

Guaranteed delivery for geographical anycasting in wireless multi-sink sensor and sensor-actor networks

Nathalie Mitton¹, David Simplot-Ryl¹ and Ivan Stojmenovic²

¹INRIA Lille-Nord Europe, LIFL, France, {nathalie.mitton,david.simplot-ryl}@inria.fr

²SITE, University of Ottawa, Canada, stojmenovic@storm.ca

This work was supported by a grant from CPER Nord-Pas-de-Calais FEDER CIA and NSERC Canada Strategic grant STPGP 336406 - 06.

Abstract—In the anycasting problem, a sensor wants to report event information to one of sinks or actors. We describe the first localized anycasting algorithms that guarantee delivery for connected multi-sink sensor-actor networks. Let $S(x)$ be the closest actor/sink to sensor x , and $|xS(x)|$ be distance between them. In greedy phase, a node s forwards the packet to its neighbor v that minimizes the ratio of cost $cost(|sv|)$ of sending packet to v (here we specifically apply hop-count and power consumption metrics) over the reduction in distance $(|sS(s)| - |vS(v)|)$ to the closest actor/sink. A variant is to forward to the first neighbor on the shortest weighted path toward v . If none of neighbors reduces that distance then recovery mode is invoked. It is done by face traversal toward the nearest connected actor/sink, where edges are replaced by paths optimizing given cost. A hop count based and two variants of localized power aware anycasting algorithms are described. We prove guaranteed delivery property analytically and experimentally.

I. INTRODUCTION

Wireless sensor networks are receiving a lot of attentions in recent years due to their various potential applications. Sensor measurements from the environment may be sent to a base station (sink) in order to be analyzed. Other sensors may serve as routers on a path established to deliver the report. In larger sensor networks, there exists a bottleneck (around sink) if a single sink collects reports from all sensors. Scenarios with multiple sinks are then being considered, where each sensor reports to one of sinks. In wireless multi-sink sensor-actor networks, anycasting is performed when any of sinks may receive the report from sensors, and meet application demands. However, the cost of anycasting and providing service may depend on the distance of the receiving sinks/actors to the reporting sensor. It is therefore desirable that selected anycasting algorithm reaches one of actors which is close to the event.

Energy consumption and scalability are two challenging issues since wireless networks operate on limited capacity batteries with a potentially large number of nodes. To the best of our knowledge, only few protocols have been designed for anycasting in wireless networks. Most of them are based on an adaptation of anycasting for wired networks [1] and need non-scalable flooding techniques. Other ones [2], [3] need costly tree structure that is not robust. The only known position based anycasting algorithm [4] is greedy and localized but optimizes neither hop count nor power consumption.

Centralized algorithms are not scalable due to communication overhead needed to gather information at a single node. Localized algorithms are thus desirable. In case of localized position based anycasting problem considered in this article, sensor nodes are merely aware of their positions, positions of

their neighbors, and positions of all sinks. We present three geographic localized anycast routing protocols that guaranteed delivery (therefore loop-less), are memory-less, and scalable: GFGA, COPA and EEGDA. They are generalizations of the well-known greedy-face-greedy (GFG) [5] unicast routing protocol. GFGA anycasting algorithm is used in applications where path needs to be as short as possible in terms of number of hops while still guaranteeing delivery. COPA and EEGDA are two variants for applications in which energy consumption has to be reduced.

II. RELATED WORKS

Anycasting for wireless networks has first been modeled in [1]. Although many anycast protocols have been deployed in wired networks [6], there are very few for anycasting in wireless networks in the literature and only one of them [4] is geographical. Most of existing solutions are based on anycast for wired networks and need to build some structures. For example, in [2] a shortest path anycast tree rooted at each source is constructed for each event source. Tree-based protocols are not scalable, since the maintenance is costly when network has dynamic changes or when actors are moving.

A number of articles refer to the anycast while solving a different problem. For example, in [7], sinks flood the network with queries, creating spanning trees for reporting. The protocol is similar to directed diffusion, where event sources are passive until demand for data is received. Another type is anycasting to a region, until one node from the region is reached [8], [3]. This differs from our problem formulation where actors are not in the same region, and in fact they are preferably spread to cover all the area. The only known position based anycasting algorithm is proposed in [4], where energy consumption needed to communicate at distance d is to $u(d) = d^\alpha + c$. In the startup phase, each sensor node selects its next hop as follows. Let Q be a sensor, N be one of its neighbors, and A be one of actors. Sensor Q selects neighbor N for which $u(|QN|) + u(|NA|)$ is minimized, over all neighbors and over all actors. This localized algorithm does not optimize the power consumption (despite the claim), because it makes decision in the neighbor selection process based on long edges $|NA|$ which are not power optimal (an analytical proof of this fact was given in [9]). Further, it does not guarantee delivery. Authors [4] formulate integer linear program to construct data aggregation tree to minimize overall energy spent for reporting.

