Greedy Routing Recovery Using Controlled Mobility in Wireless Sensor Networks

Nicolas Gouvy, Nathalie Mitton, Jun Zheng

Greedy Routing Recovery Using Controlled Mobility in Wireless Sensor Networks *

Nicolas Gouvy[1,2], Nathalie Mitton[2], and Jun Zheng[3]

[1] Université Lille Nord de France, LIFL, France,
[2] Inria Lille - Nord Europe, France,
[3] National Mobile Communications Research Lab, Southeast University, China
{nicolas.gouvy,nathalie.mitton}@inria.fr, junzheng@seu.edu.cn

Abstract. One of the most current routing families in wireless sensor networks is geographic routing. Using nodes location, they generally apply a greedy routing that makes a sensor forward data to route to one of its neighbors in the forwarding direction of the destination. If this greedy step fails, the routing protocol triggers a recovery mechanism. Such recovery mechanisms are mainly based on graph planarization and face traversal or on a tree construction. Nevertheless real-world network planarization is very difficult due to the dynamic nature of wireless links and trees are not so robust in such dynamic environments. Recovery steps generally provoke huge energy overhead with possibly long inefficient paths. In this paper, we propose to take advantage of the introduction of controlled mobility to reduce the triggering of a recovery process. We propose Greedy Routing Recovery (GRR) routing protocol. GRR enhances greedy routing energy efficiency as it adapts network topology to the network activity. Furthermore GRR uses controlled mobility to relocate nodes in order to restore greedy and reduce energy consuming recovery step triggering. Simulations demonstrate that GRR successfully bypasses topology holes in more than 72% of network topologies avoiding calling to expensive recovery steps and reducing energy consumption while preserving network connectivity.

Keywords: greedy routing, hole bypassing, wireless sensor networks, controlled mobility.

1 Introduction

Miniaturization, costs decrease and advances in low-power electronic and radio communication technologies have made possible the emergence of new kinds of networks such as Wireless Sensor Networks (WSN). WSN are sets of a handful to thousands of sensors communicating through the radio medium in a multi-hop fashion. Each sensor embeds a low-power processor with limited computing and memory capabilities, a radio device and sometimes a localization device. Sensors

* This work was partially supported by a grant from CPER Nord-Pas-de-Calais FEDER Campus Intelligence Ambiante and Université Lille 1.

in WSN forward their data regularly, on event or on request. It is essential for the nodes to collaborate in order to route data in a reliable and energy-efficient way to a given destination. A popular routing family for WSN is geographic routing which requires nodes to be aware of their location. Geographic protocols generally include a greedy mechanism making each forwarding node to forward the packet to one of its neighbors closer to the destination than itself. Routing fails if their is no neighbor closer.Multiple mechanisms have been developed to address the greedy routing failure such as network planarization [1], trees [2], or hole detection mechanisms [3]. However they do not fit with WSN mobility at all as they strongly rely on nodes static position to overcome failures.

A new approach is to introduce controlled mobility enabled sensors. [4] shows that deploying resourceful mobile devices in a WSN provides better results than increasing network density. Still, only a few works use this controlled mobility in order to optimize route topology. Moreover none of them integrates a recovery mechanism as the classical approaches which have been developed in static WSN are unsuited to mobility at all. As a response, we propose the Greedy Recovery Routing routing protocol. GRR adapts the network topology in order to make energy savings with regard to the routed traffic. It enhances existing greedy routing protocol CoMNet and extends it with a light recovery mechanism. Both (enhanced CoMNet and light recovery steps) take advantage of node controlled mobility. GRR aims to bypass geographic routing failure by relocating nodes such as the greedy routing can be reused. GRR shows the following properties:
- *Localized*: Routing decisions rely only on local information: forwarding node geographical location, the ones of its neighbors and of the destination.
- *Scalable*: GRR is memoryless, no routing path information has to be stored.
- *Energy efficient*: At each routing steps GRR chooses next hop and computes its relocation taking all costs into account.
- *Guaranteed connectivity*: GRR guarantees network connectivity through the use of a Connected Dominated Set or the Relative neighborhood Graph.
- *Less calls to hard recovery*: GRR implements a light recovery mechanism which aims to reduce the triggering of expensive hard recovery steps (based on face or tree traversals) by restoring greedy routing when possible. It will eventually fill routing holes restoring an end-to-end greedy routing.

