MobileR : Multi-hop energy efficient localised mobile georouting in wireless sensor and actuator networks

Nicolas Gouvy, Nathalie Mitton

MobileR: Multi-hop energy efficient localised mobile georouting in wireless sensor and actuator networks

Nicolas Gouvy[1] and Nathalie Mitton[2]

[1] Univ Lille Nord de France, USTL, CNRS UMR 8022, LIFL, INRIA, France
`nicolas.gouvy@lifl.fr`,
[2] INRIA Lille-Nord Europe, Univ Lille Nord de France, USTL, CNRS UMR 8022, LIFL, France
nathalie.mitton@inria.fr

Abstract. This paper addresses the usage of actuators (sensors with controlled mobility) for routing in wireless sensor and actuator networks. Different routing protocols have been proposed to improve routing in terms of energy efficiency through the use of controlled mobility enabled sensors . We introduce MobileR. Unlike literature proposals also using actuators, MobileR considers the cost of a full path toward one of its neighbours instead of the cost of the direct edge toward it. To do so, MobileR computes in advance the possible routing paths over the next hops relying on the one-hop neighbours and their possible relocations. Moreover MobileR is fully localised and stateless. We evaluate our solution in terms of cumulative energy consumption with regard to network density. Experiments show that, with sufficient node degree, energy used for routing is significantly reduced and so network lifetime is extended.

Key words: mobile sensor network, localised algorithm, controlled mobility routing, energy efficiency, connectivity preservation, multi-hop path computation

1 Introduction

Wireless sensor networks (WSN) are characterised with the use of constrained devices powered by batteries and with low computational power and very little memory. As a consequence it is essential to propose routing algorithms which take into account those limitations in order to improve network lifetime. Considering that radio transmission is the most energy consuming factor in WSN, a common solution is to increase sensor density in order to decrease communication range. A more recent approach is to introduce mobile sensors, since it has been proven [3] to be as efficient as the density increasing. However, only a handful of routing protocols ([8, 5]) relies on the usage of mobile sensors in order to extend the global network life. Those protocols are geographical routing protocols with cost-over-progress (COP) [12] heuristics. They are position-based,

and compute possible relocations of each node on the routing path.

In this paper we propose MobileR routing protocol. MobileR is also a position based COP geographic routing protocol. However, unlike previous solutions, MobileR does not select the best node to forward a packet in the same way. As in [10], our solution evaluates at each step of the routing all possible multi-hop paths over its neighbourhood toward the destination. Moreover MobileR takes into account mobility deeper than any other routing protocol. It computes all associated relocations on each multi-hop path and their consequences in term of energy. MobileR comes in two variants, $MultipleORouting$ and $MultipleMove_R$, similarly to [8]. These variants differ in the relocation pattern they use, in order to permit further energy savings. Each variant fits better some energy models. Both variants respect the following properties:

– Localised: routing decision is taken only based on neighbour and destination location
– Scalable: routing process is stateless which allows it to be transposable to large networks as no information is stored neither in the message nor in the network nodes.
– Loop free: routing loops are avoided in order to protect network lifetime
– Energy efficient: MobileR takes into account all the costs associated with the routing process, transmission and moving costs. Transmission cost *and* displacement cost are evaluated with regard to the geographic progress.

The paper is organized as follows. Section 2 presents the related works. Simulation models are detailed in Section 3. We present the principle of MobileR in details in Section 4. Section 5 exposes two different heuristics and their uses in MobileR. Performance evaluation results are presented in Section 6. Finally, we conclude our work in Section 7.

2 Related Work

Position based routing. Those routing algorithms rely on the assumption that every node is aware of its own location, the location of its one-hop neighbours, and the location of the destination of the message. This made possible greedy routing [1] in which the message to route is forwarded to its neighbour which is the closest to the destination. It has been further optimized with COP [12] routing. It ensures that the ratio of energy used for sending a message to a node to the measured reduction in distance to the destination is minimized.
Other routing algorithms were proposed which ensure the packet delivery, like GFG [4], LOAFR [9] or ETE [10]. LOAFR chooses the first node being closer to the destination than the current one, this node is a relay node. Then LOAFR computes an energy shortest path (using Dijkstra) between the current node and the relay. Message is accordingly modified to include the computed path, and forwarded along that path until it reaches relay node. And so on. Recovery is done through Face-Routing. ETE is a routing algorithm which computes

the energy weighted shortest path (SP) over the one-hop neighbourhood in the forward direction. In ETE computed path is not forwarded. Loops are avoided by considering in the path computing only nodes in the forwarding direction of the destination. ETE makes each node on the routing path compute its own SP according to its neighbourhood knowledge, as it allows further optimization on each step. So only the data is forwarded to the first node on the SP.

Routing in mobile sensor networks. The topic of mobility is a main concern in ad hoc and wireless sensor networks since the beginning of this research topic. But most part of the extensive investigation on this topic focused on adapting to *consequences* of mobility. The first attempt of routing with the use of *controlled* mobility is the introduction of a mobile sink which gathers data from the sensors by one-hop radio communication [13]. The mobile sink then carries the data to a fixed base station. This approach requires the nodes to buffer their data between each visit of the sink. Same authors also propose a multi-sink scenario in which each sink moves randomly and collects data from sensors in its range.

