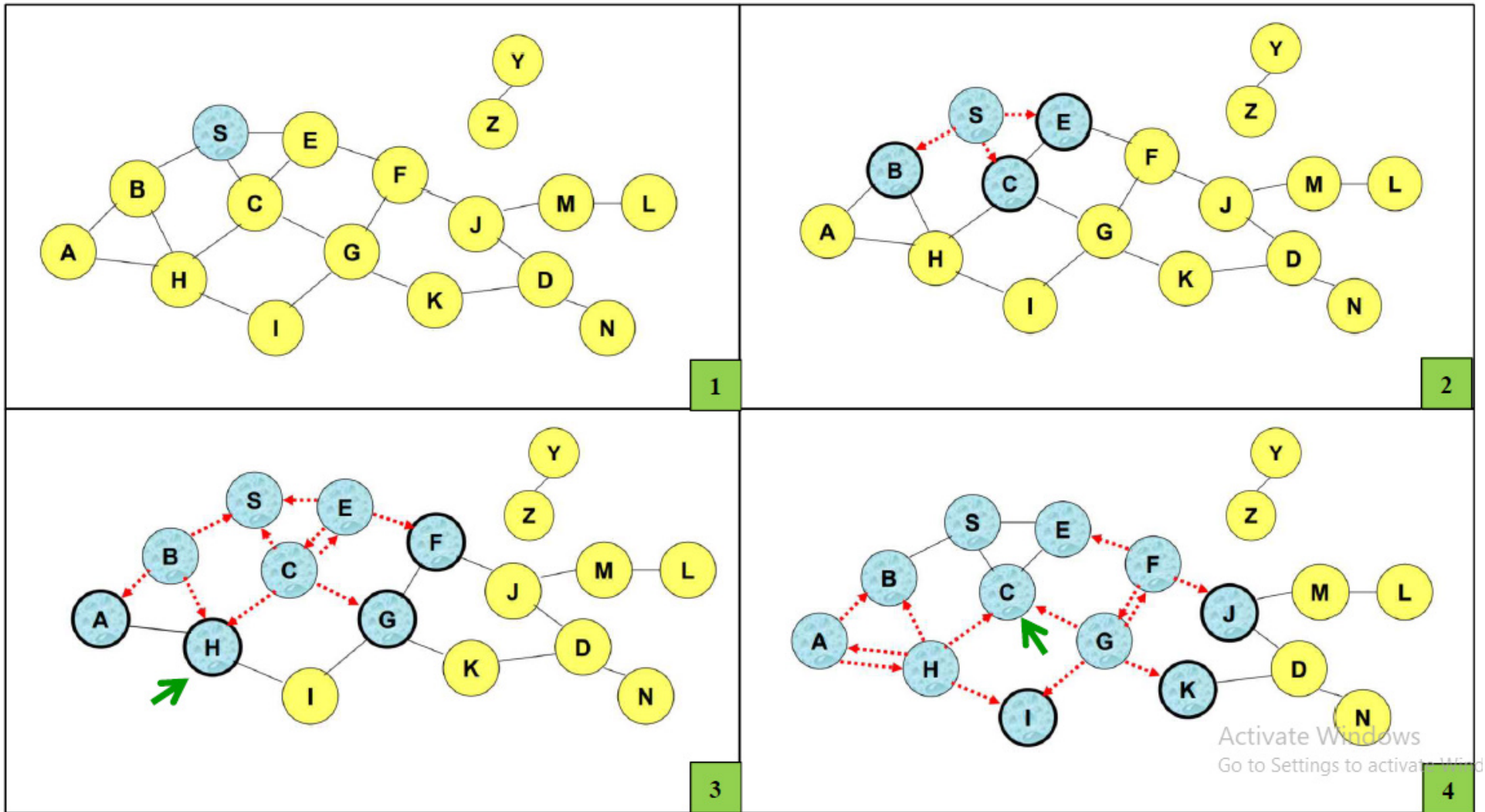
Découverte de routes

- ◆ Le noeud qui souhaite envoyer un paquet vérifie sa table de routage s'il a une **route** vers la destination ou non.
 - Si non, il lance un processus de **découverte de route**.
- ◆ Le processus commence par la diffusion d'un paquet **RREQ** – *Route Request*, par le noeud source
 - La diffusion se fait par **inondation**
- ◆ Le paquet contient:
 - **@IP** du noeud source et destination
 - **Numéro de séquence** du noeud source et destination.
 - **Numéro d'identification** de diffusion (incrémenté pour chaque nouvelle RREQ).
 - **ID** de diffusion + **@IP** source = **Identifiant unique** pour le RREQ.