The remaining of this paper is organized as follows. Section 2 reviews existing works on delivery guarantee and mobile routing in WSN. We detail the models used in the paper in Section 3, while the prerequisites are exposed in Section 4. Section 5 presents and details our approach. Simulation results are detailed in Section 6. Finally Section 7 concludes the paper and presents future work.

2 Related Work

Delivery guarantee in static networks

To the best of our knowledge, the most popular recovery strategy for geographic routing is face routing [1]. It makes a packet traverse the faces of the planarized network graph until greedy routing is possible. Faces traversal is done using the right-hand-rule. However, face is energy-consuming since it may generate long detours and makes the packet follow a succession of short edges [5]. Moreover, it requires reliable network planarization which is nearly impossible to provide in real world due to the non-uniform wireless links.

In hull trees [2] each node has an associated convex hull that "contains the location of all its descendant in the tree". When greedy routing fails, forwarding node checks its hull tree downstream to find a path to destination. If the destination belongs to the hull tree of the forwarding node, packet is forwarded to the first corresponding child node. Otherwise packet is forwarded upstream. This approach can be very memory consuming depending on network density and size and number of hull trees employed. Moreover a moving node could easily destroy trees.

Authors in [3] propose an approach which does not rely on planarization nor trees. At network bootstrap, an algorithm is applied locally on each node in order to mark those where packets may possibly get stuck. And then, each marked node identifies both its upstream and downstream nodes on the boundary of the hole. Consequently, when the greedy routing fails, the packet can be routed along the hole until greedy becomes possible again. Yet this approach requires a lot of messages at network bootstrap and in order to adapt to topology changes.

To the best of our knowledge, none of the existing approaches behaves well under the hypothesis of mobility unlike Greedy Routing Recovery which successfully takes advantage of mobility to both optimize greedy routing -enhancing a previously proposed approach- and reduce the number of calls to expensive recovery steps.

Routing with controlled mobility

Mobility has long been considered as a hazard in WSN, causing a degradation of performances or routing failures. A handful of proposals considers the use of controlled mobility in order to adapt the network topology with regard to the routing path. Existing routing protocols such as MobileCOP [6] find an initial route using a Cost-Over-Progress (COP) metric [7], and then iteratively move each forwarding node to the midpoint of its upstream and downstream nodes on the route. These routing protocols may not be efficient as they can cause energy consuming zig-zag movements and the network may be disconnected (a node may move out of range of its neighbors). In addition, none of these approaches considers the cost of moving in the routing decision. CoMNet [8] is the first fully localized COP-based geographic routing protocol that takes the moving cost into account while guaranteeing network connectivity. It has been extended in [9] in order to consider mobility consequences on a multiple hop point of view. Nevertheless, none of the controlled mobility enabled protocols can recover from a local minimum.

GRR bases on CoMNet: it enhances it in its greedy part and extends it with a light recovery mechanism to exploit controlled mobility in order to restore greedy routing when possible.

3 Models

The network is modeled as an undirected simple finite graph $G(V, E)$, with V the set of nodes and E the set of edges. $(A, B) \in E$ if A and B are in transmission range. The Euclidean distance between A and B is noted as $|AB|$. We denote by $N(A)$ the set of neighbors of A: $N(A) = \{V \in E \mid (A, V) \in V\}$ and $N_D(A)$ the subset of $N(A)$ which are closer to the destination D than A: $N_D(A) = \{B \in N(A) \wedge |BD| < |AD|\}$. Every node in V is aware of its geographical location and can be either a mobile or stationary sensor. Relocation of a node A is noted as A'. Although our approach is model-independent, we use the following widely employed cost models as a proof of concept:

Transmission cost We denote by $C_s(.)$ the energy consumption or cost for radio transmission between two nodesdistant of r [10]:

$$C_s(r) = r^\alpha + c \quad \text{if } r \neq 0 \tag{1}$$

where c represents the energy overhead due to radio device, α is a real constant (> 1) that represents the signal attenuation. The associated optimal radio transmission radius [11] for radio transmission is $\mathrm{r}^* = \sqrt[\alpha]{\frac{\mathrm{c}}{\alpha-1}}$.