A handful of more recent approaches considers the cases in which all sensors are mobile sensors (robots or actuators) or at least a significant part. Most of them [5, 6] use an existing protocol to find an initial path, and then move each node on this path to the midpoint of its predecessor and successor on the routing path. MobileCOP [5] relies on three step. *1)* It builds a path from the source to the destination, electing at each node the next hop on the route in a COP fashion. *2)* Once the destination has been reached, a route confirmation packet is sent from the destination to the source, relocating nodes on the path at this moment by placing them on the line from source to destination and equidistantly placed. *3)* Finally, the data is sent along the pre-defined path. A major drawback of this approach is that the path has to be memorised to allow further relocation of nodes. And it may cause node's memory overhead. Moreover this routing algorithm does never take into account neither the moving cost nor the network connectivity, and so can disconnect the network. However, relocation pattern in [6] may cause useless zig-zag movements.

The CoMNet [8] routing algorithm is localised and stateless. It is COP fashioned, with both sending cost and relocation cost included. CoMNet comes with three relocation patterns: *1)* CoMNet *ORouting On the Move* which aims to align nodes on the routing path on a virtual line between source and destination, *2)* CoMNet $Move_{(r)}$ which equally spaces nodes on the routing path with the optimal radio range *3)* CoMNet $Move_{DSR}$ which aligns nodes and spaces them according to the optimal number of hops of the path at once. CoMNet relies on the use of a Connected Dominated Set (CDS) to guarantee network connectivity.

Connectivity preservation Mobility is a double-edged sword. If it can be used to improve routing, mobility must be used with caution since it may provoke network disconnection through the relocation of one single node. In CoMNet [8] -and MobileR- connectivity of the network is guaranteed through

the computation of a Connected Dominated Set (CDS). If a node determines that it belongs to the CDS, it will never move. Those nodes, called dominant, are used to define a "skeleton". The other nodes, called dominated, are free to move while they remain in transmission range of at least dominant. In other words, mobility is only allowed for dominated nodes and under a one hop radio range of the skeleton. This mechanism ensures that there is always a path from one node to another in the network whatever the movements of the nodes.

3 Models

System model

The system is composed of a set of wireless sensor nodes. Each of them can be either a static sensor or an actuator. In our context, we call a static sensor a sensor which can not move by itself but can communicate through a radio device. An actuator is a wireless sensor node which can communicate through radio and moreover can move in response to the reception of a displacement order in a controlled way. Furthermore, each node of the network, either static sensor or actuator, is capable to adjust its transmission range between 0 and R to save energy. All nodes are aware of their location through an hardware device such as GPS or any other location mean.

Notation

Let $N(u)$ be the neighbourhood of node u: the physical set of nodes which are in the transmission range of node u. We also define as $N_d(u)$ the subset of $N(u)$ in which each node is closer to node d than u itself, *i.e.* such that:

$$N_d(u) = \{v \in N(u) \text{ such as } dist(v,d) \leq dist(u,d)\} \tag{1}$$

Node's X relocation is noted as X'.
We define here a set of functions used in the following:

location() : returns the location where a node should move, based on MobileR variant,
dist() : gives the Euclidean distance between two nodes location.
isCoveredByDominant() : returns TRUE if a location is in the radio range of a dominant node, FALSE otherwise.

Energy costs

It is worth noting that MobileR is energy model independent. Nevertheless, in order to measure MobileR performance, we use the most common energy model to compute the cost of the radio transmission of a packet, such as defined in [2]:

$$cost_s(r) = c + r^\alpha \qquad (2)$$

where r to the Euclidean distance the signal has to cover, c corresponds to the activation cost of the radio device, and where α is a real constant ($1 < \alpha < 8$) which represents the signal attenuation.

The moving cost model we employ is the one used in previous similar works [5]:

$$cost_m(r) = a \times r \qquad (3)$$

in which r is the Euclidean distance between the actual actuator location and the location the actuator is ordered to reach (*i.e.* the distance $dist(A, A')$ the actuator has to travel). a is a numerical constant determined by the engine used.

4 MobileR general idea

Unlike previous solutions such as [5] and [8], MobileR does not simply compute the best node toward the destination and relocate it before routing. On the contrary, MobileR computes recursively all possible paths through relocations over the one hop neighbours in forward direction. Those computations aim to anticipate all possible routing paths over next hops toward destination. In Figure 1 node S computes the shortest path toward each of its neighbour, taking into consideration only nodes A,C and B which are towards D, in order to avoid routing loops. Transmission and relocation costs relative to each path are considered with regard to the total progress in a COP-fashion. Relocation of a node is done accordingly to a relocation pattern, and only if possible. MobileR can use any relocation pattern, but does always use the same over a network. MobileR selects the best path (BP) computed : the path which minimizes the COP.

Assuming that $x_0 x_1 ... x_i x_{i+1} .. x_n$ is the node sequence on the computed path from $a = x_0$ to $b = x_n$, its cost can be defined as follows:

$$cost(a, b) = \sum_{i=0}^{n} (cost_s(x_i, x'_{i+1}) + cost_m(x_{i+1}, x'_{i+1})) \qquad (4)$$

where x'_i or x'_{i+1} are the new positions of respectively node x_i and node x_{i+1}, while the progress of this path is defined as:

$$progress = dist(a, D) - dist(x_n, D) \qquad (5)$$

However MobileR does not communicate the BP from one node on the to another. It forwards the message to route to the first node on the path without any computed information about BP. It allows each node on the routing path to compute its BP toward the destination according to its neighbourhood knowledge. So, only one node may move at each set of the routing process, even if computed BP includes multiple nodes relocation.