We now review existing unicast routing protocols whose basic concepts are applied in our localized geographic any-

casting schemes. In well known *greedy* routing [10], node S currently holding the packet for destination D forwards it to the neighbor that is closest to D . This simple localized algorithm does not guarantee delivery. Power aware greedy routing algorithms were first studied in [11]. Cost over Progress (COP) based routing [12] is a localized metric aware greedy routing scheme. A node u forwards a packet to neighbor v such that the ratio of the cost to reach v to the progress made (measured as the reduction $|uD| - |vD|$ in distance to D) is minimized. Only neighbors closer to D than u are eligible ($|uD| > |vD|$). Cost could be an arbitrary additive metric, such as hop count, etc. This algorithm does not guarantee delivery.

In GFG [5], greedy routing is hop count based and is applied till reaching either the destination or a local minimum. Face routing is applied to recover from greedy failure. It requires the network topology to be a planar graph (edges do not intersect each other). The graph planarization divides the graph into faces. The face that contains the imaginary line SD , where S is the failure node, and D is the destination node, is traversed by right/left-hand rule until a node A closer to destination than S is met. It has been shown in [13] that face routing, when applied on Gabriel graph, guarantees recovery while traversing the first face. Greedy routing continues from A until delivery or another failure node is reached.

EtE [14] combines advantages of localized greedy, power aware and recovery approaches. In the greedy part, the node currently holding a packet computes the cost of the energy-weighted shortest path (ESP) towards each of its neighbors in the forwarding direction of the destination. It then selects the one with minimum cost of ESP over the progress and sends the message to the first node on that ESP. For face routing, a connected dominating set (CDS) is first constructed, and every edge on Gabriel graph over CDS, traversed by face routing, is followed through an ESP (over all common neighbors of edge endpoints, regardless of their CDS status). CDS construction algorithm [15] is applied. Each node u constructs subgraph of its neighbors with higher IDs. If that subgraph is empty or disconnected, u is in CDS.

III. ASSUMPTIONS AND NOTATIONS

We suppose that sensor nodes are able to tune their range between 0 and R . Let $N(u) = \{v \mid |uv| \leq R\}$ be the set of physical neighbors (within communication range) of node u , where $|uv|$ is the Euclidean distance between nodes u and v .

Let $\mathcal{S} = \{s_i\}, i = 1, \dots, k$ be the set of k sinks or actors in the network. We denote by $S(u) \in \mathcal{S}$ the closest sink to sensor u : $|uS(u)| = \min_{s \in \mathcal{S}} |us|$.

We assume that sensors are aware of the physical locations of themselves, their neighbors and every sink/actor. All sensors, and sinks/actors are assumed to be static. These assumptions allow us to describe concise path based anycasting solutions, as otherwise changing any of the assumptions leads to complicated solutions or violation of guaranteed delivery property. To guarantee delivery, our algorithms also require a natural condition that reporting sensor is connected (under transmission range R) to at least one of sinks/actors.

The power consumption cost for transmitting a packet at distance r [16] is $\text{cost}(r) = r^\alpha + c$ if $r \neq 0$, or 0 otherwise,

where c is the overhead due to signal processing, and $\alpha \geq 2$ is the signal attenuation power degree. Let $P = x_0, \dots, x_n$, be a path from $u = x_0$ to $d = x_n$ ($x_{i+1} \in N(x_i), 0 \leq i < n$). The cost of the path P is $\text{cost}(P) = \sum_{i=0}^{n-1} \text{cost}(|x_i x_{i+1}|)$. The energy-weighted shortest path (ESP) from u to v is denoted by $\text{ESP}(u, v, A)$, where A is the set of intermediate nodes which can be involved in the ESP. For brevity, we denote $\text{cost}(\text{ESP}(u, v, A))$ by $\text{cost}(u, v, A)$. We denote by $f\text{SP}(u, v, A)$ the first node on the ESP from source u to destination v using nodes from A .