Mobility cost We denote by $C_m(.)$ the cost to relocate a node B to B'. We use the model adopted in previous similar works [6], in which k is a constant :

$$C_m(|BB'|) = k * |BB'| \tag{2}$$

4 Preliminaries

4.1 Greedy Routing

The greedy step of GRR proposes an enhanced CoMNet (COnnectivity preservation Mobile routing protocols for actuator and sensor NETworks) [8]. CoMNet is a geographic routing protocol for WSN which takes advantage of nodes controlled mobility in order to adapt network topology to the routing traffic. Precisely, CoMNet uses a Cost-Over-Progress (COP) [7] approach: current node A chooses $B \in N_D(A)$ which minimizes the ratio of the global cost (packet transmission cost plus node relocation cost) to the progress made towards the destination. Indeed, B satisfies the following optimization problem:

$$B = arg_{min_{K \in N_D(A)}} \frac{\mathrm{C_s}(|AK|) + \mathrm{C_m}(|KK'|)}{|AD| - |K'D|} \tag{3}$$

CoMNet comes in three different variants to fit the best to various environments:

- $CoMNet - Move_{(DSr)}$ aligns nodes on the Source Destination (SD) line with all hop lengths to be equal to the optimal transmission distance r^*.
- $CoMNet - ORouting\ on\ the\ Move$ aligns nodes on the (SD) line.
- $CoMNet - Move_r$ makes next hop node B is relocated on the intersection of the circle $C(A, r^*)$ of radius r^* centered at A and the (BD) line.

4.2 Connectivity

Relative Neighborhood Graph (RNG): RNG [12] is a graph reudction that can be computed locally. An edge (U, V) exists if distance $|UV|$ is less than or equal to the distances $|UW|$ and $|VW|$ for any other vertex W:

$$\forall W \neq U, V \in E : |UV| \geq max[|UW|, |VW|] \tag{4}$$

It reduces the average node degree to $\simeq$3 while preserving networking connectivity. A moving node which stays connected to its RNG neighbors will keep network connectivity unchanged.

Connected Dominated Set (CDS): CDS is a connected subset of V that covers the same area. If nodes in the CDS (*i.e. dominant*) are static, we ensure that all other nodes which move stay in transmission range of the CDS. It guarantees that their is always a path between every pair of nodes of the network. In [13] authors have proposed a fully localized algorithm. Giving a node A, $N(A)$ and S the subset of $N(A)$ with higher priority (such as id, battery level, etc...) than A: $S \leftarrow N(A) - \{U \in E \mid p(U) < p(A)\}$, A is dominant if one of the following statement do not hold:
- S is not empty: $S \neq \{\emptyset\}$
- S is connected: $\forall A \in S, \exists B \in S\ s.a.\ A \neq B \wedge |AB\ < radio\_range$
- every node in $N(A)$ is in S or in range of S: $\forall B \in N(A), B \in S \vee N(B) \cap S \neq \emptyset$

5 Geographic Routing Recovery

5.1 Overview

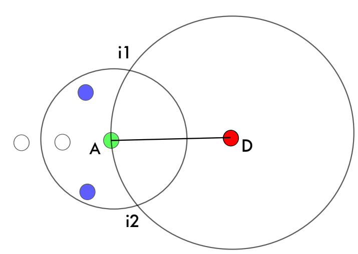


Fig. 1: $N_D(A)$ is empty, greedy routing to D fails on A. Previous routing nodes are in white.

Greedy routing failure is due to the lack of a neighbor closer to the destination than the current forwarding node. Figure 1 represents such a case: forwarding node A has no neighbor closer to destination D. A represents a local extremum of the routing path toward destination node D, it faces a hole. Greedy forwarding requires at least one node in the area defined by the intersection of $C_1(A, r*)$ the disk centered at A and of radius $r*$, and $C_2(D, |AD|)$ the disk centered at D of radius $|AD|$ to succeed.

Hence we propose Greedy Routing Recovery which combines greedy routing and a light recovery mechanism which aims to restore the greedy forwarding.

In greedy forwarding, GRR routes packet and relocates nodes in a CoMNet way. Forwarding nodes are aligned on the source-destination line in order to save energy. However, when greedy fails, GRR switches to light recovery step. In light recovery, forwarding nodes relocates next hop on the intersection locations i_1 (or i_2) in order to bypass the routing hole. When greedy routing become possible again, GRR switches back to it. GRR uses nodes controlled mobility to both optimize network topology to the traffic and create greedy routing paths in order to avoid expensive recovery steps.