AODV (Ad hoc On demand-DV)

Découverte de routes

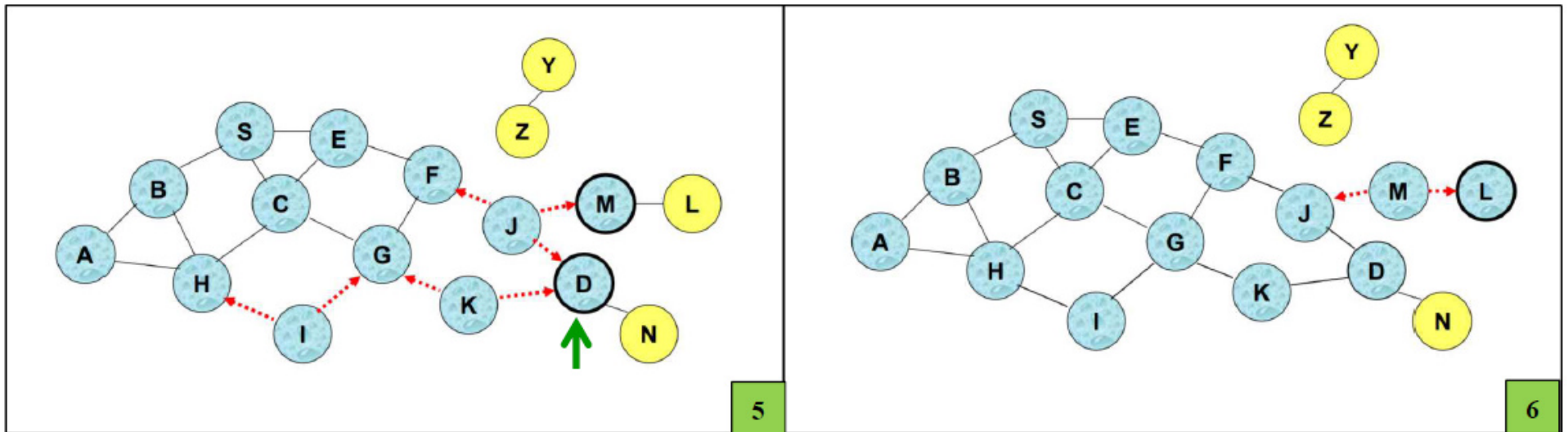
• Envoi RREQ



AODV (Ad hoc On demand-DV)

Découverte de routes

- Envoi RREQ



- ✓ Simplicité.
- ✓ Efficacité dans la recherche de routes (possibilité d'acheminer les données par de multiple routes).
- ✗ surcharge du réseau.