IV. CONTRIBUTIONS

In this paper, we propose three geographic anycast algorithms: GFGA, COPA, and EEGDA which are inspired, respectively, by GFG [5], COP [12] and EtE [14]. Each of them consists of greedy and recovery phases, and they all guarantee delivery for a reporting sensor connected to at least one sink/actor. GFGA uses hop count as the metric, while others apply power consumption, where both greedy and recovery steps are energy efficient. The two energy-efficient algorithms have different computing complexities. All algorithms construct a path from the source sensor node to one of sinks/actors. During the path construction, there exists a single destination to reach. The main feature is that this destination may change along the path according to the network topology. Anycasting may start from current node s toward sink/actor $S(s)$ that is the closest to it. However, s could in fact be even disconnected from $S(s)$ or closer in number of hops to another sink.

A. Guaranteed Delivery Anycasting

This protocol, GFGA in the following, is based on greedy [10] and face recovery [5] routing approaches. Both steps need to reduce the distance between the node holding the packet and its closest sink/actor.

1) *Greedy step*: Suppose that the distance from node u currently holding a packet to its closest sink $S(u)$ is $d = |uS(u)|$. In the greedy step of GFGA, GreedyA, at each hop, node u considers all its neighbors v whose distance $|vS(v)|$ to their corresponding closest sinks $S(v)$ is $< d$. Then, among these candidate nodes, s selects the one whose distance $|vS(v)|$ is the smallest. Note that nodes v such that $|vS(v)| > d$ may also be suitable candidates for forwarding, aiming at a different sink than u . On Fig. 1, node f is reporting. Node f forwards report to its neighbor i that has the smallest distance to its closest actor S_1 (whereas $S(f) = S_3$). Greedy routing then proceeds to sensor n and report is delivered to actor S_1 .

2) *Recovery step*: Algorithm 1 formally describes the GFGA algorithm. Recovery mode is applied with procedure $\text{Next_Face}(v, u, S(u))$ which returns the next node v on the traversed face, where face traversal started at node u with the goal of reducing distance $|uS(u)|$. It has two possible outcomes. One is to find a node v at distance $|vS(v)|$ to its own closest actor/sink $S(v)$ smaller than the initial distance $|uS(u)|$ when recovery mode started, in which case the algorithm can go back to the greedy mode. Recovery, if any, occurs on the first traversed face, so there is no change of face. However, node u , where recovery mode started, may be disconnected

from its closest sink/actor $S(u)$, while being connected to some other one. In this case, call to $\text{Next_Face}(v, u, S(u))$ ends up in a cycle detected upon return to the initial node u while traversing the face containing imaginary line from u to $S(u)$ (it is then an open face 'behind' line $uS(u)$; see details in [13]), and in such a case, it returns unchanged value for u . The closest actor $S(u)$ needs to be deleted from set \mathcal{S} at node u , because it is not reachable by u . Node u then selects new closest actor/sink $S(u)$ and starts new greedy step toward it (note that it may switch immediately to recovery phase if there is no advance in greedy mode). When the closest sink/actor is connected to u , face traversal will be successful and greedy anycasting may resume. Node u that discovers that actor/sink $S(u)$ is disconnected from it needs to inform somehow other nodes in its neighborhood about it, to prevent them from forwarding decisions and possible looping based on their belief that $S(u)$ is still connected.

Algorithm 1 GFGA(s, \mathcal{S}) Hop count based anycasting.

```

1:  $u \leftarrow s$ 
2: while  $u \notin \mathcal{S}$  and  $\mathcal{S} \neq \emptyset$  do
3:    $v \leftarrow \text{GreedyA}(u)$ 
4:   if  $u = v$  then
5:     {Greedy anycasting fails, switch to recovery step.}
6:     repeat
7:        $t \leftarrow \text{Next\_Face}(v, u, S(u)); v \leftarrow t$ 
8:     until  $u = v$  or  $|vS(v)| < |uS(u)|$ 
9:     if  $u = v$  then
10:      { $u$  disconnected from  $S(u)$  and cycle detected.}
11:      Delete  $S(u)$  from  $\mathcal{S}$  { $S(u)$  changes.}
12:     end if
13:   end if
14:    $u \leftarrow v$ 
15: end while
    
```

B. Energy-efficient guaranteed delivery anycasting

We propose two energy-aware versions of GFG for anycasting: COPA (Cost-Over-Progress GFG Anycasting) and EEGDA (Energy-Efficient Guaranteed Delivery Anycasting). Both of them use the same recovery step, based on the one in EtE [14] and differ in the greedy step they apply. Indeed, COPA is based on COP [12] greedy routing while EEGA applies a variation of the EtE greedy routing for anycasting.