Algorithm 1 GRR(A,D, p) - Node A has a packet p for node D

1: **if** isInGreedyMode(p) **then**
2:     **if** $N_D(A) \neq \emptyset$ **then**
3:         $next, next' \leftarrow SelectGreedy(A, D)$;
4:     **else**
5:         $addFailureLocation(p)$;
6:         $next, next' \leftarrow SelectRecovery(A, D)$;
7:     **end if**
8: **else**
9:     $moveToLocation(p)$ {execute relocation order while it does not disconnect the network}
10:     **if** $N_D(A) \neq \emptyset$ **and** $|AD| \leq |D, failureLocation(p)|$ **then**
11:         $next, next' \leftarrow SelectGreedy(A, D)$;
12:     **else**
13:         $next, next' \leftarrow SelectRecovery(A, D)$;
14:     **end if**
15: **end if**
16: forward(p,next,next') {forward packet p to node $next$ with relocation order in $next'$}

More precisely GRR works as follows. Upon reception of a packet in greedy forwarding (Algo 1, l1), a node A checks its neighborhood toward destination $N_D(A)$. If it exists a node *next* closer to the destination than itself, A computes its new location $next'$ and forwards the packet to it before or after relocating it (Algo 1, l3) depending on CoMNet variants. If there is no neighbor closer to the destination, A marks the packet into light recovery mode adding greedy failure location into its header (Algo 1, l5). A then computes the i_1 and i_2 intersections locations. It forwards the packet to the node *next* in its neighborhood $N_{(A)}$ whose transmission and relocation cost to the i_x is minimized (Algo 1, l6). The packet includes a relocation order to position i_x.
Upon reception of a packet in light recovery, a node A moves to the joined location. When stopped, A checks wether it can turn the packet into greedy forwarding. This is possible only if $N_A(D) \neq \emptyset$ and if $|AD| < |failure_locationD|)$ (Algo 1, l9-10). Otherwise the packet is forwarded to the node *next* in its neighborhood $N_{(A')}$ whose transmission and relocation costs to the i_x is minimized. We have to mention that network connectivity is guaranteed despite nodes movement using either the CDS or the RNG neighbors as described in Section 4.2. Every moving node moves up to its new location while its relocation does preserve network connectivity.

Figure 2 illustrates a complete GRR routing from S to D. Routing is greedy from S to B and nodes A, B and C are relocated on the (SD) line. However,

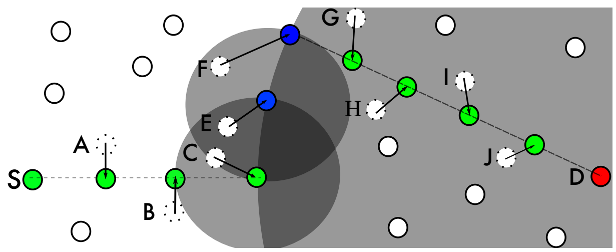


Fig. 2: GRR routing from node S to D. Original node location is dashed. Green nodes have been relocated during greedy. Blue ones during light recovery.

greedy fails on C as none of its neighbors is closer to D than itself. Consequently, C switches the packet into light recovery. Node E is selected: C forwards the packet to it with the order to relocate in i_x. E relocates. When E stops, it checks if greedy is still impossible. That is still the case: using greedy recovery E forwards the packet to F and makes it relocate. On F, greedy routing is possible since G exists, furthermore F is not further to D than C. On the next routing, the SD path will be completely greedy.
In the following Section, we detail the sub-algorithms used to select and relocate next forwarder while in greedy forwarding or in light recovery.

5.2 Greedy

Although GRR relies on CoMNet [8] relocation patterns in its greedy part, routing is different as GRR computes the COP associated to each pattern for every possible next hop at each step of the routing as described in Algorithm 2. Forwarding node A computes for each neighbor B in $N_D(A)$ (Algo 2, l4) its relocations according to the three different CoMnet variants (Algo 2, l6-8) and retains the associated lowest COP (Algo 2, l9). Finally A returns the node (and its relocation) in $N_D(A)$ which minimizes the COP (Algo 2, l10-13, l16).