We first describe the MobileR routing algorithm and then illustrate it through an example. MobileR makes a node x_s which has a message to route consider

(a) Node S and its neighbourhood   (b) Relocation   pattern   CoMNet
$ORouting$

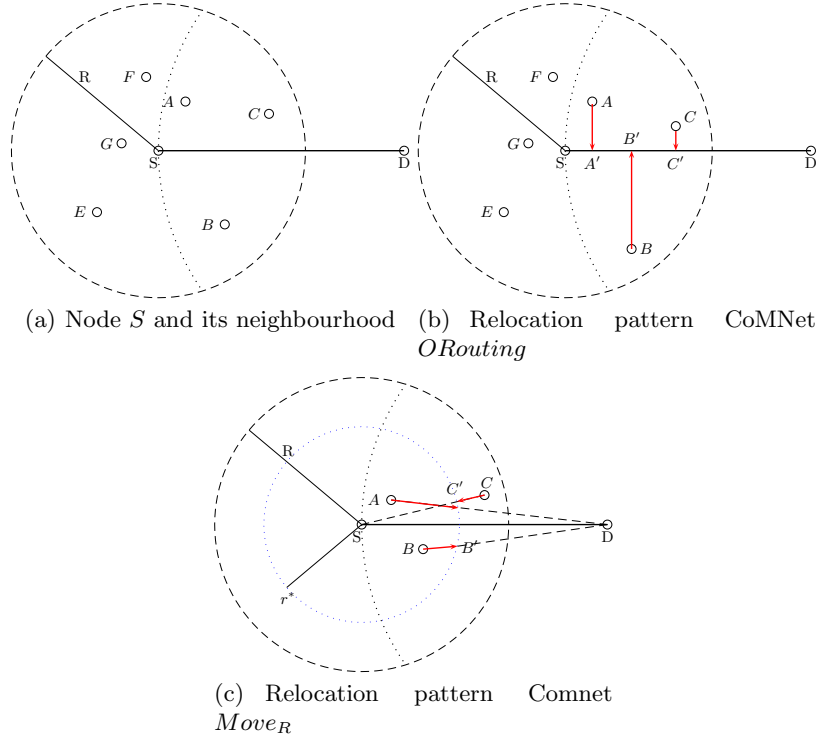(c) Relocation   pattern   Comnet
$Move_R$

Fig. 1. Illustration of relocation patterns for a routing from S to D

its neighbours in the forward direction. In first step, starting from current node position location(x_s), it tries to virtually relocate one of its neighbour x_a. The relocation of x_a is in x'_a according to the relocation pattern used. And so MobileR has virtually built a path $x_s x'_a$ and can compute its associated cost-over-progress. Recursively, MobileR tries to extend the path $x_0 x'_a$, starting from its end, the x'_a location. As before, MobileR chooses a common neighbour of x'_a and x_s in forward direction, for example x_b and relocate it in x'_b. It makes possible the COP evaluation of path $x_s x'_a x'_b$. And so on, with each possible neighbour at each step, computing all possible paths starting from x_s. It is important to notice that x_s -which runs MobileR - does never know the real x'_a neighbourhood. It only knows a subset of them: the subset of its (*i.e* node x_s) neighbours which are closer to the routing destination than x'_a. With destination D it can be noted as:

$$N_D(x'_a) = N_D(x_s) \cap \{v \in N(x'_a) \text{ such as } dist(x'_a, D) \leq dist(x_s, D)\}$$

This enables possible energy saving by anticipating the routing over possible few hops only with the one hop knowledge.

Let's illustrate MobileR with an example, linked to Algorithm 1. Considering Figure 1, MobileR will act as follows. In this particular case source node S has to send a data packet to destination node D. The first step of MobileR is to make S compute the subset of nodes of its neighbourhood closer to the destination D than itself: $N_D(S)$ (Alg. 6, Line l6), *i.e.* in our example $N_D(S) = \{A, B, C\}$. If this set is empty, the greedy routing fails (Alg. 1 Line l4). Following a relocation pattern ($MultipleORouting$ or $MultipleMove_R$), S then computes the relocation of each node belonging to $N_D(S)$ (Alg. 1 Line l7 / Alg. 2).