1) *Greedy step*: As in GFGA, in greedy steps of COPA and EEGA, at each hop, current node u considers all its neighbors v which verifies $|vS(v)| < |uS(u)|$.

COPA-Greedy: In the greedy phase of COPA, node u selects its neighbor v which minimizes COP ratio $\frac{\text{cost}(|uv|)}{|uS(u)| - |vS(v)|}$.

Energy-Efficient Greedy Anycasting (EEGA): EEGA algorithm, described by Algo. 2, is an enhancement of COPA-Greedy algorithm. It is more energy-efficient but has higher computing complexity. The main idea is to use shortest paths both for evaluating the next target and for reaching this target. Hence, we apply the ESP using only nodes closer to their sink/actor than the current node. This ensures a positive progress at each step and preserves from looping without needing to embed the path in the packet. A node u , (node f on Fig. 1) holding a packet, first selects its target neighbor t which minimizes the ratio of the cost of the ESP from u to t and the progress provided by t . Once t is determined,

(node i on figure), node u sends the message to the first node $v = \text{fSP}(u, t, A)$ on $\text{ESP}(u, t, A)$ where A is the set of neighbors of u with positive progress. On Fig. 1, node f forwards its message to node m , (while COPA-Greedy chooses node e) where greedy power aware anycasting continues.

Algorithm 2 EEGA(u) - Selects next hop v at node u .

```

1: if  $u \in \mathcal{S}$  then
2:   exit {Greedy step has succeeded.}
3: else
4:    $A \leftarrow \{v \in N(u) \mid |vS(v)| < |uS(u)|\}$ 
5:   if  $A = \emptyset$  then
6:     Return  $\{u\}$  {Greedy step has failed.}
7:   else
8:      $v \leftarrow w \in A$  which minimizes  $\frac{\text{cost}(u, w, A)}{|uS(u)| - |wS(w)|}$ 
9:     Return  $\{\text{fSP}(u, v, A)\}$ 
10:  end if
11: end if
    
```

2) *Recovery step*: As observed in [14], face routing [5] guarantees packet delivery but is energy consuming. Planarization via Gabriel graph (GG) favors short edges, whose mandatory use in recovery mode is power inefficient. We propose here energy efficient scheme for anycasting: Energy-Efficient Face Anycasting (EEFA). Like in EtE [14], this recovery algorithm needs a preprocessing step which computes a Gabriel Graph over a connected dominating set. The use of the CDS allows to increase edge length in Gabriel graph, leading to selection of better edges for power efficiency. Each node either belongs to the CDS or has at least one neighbor in the CDS. All sinks are included in CDS. The recovery source node is also added to CDS. Face routing is then used to determine the next hop v for node u . However, instead of sending message from u to v directly, packet follows the ESP between them.

Fig. 1 illustrates EEFA recovery algorithm, with b as the source, and $S(b) = S_3$. Edge bd is to be followed in GG over CDS. Node d is then reached through the shortest path, via node c . The closest sink to c is S_3 but $|cS_3| > |bS_3|$ so recovery phase continues. Node c identifies next hop a , $S(a) = S_3$ and $|aS_3| > |bS_3|$. a then identifies its next forwarder node m in the GG over CDS. m is reached via a ESP going through node j . $S(m) = S_2$ but $|mS_2| > |bS_3|$ so

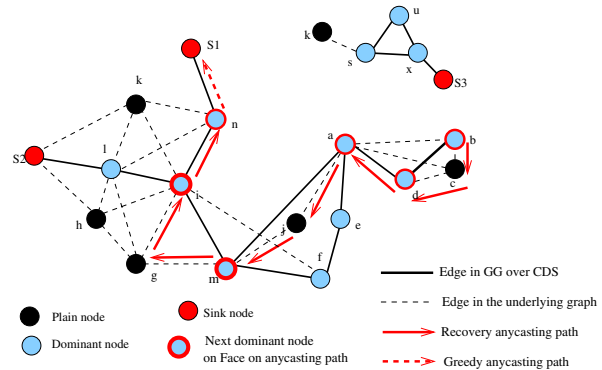


Fig. 1. Example of EEFA - 1: Extraction of a CDS, including failure node b . 2: GG over CDS. 3 - Energy aware anycasting path.

recovery phase continues. m identifies node i which is reached via with smaller cost path mgi , $S(i) = S_2$ but $|iS_2| > |bS_3|$. i continues face anycasting and forwards to node n . We then have $S(n) = S_1$ and $|nS_1| < |bS_3|$. The message is finally delivered to S_1 by greedy anycasting from n . Algorithm 3 formally describes EEFA.