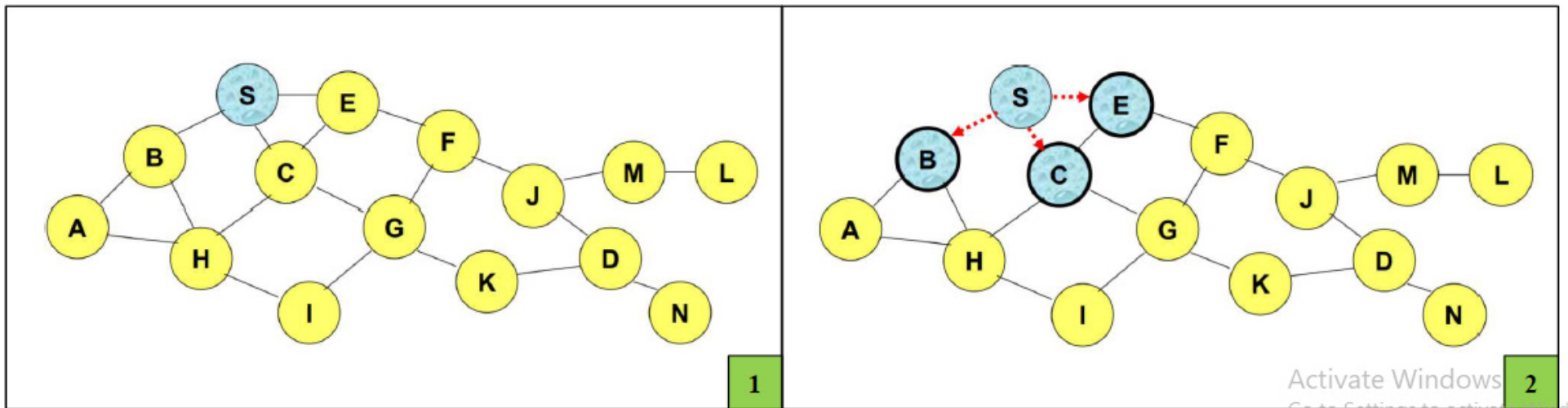
Activate Windows
Go to Settings to activate Wind

AODV (Ad hoc On demand-DV)

Découverte de routes

- **Envoi RREP**

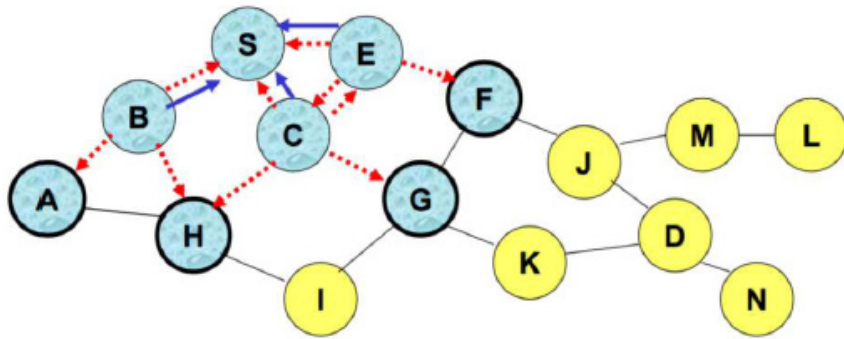
- Chaque nœud qui reçoit un RREQ établit une entrée de route inverse pour le nœud source dans sa table de routage.
 - En utilisant la route inverse, un nœud peut envoyer un RREP (Route Reply paquet) à la source.



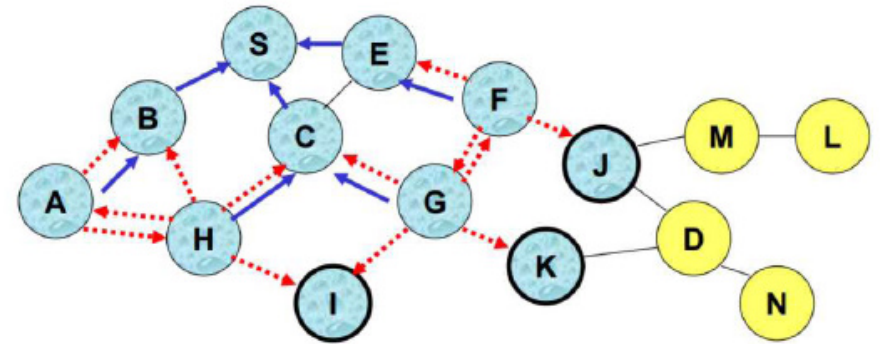
AODV (Ad hoc On demand-DV)