Algorithm 1 $Compute(s, d)$

1: $x \leftarrow s, min \leftarrow \infty$
2: **while** $x \neq d$ and $ok = 1$ **do**
3:     **if** $N_d(u) \neq \emptyset$ **then**
4:        Routing error
5:     **else**
6:        **for all** $v \in N_d(u)$ **do**
7:           $C \leftarrow rCompute(location(s), 0, v, s, d)$
8:           **if** $C < min$ **then**
9:              $min \leftarrow C, x \leftarrow$ first node on $BP(x, v)$
10:           **end if**
11:        **end for**
12:     **end if**
13: **end while**
14: **return** x

A will be virtually relocated in A', B in B' and C in C' with radio and displacement costs computed (Alg. 2, l1-2). S then computes the COP of this $S \leftarrow A'$ path. Then, using A' new location, MobileR makes S do a recursive path computation: A' is the new start of the path. $N_D(A')$ is computed, but according to the neighbourhood knowledge of S (Alg. 2 line l6).

$$N_D(A') = N_D(S) \cap \{v \in N(A') \text{ such as } dist(A', D) \leq dist(S, D)\}$$

MobileR so makes S compute relocation of nodes in this set (Alg. 2 l7) in a recursive way in order to extend the $S-> A'$ path if possible. And so on. In Figure 1(c) the path $S-> A'-> C'$ can be built.

However, if for any reason (node is dominant, node is not a mobile sensor, relocation of a node is not covered by a dominant) a node x_i can't be moved, $x_i' = x_i$ as shown in Algorithm 3.

S finally selects the path which minimizes the COP (Alg. 2, l8-9 / Alg. 1, l8-9). This BP is the best possible path in terms of COP, based on S knowledge. S then forwards the packet to the first node on the BP, like in [10]. None other information previously computed (Alg. 1-l14) is sent to the next hop. It makes each node routing the message to compute its own BP based on its neighbourhood knowledge.

Algorithm 2 $rCompute(prevNodeReloc, prevCost, nodeToConsider, origin, dest)$

1: $sCost \leftarrow cost_s(dist(prevNodeReloc, location(nodeToConsider)))$
2: $mCost \leftarrow cost_m(dist(location(nodeToConsider), relocation(nodeToConsider)))$
3: $cost \leftarrow sCost + mCost + prevCost$
4: $progress \leftarrow dist(origin, d) - dist(relocation(nodeToConsider), d)$
5: $min \leftarrow \frac{cost}{progress}$
6: **for all** $z \in (N_d(relocation(nodeToConsider)) \cap N_d(origin))$ **do**
7:     $C \leftarrow rCompute(relocation(nodeToConsider), cost, z, d)$
8:     **if** $C < min$ **then**
9:         $min \leftarrow C$
10:     **end if**
11: **end for**
12: **return** min

Algorithm 3 $relocation(nodeToConsider)$

1: $x \leftarrow location(nodeToConsider)$
2: $x' \leftarrow new\ position\ according\ to\ relocation\ pattern$
3: **if** $(isCoveredByDominant(x')$ AND $(!isDominant(nodeToConsider))$ **then**
4:     **return** x'
5: **else**
6:     **return** x
7: **end if**

MobileR so computes a multi-hop path over it's one neighbourhood toward the destination of routing. It attempts to optimize routing by trying to figure out what would be the routing path over the next hops.

In order to improve energy efficiency in the routing process, two different relocation patterns have been considered (Figure 1). Each variant has its own advantages according to the energy consumption model and/or the application. On the one hand MobileR-$MultipleORouting$ relocates the first node on BP (with a beacon) before forwarding the message. This mechanism is used in order to reduce transmission costs by reducing radio range before sending message. On the other hand, MobileR-$MultipleMoveR$ sends the relocation order with the message. It aims to reduce moving distance costs while optimizing the sending cost.

5 MobileR Variants

5.1 MobileR-$MultipleORouting$

Illustrated in Figure 2, this heuristic aims to align all nodes on the path from the source to the destination as it provides the best energy saving ([8],[5]) for sending. The relocation principle here consists in relocating a node to its orthogonal projection on the source-destination virtual line since the orthogonal projection

is the shortest distance to travel.

The relocation order is sent first through a beacon. And then, when relocation of the neighbour is done, the message is sent. We consider that a message is significantly bigger than a beacon. With a negligible beacon cost -compared to the message-, the transmission costs are reduced since the relocated node is closer than before. The cost of each path is composed by the relocation cost of each node on the path (from X to X') and the transmission cost between each node on the path (from X' to Y' for instance). Figure 2 illustrates all the possible logical
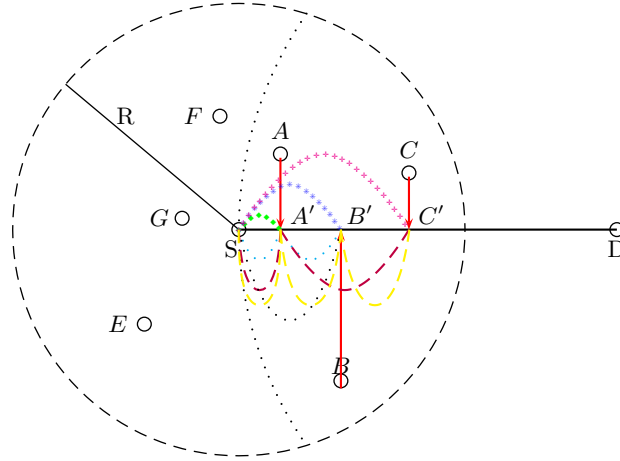
Fig. 2. Illustration of relocations in MobileR-*MultipleORouting* (red arrows). Some of the possible paths considered between S and the relocations of its neighbours are also depicted.

paths. The path from S to D by A' is direct $S \rightarrow A'$ since there is no possibility to have any energy shorter path. There are only two possibilities to use B: $S \rightarrow B'$, or $S \rightarrow A' \rightarrow B'$ since A is closer to S than B. All possible paths from S to C' are the following ones: $S \rightarrow C', S \rightarrow A' \rightarrow C', S \rightarrow B' \rightarrow C', S \rightarrow A' \rightarrow B' \rightarrow C'$. At each step of path computations, only nodes which are closer to the destination than the last one on the current computed path are considered to avoid routing loops.