Algorithm 2 SelectGreedy(A,D) - Run at node A toward destination D.

1: $next \leftarrow -1$; {next hop elected}
2: $next\prime \leftarrow \{0,0,0\}$; {next hop computed relocation}
3: $minCOP \leftarrow +\infty$ {lowest COP over all $N_D(A)$}
4: **for all** $\{B \in N_D(A)\}$ **do**
5: $\quad B\prime_{or}, B\prime_{mr}, B\prime_{mdsr} \leftarrow \{0,0,0\}$;
6: $\quad B\prime_{or} \leftarrow \mathrm{ORouting}(A,B,D)$; $B\prime_{mr} \leftarrow \mathrm{Move_R}(A,B,D)$; $B\prime'_{mdsr} \leftarrow \mathrm{Move_{DSR}}(A,B,D)$
7: $\quad bCOP \leftarrow min[COP(A,B,B\prime_{or}), COP(A,B,B\prime_{mr}), COP(A,B,B\prime_{mdsr}]$;
8: $\quad$ **if** $(bCOP < minCOP)$ **then**
9: $\quad\quad next \leftarrow v, next\prime \leftarrow$ {relocation which minimizes COP}
10: $\quad\quad minCOP \leftarrow bCOP$
11: $\quad$ **end if**
12: **end for**
13: **return** $next, next\prime$ {return next node $next$ with its computed relocation in $next'$}

5.3 Light recovery

During light recovery, GRR stops considering CoMNet relocations patterns as all of them would fail. However GRR makes the forwarding A node compute for every node B in its $N(A)$ –and not only in $N_D(A)$– the relocation cost on the intersection between $C_1(A, |r^*|)$ $C_2(D, |AD|)$ noted as i_1 or i_2 plus the transmission cost from A to B. Those specific i_x locations make possible the bypass of the routing hole as they restore greedy routing on the next routing. Moreover, those locations minimize the moving distance for next hop B and the $|AB'|$ radio transmission cost is optimal. Elected next hop B satisfies this optimization problem:

$$B = arg_{min_{K \in N(A)}} \mathrm{C_s}(|AK|) + \mathrm{C_m}(|Ki_x|) \tag{5}$$

where i_x is replaced by respectively i_1 or i_2.

Algorithm 3 details light recovery. Forwarding node A first computes the intersection locations i_1 or i_2 (Algo 3, l4-5). Then, A computes for each of its neighbor B in $N_D(A)$ (Algo 3, l6) the total cost of its relocation both in i_1 and i_2 and the associated transmission cost from B to A (Algo 3, l7-8). A finally forwards packet and relocation order to the node which minimizes both costs the most (Algo 3, l9-18, l20).

Algorithm 3 SelectRecovery(A,D) - Run at node A when $N_D(A)$ is empty.

1: $next \leftarrow -1$ {next hop elected}
2: $next\prime \leftarrow$ {0,0,0} {next hop computed relocation}
3: $minCOST \leftarrow +\infty$ {lowest total cost computed over all $N_D(A)$}
4: $(i_1, i_2) \leftarrow intersections(C_1(A, |r^*|), C_2(D, |AD|))$;
5: **for all** $\{B \in N_D(a)\}$ **do**
6: $tCOST1 \leftarrow \mathrm{C_s}(|AK|) + \mathrm{C_m}(|Ki_1|)$; $tCOST2 \leftarrow \mathrm{C_s}(|AK|) + \mathrm{C_m}(|Ki_2|)$
7: **if** $(tCOST1 < minCOST)$ **then**
8: $next \leftarrow B$; $next\prime \leftarrow i_1$; $minCOST \leftarrow tCOST1$
9: **end if**
10: **if** $(tCOST2 < minCOST)$ **then**
11: $next \leftarrow B$; $next\prime \leftarrow i_2$; $minCOST \leftarrow tCOST2$
12: **end if**
13: **end for**
14: **return** $next, next'$ {return next node $next$ with its computed relocation in $next'$}

