

Routage dans les réseaux ad hoc

Classification

Réactifs

- Recherche de route sur demande
- Deux phases
 - ✓ Découverte de routes
 - ✓ Maintenance de routes

Hiérarchiques

- Organisation du réseau en cluster

Proactifs

- Maintenance de route
- Echange d'information de topologie

Hybrides

- en général :

- ✓ Pro-actif dans le voisinage d'un nœud
- ✓ Réactif au-delà

Géographiques

- Information de position des nœuds

Routages réactifs

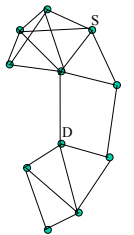
Réactif

Routage par inondation

- **Source diffuse son message à tous ses voisins**
- **Chaque mobile retransmet le paquet reçu**
- **La destination ne retransmet pas le message**

Problèmes

- La diffusion continue après réception par le destinataire
- Tous les mobiles risquent de recevoir le message
 - ✓ Gâchis de bande passante et d'énergie
- Réceptions simultanées
 - ✓ Collision probable (Dépense d'énergie inutile)
- Réceptions multiples, boucles infinies
 - ✓ Utilisation d'un numéro de séquence



Réactif

Dynamic Source Routing (DSR)

Principes :

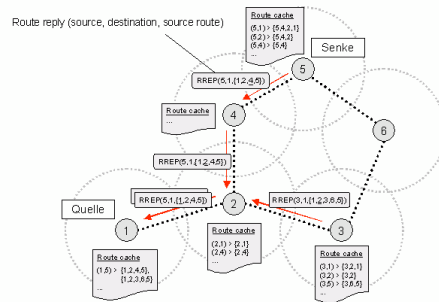
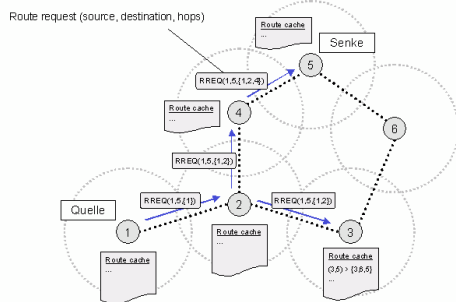
- Routage par la source
 - ✓ La route entière vers la destination est déterminée par la source
- Les adresses des nœuds intermédiaires sont inscrites dans le paquet
- N'utilise pas les paquets hello
- Développé pour des MANET de petits diamètres (entre 5 et 10 sauts) et de mobilité modérée
- Liens bi-directionnels

Réactif

Dynamic Source Routing (DSR)

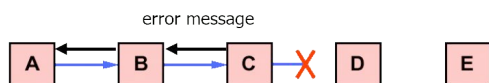
Découverte de route:

- Une source A envoie un paquet *RouteRequest* par inondation.
- Sur réception du *RouteRequest*
 - ✓ Si le nœud B est la destination, il retourne un paquet *RouteReply* à A en incluant le chemin dans le paquet.
 - ✓ Si B n'est pas la destination
 - S'il voit passer le *RouteRequest* pour la première fois, il inscrit son adresse dans le paquet (*Route Record*) et le fait suivre.
 - Sinon, il ignore le paquet.
- Sur réception d'un *RouteReply*
 - ✓ Chaque nœud intermédiaire inscrit la route dans son cache et fait suivre jusqu'à la source.
 - ✓ La source récupère la route et la stocke.



Maintenance des routes :

- Chaque nœud sur la route est responsable du saut suivant sur la route. Chaque nœud doit s'assurer que le prochain nœud sur la route reçoit le message.
- Si au bout de plusieurs essais, le nœud suivant ne reçoit pas le message, un message *RouteError* est renvoyé à la source, qui peut supprimer cette route de son cache.
- La source peut chercher dans son cache une autre route pour la destination ou initier une nouvelle demande de route.



AODV : Ad Hoc On Demand Vector

Norme RFC3561

Principes

- Limitier l'overhead du source routing
- Construire des tables de routage le long du chemin
- Suppose des liens bidirectionnels
- Un nœud intermédiaire peut renvoyer la route s'il la connaît déjà.
- Chaque nœud stocke le prochain saut vers la destination.