Algorithm 3 EEFA(u, s) Face anycasting from failure node u to node s .

```

1:  $G' \leftarrow \text{CDS}(G)$  {Extract a CDS graph  $G'$  from  $G$ .}
2:  $G'' \leftarrow \text{GG}(G')$  {Build the Gabriel graph  $G''$  of  $G'$ .}
3:  $d \leftarrow |us|$ ,  $v \leftarrow u$  { $v$  denotes the current node.}
4: repeat
5:    $t \leftarrow \text{Next\_Face}_{G''}(v, S(u))$ 
6:   { $t$  is the next node on the proper face on  $G''$ .}
7:   while  $v \neq t$  do
8:      $v \leftarrow \text{fSP}(v, t, N(v) \cap N(t))$ 
9:     {packet follows the ESP from  $v$  to  $t$ .}
10:  end while
11: until  $|vS(v)| < d$  or  $v = u$ 
12: Return  $\{v\}$ 

```

When greedy anycasting fails at a node u such that $|uS(u)| = d$, we run EEFA till reaching a node v such that $|vS(v)| < d$ to ensure the delivery. If $S(v)$ is not reachable, it is removed from \mathcal{S} at node v like in GFGA. Once such a node v is reached, greedy routing is run again till final delivery or till triggering again the recovery phase, as described in Algo. 4. It has two names, depending on the choice of greedy step. EEGDA uses EEGA while COPA uses COPA-Greedy.

Algorithm 4 EEGDA(s, \mathcal{S}) or COPA(s, \mathcal{S}) - Power aware anycasting from source s .

```

1:  $u \leftarrow s$ 
2: while  $u \notin \mathcal{S}$  and  $\mathcal{S} \neq \emptyset$  do
3:    $v \leftarrow \text{COPA-Greedy}(u)$  or  $\text{EEGA}(u)$ 
4:   if  $u = v$  then
5:     {Greedy anycasting fails, switch to recovery step.}
6:      $v \leftarrow \text{EEFA}(u, S(u))$ 
7:     if  $u = v$  then
8:       { $u$  disconnected from  $S(u)$  and cycle detected.}
9:       Delete  $S(u)$  from  $\mathcal{S}$  { $S(u)$  changes.}
10:    end if
11:  end if
12:   $u \leftarrow v$ 
13: end while

```

V. ALGORITHM ANALYSIS

In this section, we prove basic guaranteed delivery property of the algorithms introduced in this paper.

Lemma 1: Greedy algorithms, GreedyA, COPA-Greedy and EEGA (Alg. 2) are loop free.

Proof: All three greedy algorithms reduce the distance between current node u and its closest sink $S(u)$. Loop cannot be created, since it is impossible to go back to a larger distance at a given node by repeated applications of greedy algorithm. ■

Theorem 2: GFGA, COPA and EEGDA guarantee delivery whenever source node is connected to at least one actor.

Proof: GFGA consists of repeated application of greedy and recovery phases until message is delivered. Greedy step is only applied if distance of current node u to its closest

actor/sink $d = |uS(u)|$ can be reduced. The recovery step also has the same goal, to reduce distance d after following a face. Gabriel graph preserves connectivity, and following very first face will either recover (reduce distance d) or lead to a cycle [13]. In the first case, distance d continues to decrease, and loop cannot be created. In the second case, one of actors/sinks is deleted from the network (by separate scheme as described), because it is disconnected from current node, and closest distance $d = |uS(u)|$ is reinstated at a larger value. It then continues to decrease until either delivery or discovery of a new cycle. Since the number of actors/sinks is finite (and source is connected to at least one sink/actor by the assumptions stated for our problem statement), one of such iterations will succeed. The delivery is guaranteed since d , at each iteration, can always be reduced, until it eventually becomes 0, or another cycle is detected.