6 Experiments and performance analysis

6.1 Simulation Environment

We simulate GRR using the last release of WSNet [14] network simulator. Different node degrees δ (from 10 to 25) are considered and maximum node speed is set to 1m.s^{-1}. Relocations engender delays in packet delivery which are not in the scope of this study. Maximal radio range is set to 50m. Simulations are performed on 25 randomly generated 500x500m maps with nodes uniformly deployed on which 3 holes of diameters 100m has been created. Greedy routing

fails on those maps for the selected source and destination couple. Initial battery level of every node is 1J and every node is capable of moving. $Cs(.)$ is computed using Eq. 1 in which we use common values: *i.e.* c = 3.10^9J and α=2, which leads to an optimal transmission range of $r^* = 22.36$m. $Cm(.)$ is computed using Eq. 2, with mobility parameter k computed as follows:
1. if $C_s(.) = C_m(.)$, k is solution to $C_s(r^*) = C_m(r^*)$.
2. if $C_s(.) >> C_m(.)$, then k is solution to $C_s(r^*) = 10^2 C_m(r^*)$.
3. if $C_s(.) << C_m(.)$ then k is solution to $C_s(r^*) = 10^{-2} C_m(r^*)$.

6.2 Routing success rate

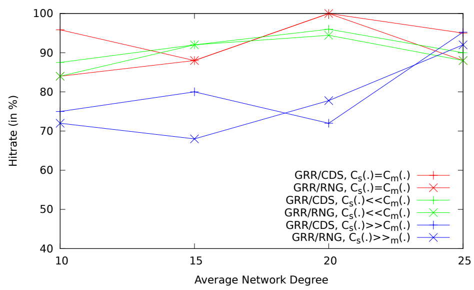


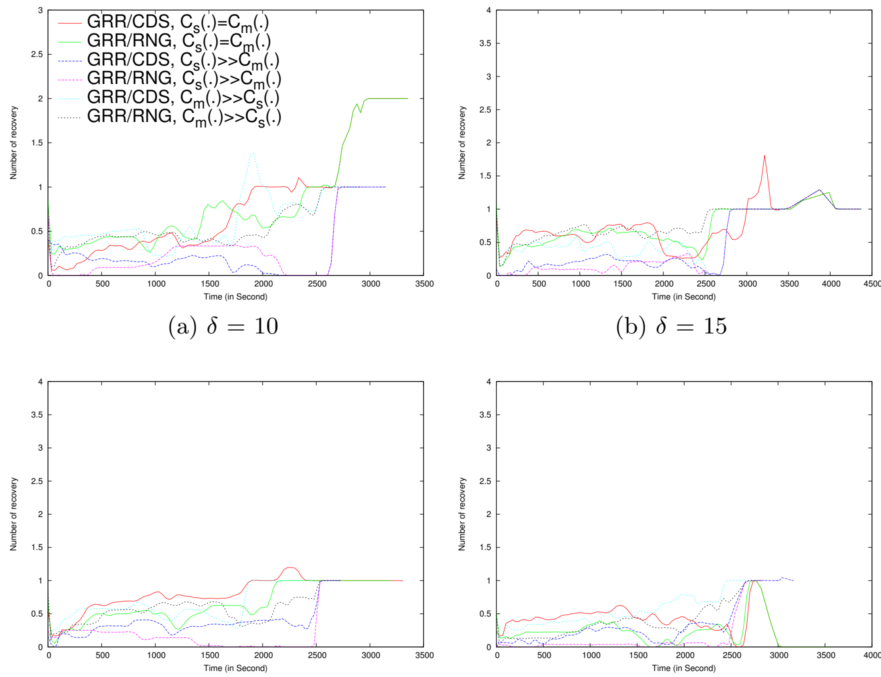
Fig. 3: Percentage of topologies where GRR succeed with regard to δ

Figure 3 shows the percentage of topologies on which routing succeeds using GRR with regards to various network densities and cost models. Note that each of the topologies used in those simulations is connected at network bootstrap. In other words, it exists at least one multi-hop path between every pair of nodes in the network at network start. Even so, a classical geographic greedy heuristic would fail because of network holes.