Découverte de routes

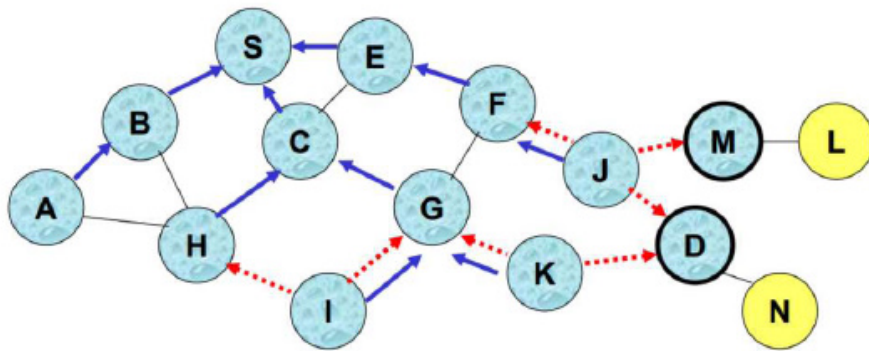
- Envoi RREP



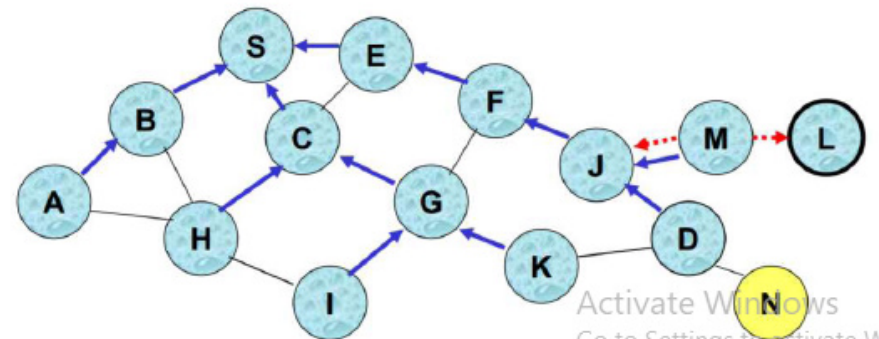
3



4



5



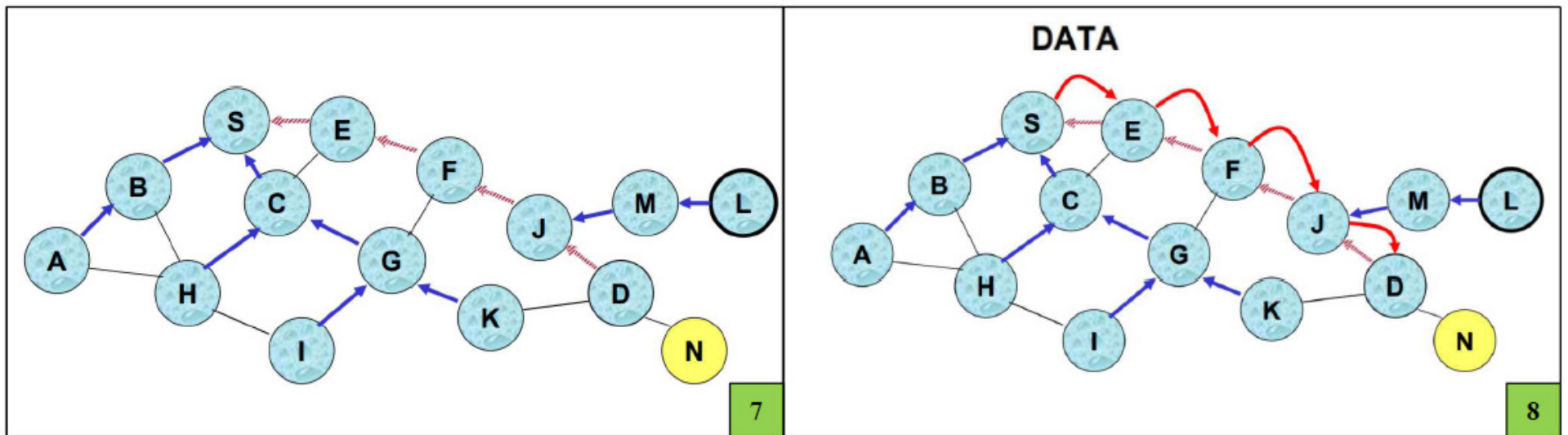
6

Activate Windows
Go to Settings to activate Windows

AODV (Ad hoc On demand-DV)