COPA and EEGDA also consist of greedy and recovery phases that again reduce distance d . CDS construction preserves connectivity of selected sensors, and also includes all actors in the set. Algorithms then simply run on smaller connected sets. The only other change is that every edge, in both greedy and recovery phases, is followed by shortest paths between them, not necessarily by direct link. It is followed only via common neighbors, if any exists, of both endpoints. This guarantees that such paths always exist (sometimes the best or only choice is simply the edge in Gabriel graph). ■

VI. SIMULATION RESULTS

In this section, we evaluate the performance of our algorithms: GFGA, COPA and EEGDA. We use the energy consumption model described in Section III with $c = 10^7$ and $\alpha = 4$. The evaluation is normalized by computing energy or hop count overhead of anycasting protocols using as reference the optimal centralized Dijkstra [17] ESP or SP resp.. Let e_i and e^* be the energy consumed (resp. the number of hops) using any described protocol and the ESP protocol (resp. SP), respectively. The energy consumption overhead (ECO) (resp. the Hop Count Overhead (HCO)) is the ratio $\frac{e_i - e^*}{e^*}$. We use a home-made simulator that assumes no packet collision. We deploy N nodes at random over a 1000×1000 square around a void crescent area, in order to provoke the triggering of the recovery step. N is obtained from a Poisson law $\delta = N \times \frac{R^2 \pi}{A}$, where δ is average node degree, A is area and R is transmission radius. Nodes can adapt their range between 0 and $R = 100$. Sinks are randomly chosen among network nodes. Source node is randomly chosen among the nodes which are further than 2 hops away from any sink. Only connected source-destination pairs are considered in our measurements. Note that when the number of sinks is set to 1, anycasting yields to unicast routing.

A. ECO with respect to network density

Fig. 2 shows ECO of every algorithm. We observe that GFGA has high ECO, because steps are suboptimal. It tends to select too short edges in recovery mode and too long edges in greedy mode, compared to energy optimal length. ECO of two energy-efficient anycasting algorithms decreases with the network density since the GG over a DS provides sufficiently long edges so that energy efficient anycasting algorithms can

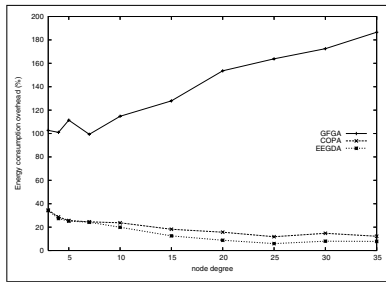


Fig. 2. ECO of anycasting paths as a function of the network density.

compute an energy shortest path. For low densities, EEGDA and COPA achieve similar results. This is because they use the same recovery algorithm which is often triggered. At higher densities greedy phase prevails, and the difference in performance is already explained.

B. ECO with respect to the number of sinks

Fig. 3 shows that ECO of energy-efficient algorithms decreases when the number of sinks increases. This is due to the fact that algorithms are more likely to find a closer sink. EEGDA achieves better results than COPA. However the gap between both algorithms appears constant. Both of them are very efficient, considering they are localized, as the overhead is under 25% for densities > 5 .

C. Hop count overhead

Fig. 4 plots the HCO of every algorithm, when the node density increases for 3 sinks (a) and when the number of sinks increases for a fixed node density $\delta = 10$ (b). Results show that the HCO of every algorithm decreases when the node density and the number of sinks increase. This is because the denser the network or the more sinks, more choices for greedy steps are given. We note that GFGA is very close to the optimal number of hops as the overhead is under 25% for densities over 15 or for more than 5 sinks for $\delta = 10$.

VII. CONCLUSION AND PERSPECTIVES

In this paper, we introduce three geographic guaranteed delivery localized anycasting protocols: GFGA, COPA and EEGDA. GFGA anycasting algorithm uses hop count as metric. It is simple and provides short paths in terms of number of hops. COPA and EEGDA are energy-efficient and provide paths whose energy consumption is less than 15% higher than the optimal one based on shortest weighted paths. Moreover, to the best of our knowledge, they are the first geographic anycasting protocols which guarantee delivery.

The next step of this work is to relax assumptions. First, we will study in more detail possible resolutions for the case when

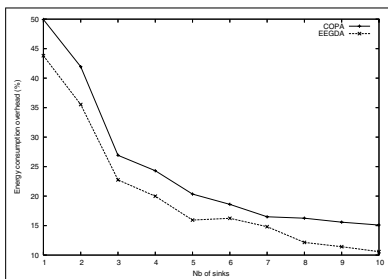


Fig. 3. ECO as a function of the number of sinks for $\delta = 10$.