5.2 MobileR-*MultipleMove$_R$*

An illustration can be found in Figure 3. This relocation pattern relies on the idea to equally space each node on the path. Moreover, nodes' relocation is such that the distance between two nodes on the path is equal to the optimal radio range of their transmission device [8]. This range is extracted from the common energy model for radio transmission $cost(r) = r^\alpha + c$ and is equal to $r^* = \sqrt[\alpha]{\frac{c}{\alpha-1}}$ [11]. In this variant, relocation order is added to the message. As a consequence,

relocation is done after message reception.

The relocation of a node X is on the intersection point of the X-destination virtual line and the circle $C(S)_{r^*}$ of radius r^* centred in S if $|SX| \leq r^*$. Otherwise, relocation is on the intersect between $C(S)_{r^*}$ and SX.
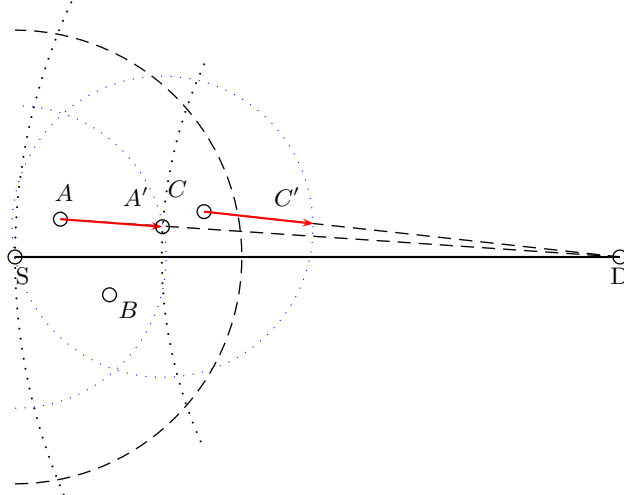

Fig. 3. Relocation principle MobileR-$MultipleMove_R$ (red arrows). The relocation of A in A' makes possible the relocation of C in C' and as a consequence the evaluation of the path $S \rightarrow A' \rightarrow C'$

Figure 3 illustrates a step in the path computations, and demonstrates the possible relocations of a node engendered by the relocation of another. Furthermore, the higher the $\frac{r^*}{maximal\ range}$ ratio is, the further the path computation is. And so a node can predict more accurately the optimal path toward the destination on multiple hops. In this heuristic, the message is sent *with* the relocation order in it.

6 Performance Evaluation

Simulations are performed using WSNet network simulator[1]. Since we focus on the network layer, simulations are conducted with no interferences and/or collisions. On the network point of view, nodes are deployed uniformly on a 1000×1000 square topology. Each of the node is able to tune its transmission range R between 0 and 250 and can obtain their location precisely on the simulation map. We use the algorithm found in [7] for CDS computation. This algorithm is fully local and only requires knowledge of one hop neighbourhood. Nevertheless any CDS computation algorithm is possible.

---

[1] http://wsnet.gforge.inria.fr/

6.1 Routing success rate

Figure 4 represents the percentage of success routing with respect to network node density. Results show that MobileCOP is outperformed by every solution in low densities. MobileR-*MultipleMoveR* and MobileR-*MultipleORouting* and their one-hop variant (respectively CoMNet *Mover* and CoMNet *ORouting On the Move*) are equivalent with a slight improvement for MobileR. It suggests that the path anticipation in MobileR might decrease coverage hole.
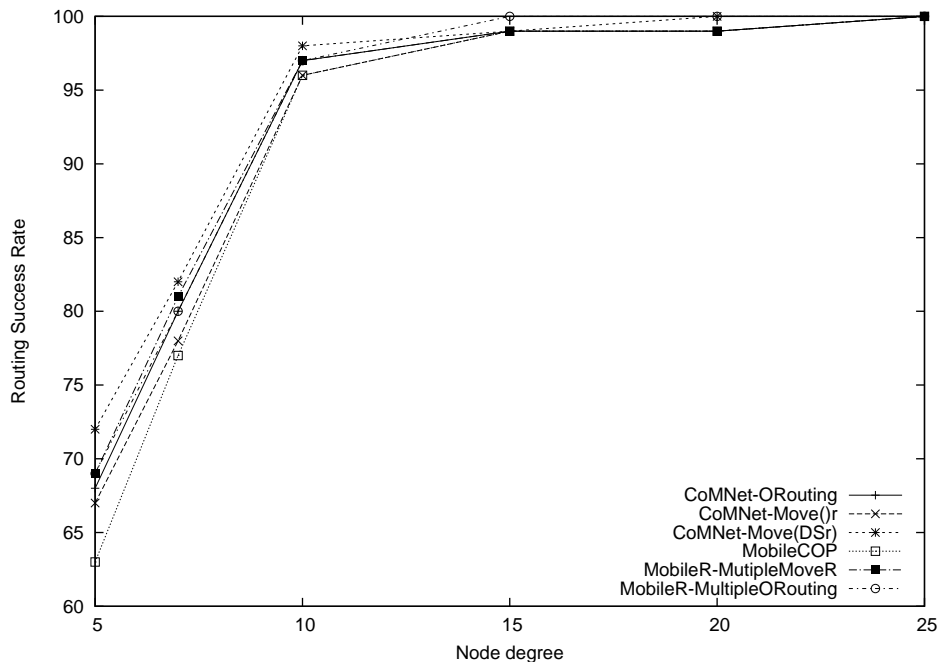