Découverte de routes

- Envoi RREP





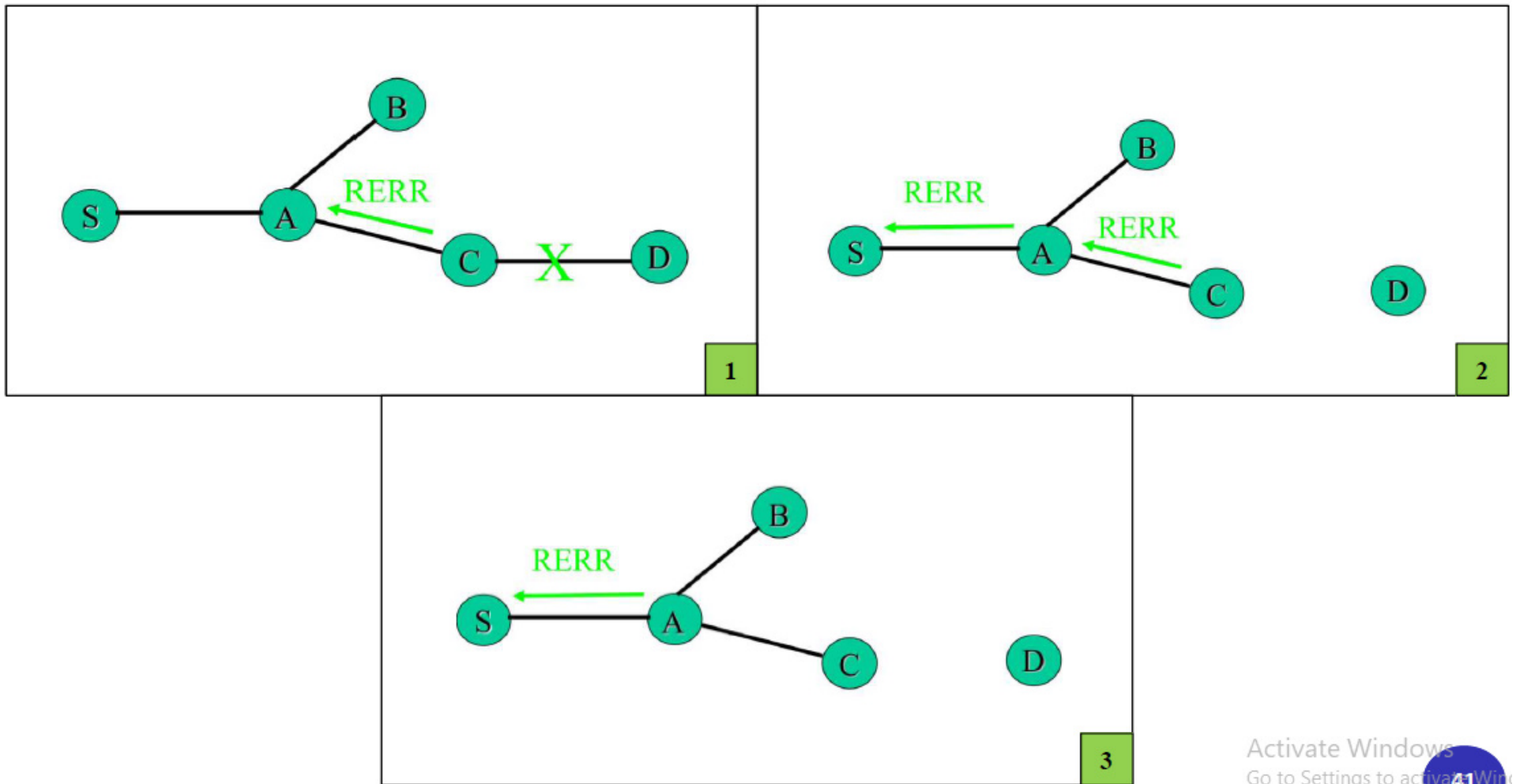
AODV (Ad hoc On demand-DV)