With regards to the connectivity guarantee mechanism and density, results are very similar. When the network density is low, most nodes belong to the CDS and so mobility is limited in GRR-CDS. RNG neighbors in GRR-RNG are also very sparse and so mobility is limited. With the increasing density, the percentage of topologies on which GRR (all variants) succeeds increases as more and more nodes are free to move with GRR-CDS. Similarly in GRR-RNG, neighbors are more numerous - and consequently closer - and so mobility freedom increases.
With regards to cost model, we see that the percentage of routing success is maximal when $C_s(.) == C_m(.)$. Considering that A is the forwarding node in recovery mode, the reason is that it makes A consider every node in $N(A)$ equally. However, success rate is minimal when $C_s(.) >> C_m(.)$ as it provokes the selection of a node which is close to A in order to reduce transmission costs. But this node has to travel a long distance to reach i_x and restore greedy. Yet this distance can be impossible to travel due to the connectivity guarantee mechanism. Middle case is when $C_s(.) << C_m(.)$: A selects as next forwarder a node close to i_x.

6.3 Number and length of recovery occurrences

Figure 4 depicts the number of transitions between greedy and recovery forwarding steps with regards to time. Results show that at network bootstrap the



(a) $\delta = 10$ (b) $\delta = 15$

(c) $\delta = 20$ (d) $\delta = 25$

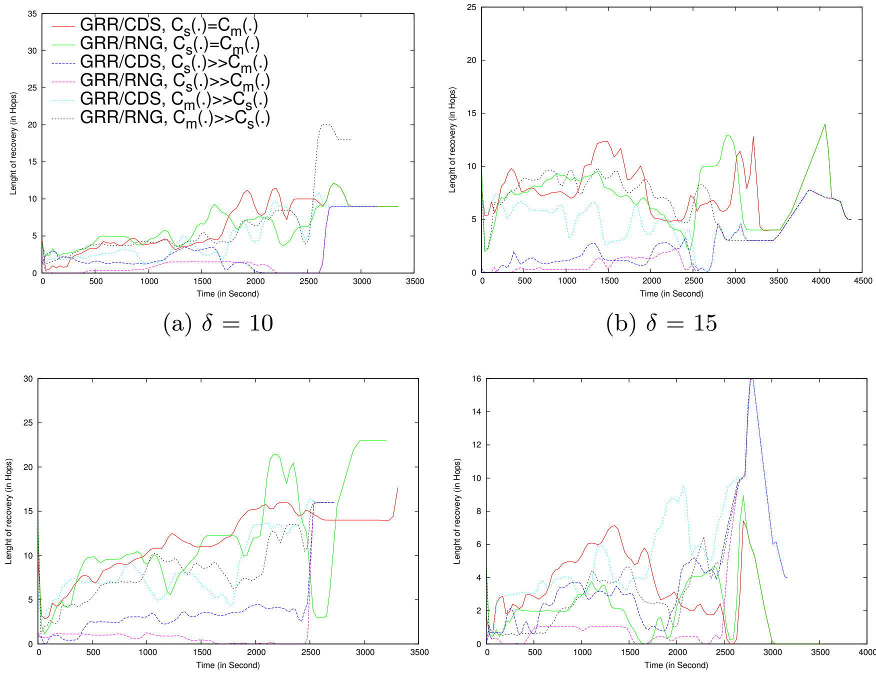
Fig. 4: Average number of recovery occurrences along time

number of recovery occurrences tends to decrease very quickly due to an initial greedy path restoration thanks to our recovery mechanism. Then, nodes that have been relocated die since they have spent energy to relocate and are now highly employed for greedy forwarding. Consequently the number of recoveries increases again after a certain time since the light recovery routing scheme has to restore greedy forwarding. However this might not be possible since movable nodes might have already moved: that is the reason why sometimes light recovery occurrences does not decrease any more.

This analysis is confirmed by Figure 5 which represents the average path length (in number of hops) in recovery step. We see that after an initial decrease, the number of hops in recovery stabilizes and then increases again after a certain time when relocated nodes start to die. The routing hole is bypassed but the GRR routing protocol has to circumvent an increasing hole along time. Please note that routing does not succeed at all without GRR light recovery.

6.4 Average packet cost

Figure 6 represents the total cost – including both transmission and relocation costs of nodes on routing path – of a delivered packet with regards to time.