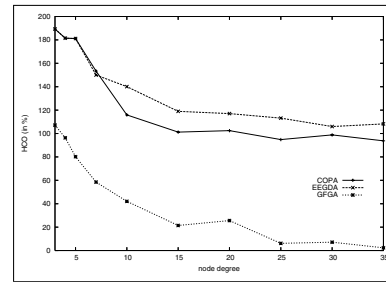


Fig. 4. HCO for every greedy algorithm. 3 sinks

a node discovers that actor/sink is disconnected from it and needs to inform other nodes to prevent from later forwarding toward it. Another interesting problem, for future work, is to consider the case of mobile actors/sinks. The difficulties arise from imprecise location information at sensors, and from change in their connectivity status toward different parts of the networks. Mobile sensors pose further formidable challenges. Finally, the impact of having imprecise location information, even for static sensors and actors/sinks, is not negligible. All these mentioned problems exist already for the case of one actor/sink, which corresponds to localized routing.

REFERENCES

- [1] B. Awerbuch, A. Brinkmann, and C. Scheideler, "Anycasting and multicasting in adversarial systems," Dept. of Computer Science, Johns Hopkins University, Tech. Rep., 2002.
- [2] W. Hu, N. Bulusu, and S. Jha, "A communication paradigm for hybrid sensor/actuator networks," *Journal of Wireless Information Networks*, vol. 12(1), pp. 47–59, 2005.
- [3] S. Chen, C. Dow, S. Chen, J. Lin, and S. Hwang, "An efficient anycasting scheme in ad-hoc wireless networks," *Consumer Com. and Networking Conf.*, pp. 178–183, 2004.
- [4] T. Melodia, V. C. G. D. Pompili, and I. Akyildiz, "A distributed coordination framework for wireless sensor and actor networks," in *ACM Mobihoc*, 2005, pp. 99–110.
- [5] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *Wireless Networks*, vol. 7(8), pp. 609–616, 2001.
- [6] C.-J. Wu, R.-H. Hwang, and J.-M. Ho, "A scalable overlay framework for Internet anycasting service," in *Symp. on Applied Computing (SAC)*, New York, USA, 2007, pp. 193–197.
- [7] M. Koziuk and J. Domaszewicz, "Tree-based anycast for wireless sensor/actuator networks," in *Int. Conf. on Distr. Computing and Networking (ICDCN)*. LNCS, 2008, pp. 322–331.
- [8] P. Jeon and G. Kesidis, "GeoPPRA: An energy efficient geocasting protocol in mobile ad hoc networks," in *Int. Conf. on Networking*, 2007.
- [9] I. Stojmenovic, "Energy conservation in sensor and sensor-actuator networks," *Wireless Ad Hoc Networking (Y.C. Tseng, S.L. Wu, eds., Auerbach Publ.)*, pp. 107–133, 2007.
- [10] G. Finn, "Routing and addressing problems in large metropolitan-scale," *Internetworks, ISI ISU/RR-87-180*, 1987.
- [11] I. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *IEEE TPDS*, 12(11), pp. 1122–1133, 2001.
- [12] I. Stojmenovic, "Localized network layer protocols in sensor networks based on optimizing cost over progress ratio," *IEEE Networks*, vol. 20(1), pp. 21–27, 2006.
- [13] H. Frey and I. Stojmenovic, "On delivery guarantees of face and combined greedy-face routing in ad hoc and sensor networks," in *ACM MOBICOM*, New York, USA, 2006, pp. 390–401.
- [14] E. H. Elhafi, N. Mitton, and D. Simplot-Ryl, "Energy Efficient Geographic Path Discovery With Guaranteed Delivery in Ad hoc and Sensor Networks," in *IEEE PIMRC*, 2008.
- [15] J. Carle and D. Simplot-Ryl, "Energy efficient area monitoring by sensor networks," *IEEE Computer Magazine*, vol. 37, 2004.
- [16] V. Rodoplu and T. Meng, "Minimizing energy mobile wireless networks," *IEEE JSAC*, vol. 17, pp. 1333–1347, 1999.
- [17] E. Dijkstra, "Solution of a problem in concurrent programming control," *Com. of the ACM*, vol. 8(9), 1965.