Fig. 4. Routing success rates of different routing algorithms

6.2 Energy consumption

We set the simulation parameters to the common values used in literature. On the one hand we used the values in $c = 3^8$ and $\alpha = 4$ in the energy model (Eq. 2). On the other hand, we use the same approach as the one developed in [8] regarding the moving cost model (Eq. 3). With E_T the radio transmission cost and E_D the displacement cost, three different cases can be envisaged:

– $E_T << E_D$: case in which the displacement cost over a distance d equals the transmission cost over the same distance. In that case we set the a parameters in (Eq. 2) such as $100 \times cost_s(d) = cost_m(d)$

- $E_T == E_D$: where $cost_s(d) = cost_m(d)$,
- $E_T >> E_D$: in which the radio transmission cost is much energy-hungry than the relocation cost over a same distance. a is here set such as $cost_s(d) = 100 \times cost_m(d)$

The simulator runs the different routing algorithms with the same source-destination couple for 10 consecutive times over 1000 maps.

With respect to the number of routings

Figure 5 displays the energy consumption as a function of the number of successive routings. Each mobility model is presented. Density is set to $\delta = 25$. As nodes may move on second or more iteration of routing with some of the algorithms, we decide to integrate both transmission and displacement costs in order to make possible true comparisons.

As expected, for all models, in any case, MobileR $MultipleMoveR$ and MobileR $Multiple\ ORouting$ outperform their CoMNet variant. This is due to their recursive computation of paths over their one hop neighbourhood. It permits MobileR variants to anticipate routing over few hops in best cases. And so they are able to perform better choices than the previously existing algorithms in most cases. Moreover, MobileR $MultipleMoveR$ appears to perform the best in most cases, as it aims to minimize both transmission and relocation costs.

Compared to MobileCOP, our approach is significantly better in terms of energy saving. This is due to the fact that MobileCOP does not take into account the cost of mobility. This is particularly obvious when displacement costs are expensive, see Figure 5(b). Moreover, as MobileCOP relies on a route request and then a route confirmation with relocation before message sending, its initial cost is higher than any other. On the one hand, route request and confirmation are the reason of the initial cost when transmission cost are the most expensive (Figure 5(c)). But on the other hand, the nodes relocation during route confirmation explain the initial cost when displacement cost are a lot more expensive than transmission ones(Figure 5(b)). MobileCOP and CoMNet heuristics tend to converge over the successive routing.

With respect to network density

Figure 6 displays the cumulative energy for ten consecutive routings along the same path by each algorithm as a function to network density. The presence of the CDS explains that MobileCOP outperforms our approach for low densities networks ($\delta \leq 10$) and when transmission costs are less or as costly than displacement costs. This is because most of the nodes are member of the CDS and thus can not move. With the increase of node density, more and more nodes are free to move, and thus they perform similarly to MobileCOP. Anyway, MobileCOP is outperformed when the moving costs are significantly higher: it does not take into account the cost of mobility.

Results show that MobileR $MultipleMoveR$ and MobileR $MultipleOrouting$ consume the same ($\delta < 10$) or less ($\delta > 10$) energy than their one hop CoMNet variant, respectively CoMNet $Move_r$ and CoMNet $ORouting$. In fact, the higher

the density is, the lower the energy consumed for routing is. Once a minimum density is reached ($\delta \equiv 10$), the required nodes for the CDS are selected, and so there are more nodes available for routing and Best Path computation.

7 Conclusion

MobileR routing protocol promises to lower the energy consumption of the routing process in wireless sensor and actuator networks by taking advantage of node mobility. Furthermore, MobileR is fully localised, scalable, energy efficient and guarantee network connectivity. It comes with two different relocation principles adapted to different traffic schemes. The performance of the these heuristics has been analysed based upon the results of exhaustive simulations. Future works include the consideration of multiple and concurrent simultaneous routing from different sources, and the guarantee of delivery. Future experiments will be conduced using realistic parameters before experimenting on the very large scale open wireless sensor network plateform Senslab.