(a) $\delta = 10$

(b) $\delta = 15$

(c) $\delta = 20$

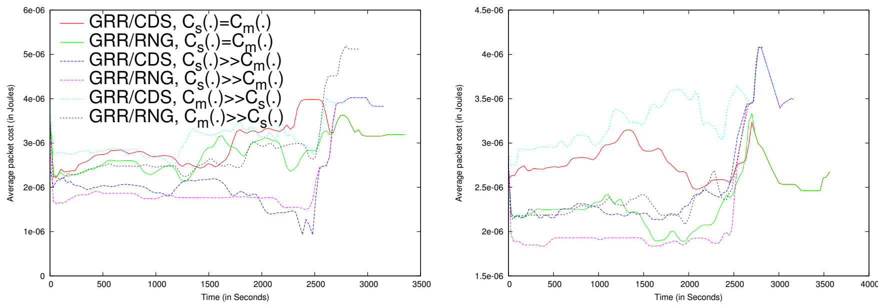
(d) $\delta = 25$

Fig. 5: Average length of recovery step in hops along time

Similarly to previous behavioral results, results show an initial energy overhead whatever the cost model, the recovery mechanism or the network density. This is due to the initial path relocation which makes node move a lot. In a second step, packet cost tends to stabilize up to a final stage where packet cost increases dramatically because GRR has to bypass major holes.

7 Conclusion

In this paper we introduce a novel protocol, GRR which takes advantages of node controlled mobility to adapt network topology to the traffic and addresses the problem of routing hole bypassing in wireless sensor networks. Our proposal relies on a COP approach and takes into account the cost of node relocation. Simulation shows that GRR successfully restores greedy routing in more than 72% of topologies. Our future work will combine GRR with a mobility enhanced face routing. Another topic of interest will be the use of controlled mobility in order to adapt network topology to the context of multiple sources and multiple destinations.



(a) $\delta = 10$ (b) $\delta = 25$

Fig. 6: Average packet total cost along time

References

1. Bose, P., Morin, P., Stojmenović, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless Networks **7**(6) (2001) 609–616
2. Leong, B., Liskov, B., Morris, R.: Geographic routing without planarization. Proc. 3rd Conf. on Networked Systems Design & Implementation. NSDI **3** (2006) 25–25
3. Fang, Q., Gao, J., Guibas, L.J.: Locating and bypassing routing holes in sensor networks. Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies. INFOCOM **4** (2004) 2458–2468
4. Wang, W., Srinivasan, V., Chua, K.C.: Extending the lifetime of wireless sensor networks through mobile relays. Trans. on Networking **16**(5) (2008) 1108–1120
5. 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. In: Proc. 19th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications. PIMRC. (2008) 1–5
6. Liu, H., Nayak, A., Stojmenović, I.: Localized mobility control routing in robotic sensor wireless networks. In: Proc. 3rd Int. Conf. on Mobile ad-hoc and sensor networks. MSN. (2007) 19–31
7. Kuruvila, J., Nayak, A., Stojmenović, I.: Progress and location based localized power aware routing for ad hoc and sensor wireless networks. Int. Journal of Distributed Sensor Networks **2**(2) (2006) 147–159
8. Hamouda, E., Mitton, N., Simplot-Ryl, D.: Energy Efficient Mobile Routing in Actuator and Sensor Networks with Connectivity preservation. In: Proc. 10th Int. Conf. on Ad Hoc Networks and Wireless. AdHocNow. (2011) 15–28
9. Gouvy, N., Mitton, N.: Mobiler : Multi-hop energy efficient localised mobile georouting in wireless sensor and actuator networks. (2011)
10. Rodoplu, V., Meng, T.: Minimum energy mobile wireless networks. IEEE Journal on Selected Areas in Communications **17** (1999) 1333–1344
11. Stojmenović, I., Lin, X.: Power-aware localized routing in wireless networks. IEEE Trans. on Parallel and Distributed Systems **12**(11) (2001) 1122–1133
12. Toussaint, G.: The relative neighbourhood graph of a finite planar set. Pattern Recognition **12**(4) (1980) 261–268
13. Carle, J., Simplot-Ryl, D.: Energy-efficient area monitoring for sensor networks. Computer **37**(2) (2004) 40–46
14. Fraboulet, A., Chelius, G., Fleury, E.: Worldsens: development and prototyping tools for application specific wireless sensors networks. In: Proc. 6th ACM/IEEE Int. Conf. on Information Processing in Sensor Networks. IPSN. (2007) 176–185