Recherche de route

- La source diffuse un paquet RREQ par inondation
- Sur réception d'un RREQ
 - Sans connaissance de la destination
 - Mise à jour du nombre de sauts
 - Mémorisation du nœud précédent
 - Transmission à ses (autres) voisins
 - Connaissance d'un chemin vers la destination
 - Envoi d'une réponse RREP à la source
 - Arrêt de l'inondation
- Sur réception d'un RREP
 - Mise à jour de la table de routage locale
 - Transmission du RREP vers le nœud précédent mémorisé

Routes inverses invalidées si pas de RREP (timeout)

Routes sans boucles

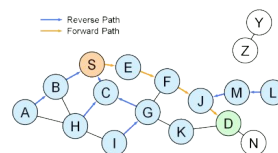
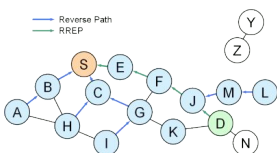
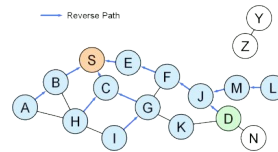
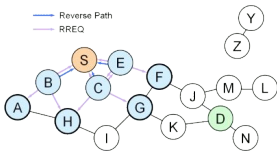
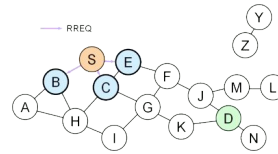
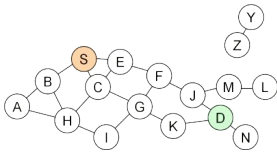
- Utilisation de numéros de séquences

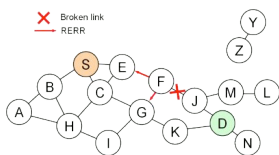
Cassure de lien

- Envoi d'un message RERR à la source qui décide ou non de recommencer suivant le taux d'utilisation de la route

Optimisations : Réparation locale, TTL, ...

- Réparations pro-actives en local
- TTL donne la durée de vie d'une information dans une table
 - Si trop grand, on garde des routes inutilisées
 - Si trop petit : on peut ne pas recevoir le RReply à temps.





© N. Mitton

■ **TORA (Temporary Ordering Routing)**

- Conçu pour minimiser l'effet des changements de topologies
- Stocke plusieurs chemins vers une destination
- Messages de contrôle limités à un ensemble de réduit de nœuds autour du changement de topologie.

■ **ABR (Associativity Based Routing)**

- Basé sur le degré d'associativité (stabilité de connexion en fonction du temps)
- Pas de boucles
- Pas de blocage ni de duplication de paquets

■ ...

Routages proactifs

■ **RFC 3626**

■ **Principes**

- Même base que OSPF (état de liens)
- Diffusion optimisée par les MPR

■ **Recherche de voisinage à 2 sauts**

- Message hello périodique (toutes les 2 s)
- Contenu : listes des voisins directs

■ **Diffusion avec ensemble MPR**

- Tous les nœuds reçoivent les infos
- Seul un sous-ensemble retransmet

■ **Diffusion de l'information**

- Via les MPR
- Par envois périodiques de paquet TC (Topology Control)
 - ✓ Contient la liste des voisins dont je suis le relaie

■ **Construction des tables de routage**

- Plus court chemin (Dijkstra) sur les MPR

■ **DSDV (Dynamic Destination Sequenced Distance Vector)**

- Vecteur de distance (Bellmand-Ford)
- Transmission périodique de toute la table
- Transmission des changements importants de routes
- ⊗ **Pas de boucles ni de distances infinies**
- ⊗ **Lent, beaucoup d'overhead**

■ **WRP (Wireless Routing Protocol)**

- Recherche de route semblable à DSR
- Mise à jour sur changements significatifs
- Maintien de table de routage, table de distances, table des couts des liens
- ⊗ **Vérification de la constance des voisins**

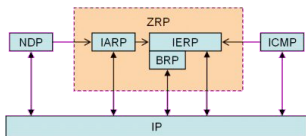
■ **FSR (Fish eye State Routing)**

- États de liens
- Grande précision sur les voisins proches, moins bonne sur les voisins lointains.
- ⊗ **Scalable, volume de données faible, pas de recherche de routes**

■ ...

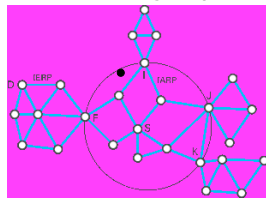
Routages hybrides

Routage hybride



■ Plus résistant à la mobilité avec de grande zone

- **IARP** : proactif simple
 - Découverte des voisins (msg hello)
 - État des liens envoyé à tous
 - Calcul table de routage
- **IERP** : réactif
 - Utilise un arbre multicast (BRP)
 - Si D hors zone, recherche par les nœuds périphériques