Acknowledgements

References

1. Fin, G.G.: Routing and addressing problems in large metropolitan internetworks. (1987)
2. Rodoplu, V., Meng, T.: Minimizing energy mobile wireless networks. IEEE JSAC, 17(8):13331347, (1999)
3. Wang, W., Srinivasan, V., Chua, K.C.: Extending the lifetime of wireless sensor networks through mobile relays. IEEE/ACM Trans. Netw., 16(5):11081120, (2008)
4. Bose, P., Morin, P., Stojmenovi, I., Urrutia, J.: Routing with Guaranteed Delivery in ad hoc Wireless Networks. Wireless Networks, (2001)
5. Liu, H., Nayak, A., Stojmenovic, I.: Localised mobility control routing in robotic sensor wireless networks. LNCS, Mobile Ad-Hoc and Sensor Networks, 4864:1931, (2007)
6. Goldenberg, D.K., Lin, J., Morse, A.S.: Towards mobility as a routing primitive. Mobihoc, pages 163-174, (2004)
7. Carle, J., Simplot-Ryl, D.: Energy efficient area monitoring by sensor networks. IEEE Computer Magazine, 37:4046, (2004)
8. Hamouda, E., Mitton, N., Simplot-Ryl, D.: Ensuring Connectivity for Energy Efficient Mobile Routing in Actuator and Sensor Networks. 10th International Conference on Ad Hoc Networks and Wireless. AdHocNow. (2011)
9. Sanchez, J. A., Ruiz, P.M.: Exploiting local knowledge to enhance energy-efficient geographic routing. Mobile Ad-hoc and Sensor Networks, MSN, (2006)

10. Elhafsi, E.H., Mitton, N., Simplot-Ryl, D.: End-to-End energy efficient geographic path discovery with guaranteed delivery in ad hoc and sensor networks. Personal, Indoor and Mobile Radio Communications, (2008)
11. Stojmenovic, I., Lin, X.: Power-aware localized routing in wireless networks. IEEE TPDS, 12(11):11221133, (2001)
12. Kuruvila, J., Nayak, A., Stojmenovic. I.: Progress and location based localized power aware routing for ad hoc sensor wireless networks. IJDSN, 2:147159, (2006)
13. Shah, RC., Roy, S., Jain, S., Brunette,W.: Data MULEs: Modeling and Analysis of a Three-tier Architecture for Sparse Sensor Networks. Ad Hoc Networks, 1(2-3):215-233, (2003)
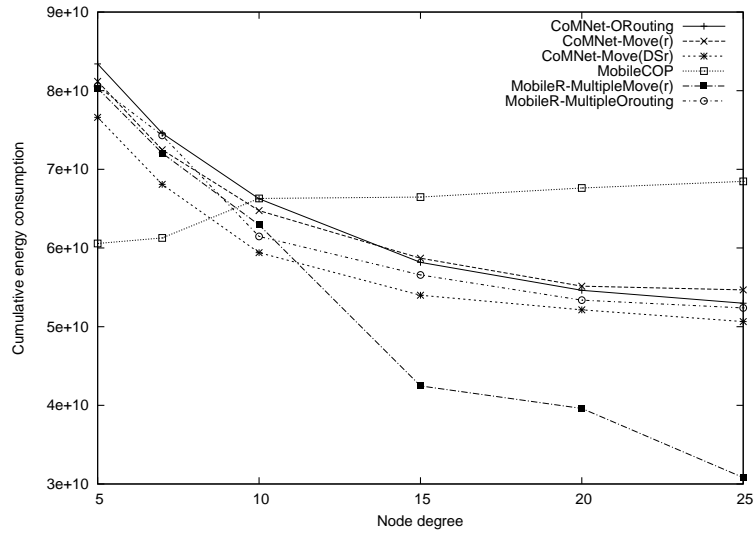14. htpp://www.senslab.info
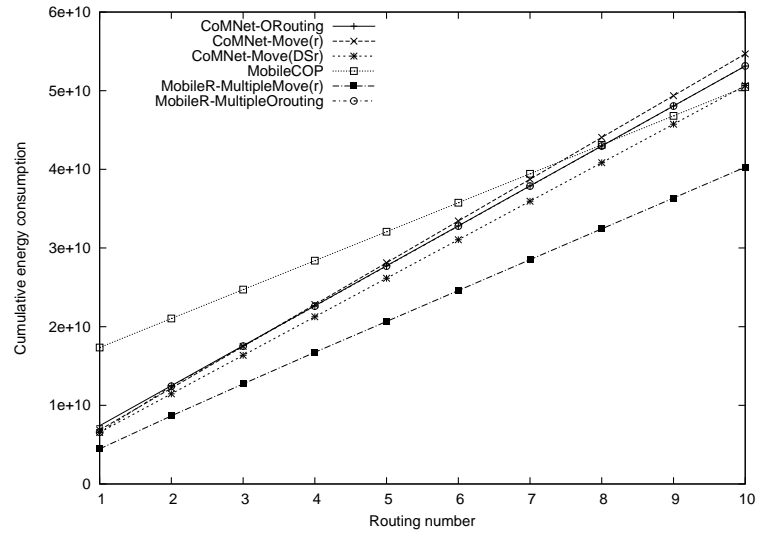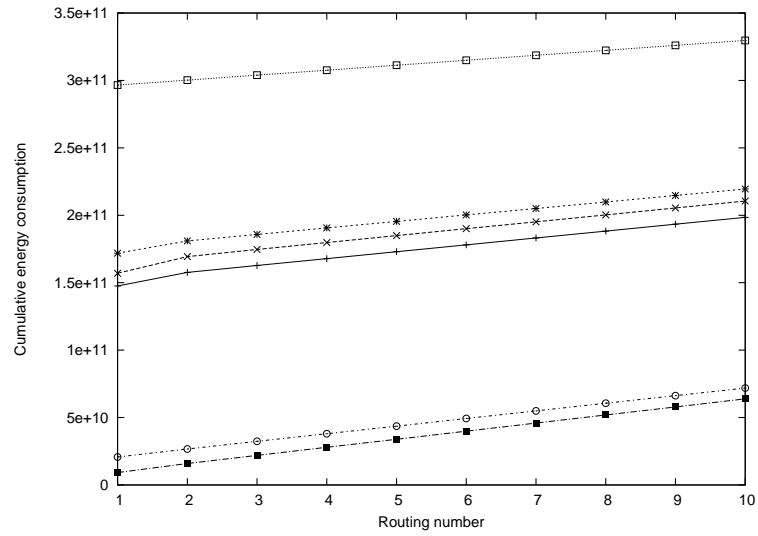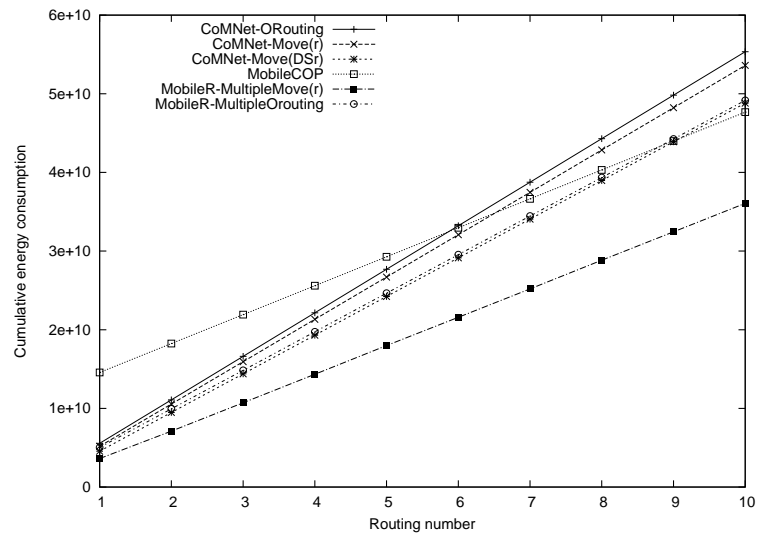
(a) $E_T == E_D$

