

## Résumé de l'article de Rahul C. Shah et Jan M. Rabaey Energy Aware Routing for Low Energy Ad Hoc Sensor Networks

Cet article traite des différents aspects du projet PicoRadio. Ce projet consiste à utiliser un réseau de capteurs (PicoNodes) pour gérer la consommation énergétique d'un bâtiment. Ces PicoNodes sont des petits capteurs de moins d'un centimètre cube qui coûtent moins d'un dollar.

Le début de cet article traite des couches physique, liaison et réseau des PicoNodes. La couche réseau s'occupe des adresses des capteurs. L'auteur a choisi de former ces adresses par le triplet suivant : <Position géographique, Type de nœud (capteur, contrôleur, déclencheur), Sous-type de nœud (capteur de température, humidité...)>.

Toute la deuxième moitié de l'article traite des protocoles de routage. Il existe deux types de protocoles : les proactifs et les réactifs. Les protocoles proactifs font en sorte que tous les nœuds aient toujours une vision correcte de l'ensemble du réseau. Ainsi, tous les changements de topologie sont immédiatement transmis à tout le réseau. Exemples de protocoles proactifs : DSDV (Destination-Sequenced Distance-Vector Routing protocol) [4] ; Link-state Routing [2]. Les protocoles réactifs, quant à eux, ne découvrent un chemin entre deux nœuds qu'en cas de besoin. Certains protocoles initient la découverte d'un chemin depuis la source vers la destination ; d'autre partent de la destination vers la source. Exemples : AODV (Ad-Hoc On-Demand Distance Vector Routing) [5] ; DSR (Dynamic Source Routing) [3] ; Directed Diffusion [1].

L'auteur propose un nouveau protocole de routage : Energy Aware Routing. Ce protocole est réactif et les requêtes sont initiées depuis la destination. Il se décline en trois phases : la phase d'initialisation qui permet de trouver tous les chemins de la source à la destination ; la phase de communication de données ; et la maintenance des routes. Les deux premières phases sont détaillées dans les algorithmes ci-dessous.

## Références

- [1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *Networking, IEEE/ACM Transactions on*, 11(1) :2–16, 2003.
- [2] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International*, pages 62–68. IEEE, 2001.
- [3] D.B. Johnson, D.A. Maltz, J. Broch, et al. Dsr : The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 5 :139–172, 2001.
- [4] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24(4) :234–244, 1994.
- [5] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on*, pages 90–100. IEEE, 1999.

---

**Algorithm 1** Setup phase

---

$Cost(N_D) = 0$   
{Flood the network from  $N_D$  to  $N_S$  :}  
**for** each intermediate node  $N_i$  **do**  
  **for** each neighbor  $N_j$  of  $N_i$  **do**  
    **if**  $d(N_i, N_S) \geq d(N_j, N_S)$  and  $d(N_i, N_D) \leq d(N_j, N_D)$  **then**  
      Forward the request from  $N_i$  to  $N_j$   
    **end if**  
  **end for**  
**end for**

**for** each intermediate node  $N_i$  **do**  
  **for** each neighbor  $N_j$  of  $N_i$  **do**  
    On receiving a request :  
     $C_{N_j, N_i} = Cost(N_i) + Metric(N_j, N_i)$   
     $FT_j = \{i | C_{N_j, N_i} \leq \alpha(\min_k C_{N_j, N_k})\}$   
$$P_{N_j, N_i} = \frac{\frac{1}{C_{N_j, N_i}}}{\sum_{k \in FT_j} \frac{1}{C_{N_j, N_k}}}$$
  
     $Cost(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i}$   
  **end for**  
**end for**

---

---

**Algorithm 2** Data communication phase

---

$N_S$  chooses randomly a number  
This number points a neighbor in the forwarding table  $FT_S$  out  
Send the packet from  $N_S$  to this neighbor  $N_j$   
Do the same thing for all intermediate nodes until the packet reaches the destination

---