Maintenance de routes

- Une route est **larguée** de la table de routage après un **délai d'expiration**.
 - Maintenance des routes actives seulement.
 - Réduire les voies inutiles et le besoin d'entretien des routes.
- Si le noeud **source** se déplace, un nouveau processus de **découverte** de route est **lancé**.
- Si les noeuds **intermédiaires** ou la **destination** se **déplacent** alors:
 - Les liens du saut suivant se **brisent**, entraînant un échec de liaison.
 - Les tables de routage sont **mises à jour** pour les liens brisés.
 - Tous les voisins actifs sont **informés** par un message **RERR** – **Route ERROr** –
- **RERR** est initialisé par le noeud le plus proche de la source de la **rupture**.
- Un noeud reçoit un **RERR**, marque sa route vers la destination comme **non valide**
- Le noeud source peut relancer la découverte de route à la réception d'un **RERR**.

AODV (Ad hoc On demand-DV)

Maintenance de routes





AODV (Ad hoc On demand-DV)

Propriétés

- Découverte des routes à chaque fois que cela est nécessaire.
 - Implémentation d'une seule route parmi plusieurs déterminées entre source - destination
 - Ne gère pas les routes de chaque noeud vers tous les autres.
- Les routes sont maintenues seulement si nécessaire.
- Chaque noeud maintient l'augmentation de son numéro de séquence.
- Chaque entrée de la table de routage contient, en plus de @ destination et @ de prochain saut, un numéro de séquence de destination et une durée de vie
 - Prévenir les boucles
 - Désigner les routes récentes.
- Mise à jour de la durée de vie à chaque fois que la route est utilisée.
 - Une route qui n'est pas utilisée pendant sa durée de vie -> elle expire.



Quelques bonnes références

- ♦ MANET homepage :
<http://www.ietf.org/html.charters/manet-charter.html>
- ♦ Site du gouvernement américain :
<http://w3.antd.nist.gov/wctg/manet/index.html>,
http://protean.itd.nrl.navy.mil/manet/manet_home.html
- ♦ Les universités :
<http://menetou.inria.fr/olsr/> (olsr)
<http://www.monarch.cs.rice.edu/> (dsr)
<http://moment.cs.ucsb.edu/AODV/aodv.html> (aodv)



Comparaisons de routage ad hoc

- ◆ **Protocoles basé-Table**
 - Complexité en temps : élevé
 - Philosophie de routage : plat, CGSR est hiérarchique
 - Capacité de Multicast : Non
 - #tables requises : élevé

- ◆ **Routage basé-Source Initialisation ou à la demande**
 - Philosophie de routage : plat
 - AODV supporte le multicast
 - N'utilisent pas les messages de contrôle périodiques => préservent la bande passante et la puissance

OLSR: Algorithme (1/2)

- ◆ Chaque nœud calcule son ensemble MPR
 - Envoi un message **Hello** à ses voisins contenant son ensemble MPR,
- ◆ Chaque voisins recevant Hello, met à jour l'ensemble des **nœuds qui l'ont choisi comme MPR** (MS : MPR *selectors*),
 - Chaque message Hello à un **Numéro de Séquence** (âge).
- ◆ La mise à jour des états de liens d'un nœud **x** est **réduite** aux **MPR selectors MS** de **x** (**MS(x)**).
 - Le nœud **x** **ne peut être joint** que par ses MPR