(b) $E_T << E_D$
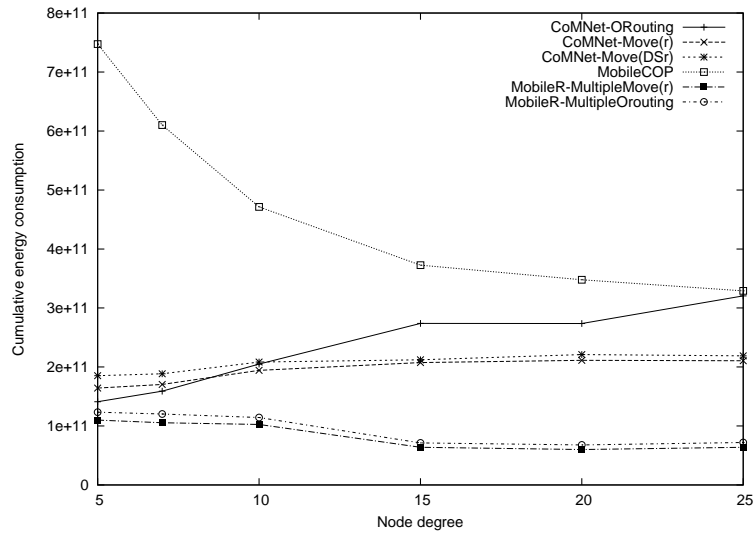

(c) $E_T >> E_D$

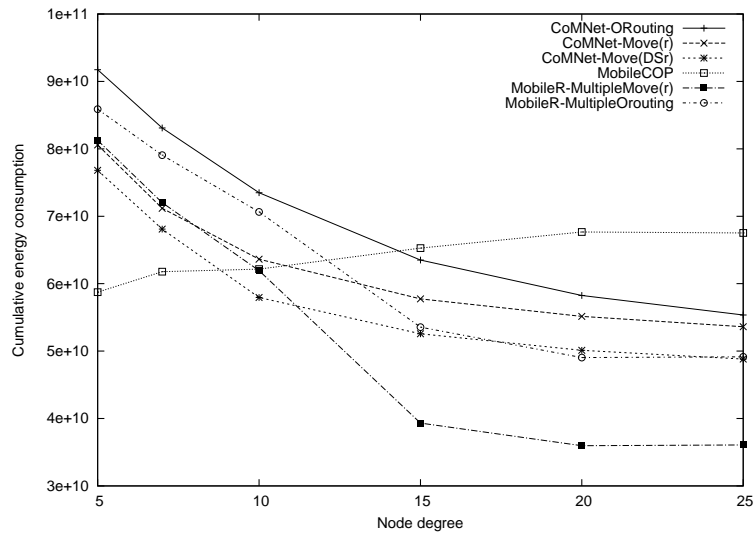**Fig. 5.** Cumulative Energy Consumption over 10 routes

(a) $E_T == E_D$

(b) $E_T << E_D$

(c) $E_T >> E_D$

Fig. 6. Energy consumption used for routing with regard to node degree