Routages géographiques

Routage géographique Positionnement

- **Sans mesure physique**
 - Proportion de surfaces de communication communes
 - μ -distance : Approximation des distances par le nombre de voisins
 - Centroïde (barycentre sur l'enveloppe convexe des voisins)
- **Triangulation (angle), Trilatération (distance)**
 - ✓ Min-Max (area based – carrés)
 - ✓ APIT (area based - triangles)
 - ✓ DV-Hop (dstce = nb sauts)

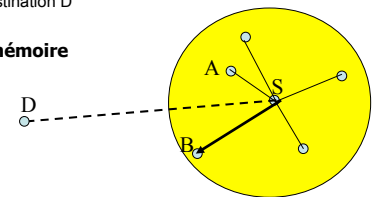
- **Avec mesure physique**
 - GPS embarqué
 - Temps d'arrivée (ToA), déviation temporelle (TDoA)
 - Angle d'arrivée (AoA)
 - Puissance du signal reçu (RSSI)

Routage géographique Algorithmes gloutons

■ **Glouton** : choix du nœud le plus proche de la destination

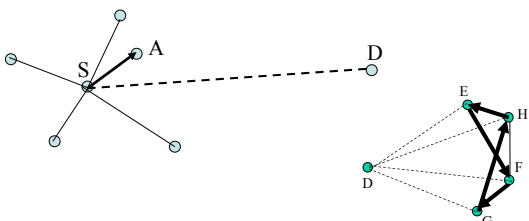
- **la source connaît**
 - Sa propre position
 - La position de ses voisins
 - La position de la destination D

■ **Algorithmes sans mémoire**



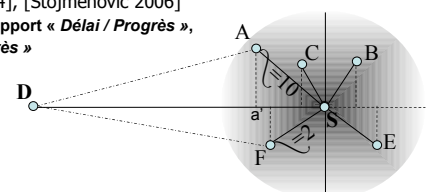
Routage géographique Algorithmes gloutons


- **DIR** : angle le plus proche de SD
 - Minimise la direction
- ⊗ Il existe des boucles !



Routage géographique Algorithmes gloutons

- **Random progress** : A, C or F
 - Le hasard fait bien les choses
- **NFP**, Nearest Forward Progress : C
 - On économise l'énergie
- **MFR**, Most Forward within Radius : A
 - Minimise le nb de saut
- [Datta et al. 2004], [Stojmenovic 2006]
 - Minimise le rapport « Délai / Progrès », « Coût / Progrès »

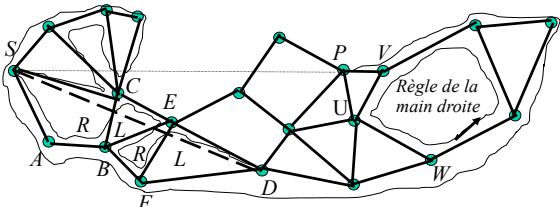




IRCICA

Routage géographique

Algorithme glouton





[Bose et al. 1999]

- Nécessite un graphe planaire
- Route dans le sous-graphe planaire
 - SABCEBFED
 - SC...ABFD...W...VP
- Garantie de livraison

J. Carle

Master TAC

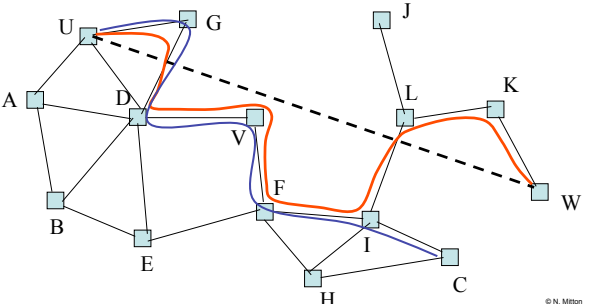




IRCICA

Routage géographique


DIR, FACE




© N. Mitton

J. Carle

Master TAC

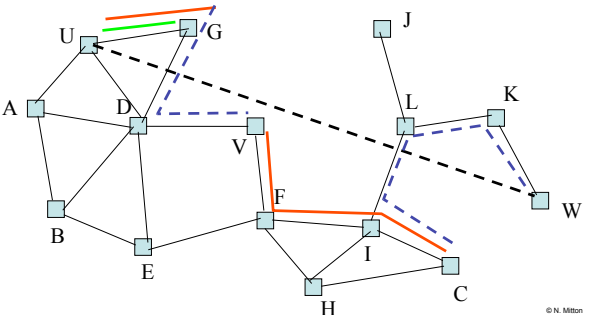




IRCICA

Routage géographique

Greedy, GFG (greedy-face-greedy)



© N. Mitton

J. Carle

Master TAC