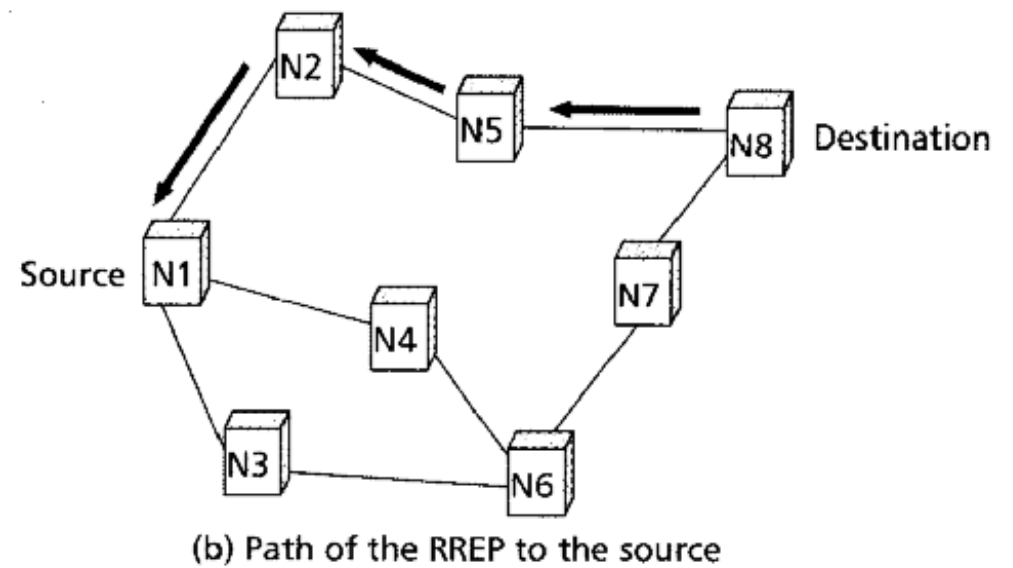
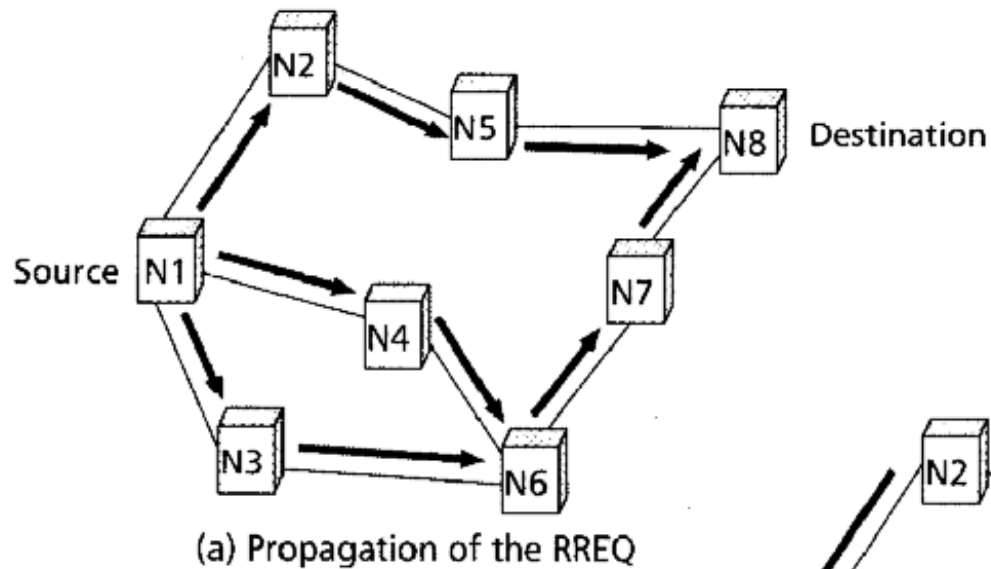
OLSR: Algorithme (2/2)

- ◆ Le routage n'est rien d'autre que **l'inondation périodique** avec des **messages TC (Topology Control)** envoyé par chaque nœud pour déclarer son ensemble MS (avec TTL=255).

→ Tous les nœuds qui reçoivent cette information **mettent à jour** leurs tables de **topologie** et ensuite leur table de **routage**.

→ Un numéro de Séquence **MSSN** pour les MS est attaché à **TC** pour permettre les mises à jours.

Recherche de route AODV





Défis de Recherches: réseaux ad hoc

Défis de Recherches

- ◆ Problèmes de la **couche MAC**
 - Fiabilité de la couche liaison ?
 - QoS dans MAC? Économie de l'énergie ?
- ◆ Problèmes de la **couche réseau**
 - Routage ; (QoS?) Économie de l'énergie ?
 - Multicast ?
- ◆ Problèmes de la **couche transport**
 - Fiabilité de bout-en-bout; Contrôle de congestion? QoS?
- ◆ Problèmes de la **couche application** : Sécurité? QoS?
- ◆ **Interactions** avec les couches; Interconnexion avec Internet;