

# Geographic GReedy Routing with ACO Recovery Strategy GRACO

Mouna Rekik<sup>1,2</sup>(✉), Nathalie Mitton<sup>1</sup>, and Zied Chtourou<sup>2</sup>

<sup>1</sup> Inria Lille – Nord Europe, Villeneuve-d’Ascq, France  
{mouna.rekik,nathalie.mitton}@inria.fr

<sup>2</sup> CMERP, Sfax, Tunisia  
ziedchtourou@cmerp.net

**Abstract.** Geographic routing is an attractive routing strategy in wireless sensor networks. It works well in dense networks, but it may suffer from the void problem. For this purpose, a recovery step is required to guarantee packet delivery. Face routing has widely been used as a recovery strategy since proved to guarantee delivery. However, it relies on a planar graph not always achievable in realistic wireless networks and may generate long paths. In this paper, we propose **GRACO**, a new geographic routing algorithm that combines a greedy forwarding and a recovery strategy based on swarm intelligence. During recovery, *ant* packets search for alternative paths and drop pheromone trails to guide next packets within the network. GRACO avoids holes and produces near optimal paths. Simulation results demonstrate that GRACO leads to a significant improvement of routing performance and scalability when compared to the literature algorithms.

**Keywords:** Wireless sensor networks · Geographic routing · Guaranteed delivery · Swarm intelligence · Ant colony optimization

## 1 Introduction

Wireless sensor networks (WSN) consist of a large number of densely deployed sensors that have communication, computing, and sensing capacities [1]. Although sensors have power and memory constraints, they are multi-functional with sensing, wireless communication, computation capabilities and low-cost devices. For these reasons, WSNs are widely used in many fields as military surveillance, disaster prediction, and environment monitor [1]. WSNs require efficient routing protocols that adapts to the unpredictable and highly dynamic environment. The network topology may change dynamically due to node mobility, node failure and various physical properties related to the propagation channel (e.g., obstructions, noise, and power limitations) [2].

Geographic routing [3] is an attractive routing technique for large scale wireless sensor networks due to its low overhead, high scalability and memory-less features. Unlike topology-based routing, it uses only local information about the geographic location of nodes to determine, at each step, the next node to forward the packet. Greedy geographic routing schemes route data closer to the

destination at each step of the routing. This can lead to stuck nodes in case of the current node fails to find a closer node to the destination. Hence, geographic routing algorithms usually combine a greedy forwarding strategy with a recovery mechanism to solve void problem.

In this paper, we present GRACO a new geographic routing protocol that combines a modified greedy forwarding (GR) phase and an Ant-Colony-Optimization (ACO)-based recovery strategy. GRACO greedy forwarding is assisted by pheromone trails from previous recovery phases, which will prevent to return to the same stuck node. The ACO based recovery phase will use ants to discover alternative paths if greedy is not possible. The *ant* packets are sent around the void searching for route to a closer node than the stuck node to destination, then the protocol can switch back to greedy mode and until the destination or a new void. GRACO is a geographic routing algorithm that dynamically avoids holes and maintains near optimal paths around them. Results show that GRACO can reduce path lengths up to 60% compare to state of the art.

The remainder of this paper is as follows. Section 2 discusses the most relevant related work in geographic routing, recovery techniques and ACO based routings. Section 4 presents GRACO. Section 5 analyzes simulation results where CRACO and GFG routing are compared. Finally, Sect. 6 concludes this work.

## 2 Related Work

This section browses the different concepts of the literature used in GRACO.

### 2.1 Geographic Routing

Geographic routing [3] is a routing concept that exploits geographic information instead of topological connectivity. Generally, only one-hop geographic information is needed to make routing decision. Geographic routing is thus memoryless, does not require the establishment or maintenance of complete routes and nodes do not have to store routing table or write additional information in the packet. There is no need to transmit routing messages to update route states either. Its localized and stateless features make it simple and scalable. A geographic forwarding strategy defines the next hop to which forward the packet using only geographic information. Greedy, MFR [4], NFP [5] and Compass [6] are the most famous geographic forwarding strategies, they differ in the way to exploit geographic information to forward a packet toward destination: the geographic distance, the largest projection, the shortest hop or the angle respectively. Greedy [7] forwards the message to the neighbor that minimizes the Euclidean distance to the destination in each step [3]. Geographic Greedy forwarding is loop free. Indeed, at each step, the message has to move toward the destination and thus can not loop by going through a node it has already visited [3]. However, Greedy forwarding may not always be possible if all neighboring nodes are further away from the destination than the sender itself. This problem is called communication void, local maximum phenomenon or local

minimum phenomenon. It is caused by deployment holes where the forwarding process is blocked at a node called stuck node. The occurrence of hole can be caused by many factors, such as sparse deployment, physical obstacles, node failures, communication jamming, power exhaustion, and animus interference [8]. An alternative mode called recovery mode is then needed to guarantee delivery otherwise the packet has to be discarded and the delivery fails.

## 2.2 Recovery Techniques

Many studies have been focusing on the communication void problem and different solutions that guarantee delivery have been introduced. The most famous and used one is face routing [9], the first geographic routing algorithm to guarantee message delivery without flooding [3]. Face routing [3,10] is applied on a plane sub-graph of the network graph, a sub-graph where no edges intersect each other. A plane graph divides the plane into faces. The line segment between the source node and the destination node intersects some faces then, the packets will be forwarded along the boundaries of these faces. Given the advantages of face routing, it has been combined with a greedy routing approach to guarantee delivery. Indeed, when the greedy fails to forward a packet, face routing is used as a recovery mechanism. Afterward, several routing algorithms using the combination greedy face routing were proposed [3]. Although face guarantees delivery, it is energy-consuming since it may generate long detours and makes the packets follow a succession of short edges [11].

## 2.3 Ant Colony Optimization

Ant colony optimization (ACO) is a bio-inspired approach from ants foraging behavior. Indeed, real ants are able to find the shortest path between their nest and a food source without any visible, central and active coordination mechanisms. They drop pheromones, a natural chemical substance, on the path. The path optimization is achieved by exploiting the pheromone quantity deposited. Then, ants select a path based on the pheromone concentration deposited on the set of paths found. The higher the concentration of pheromone on a path, the greater the probability to select it. This indirect communication mechanism is called *stigmergy*. In addition to that, real ants show an impressive behavior when finding obstacles on their way. Actually, they are able not only to avoid obstacles, but also to find a shortest path around them.

ACO based approaches are very effectively applied to NP-hard problems and results in good optimization. Networking field is one of many domains that investigated ACO-approaches to design multi-objective and multi-constraint routing protocol and solve issues like mobility, path optimization, resource utilization and energy awareness. ACO was mainly proposed by Dorigo in his thesis [12,13], and it was widely used to solve network data routing problems. Many routing protocols based on ACO meta-heuristic were proposed in the literature [14–17]. ARA (Ant Colony Routing Algorithm) [18] the first ACO-based routing algorithm aims to reduce routing overhead and most of the existing ACO based

routing techniques in WSN and mobile ad-hoc networks are derived from this algorithm [19]. ACO based routing uses two types of agents, forward ant packets and backward ant packets. The first type of ants is used to discover paths toward destination and the second one aims to drop pheromone trails on the established path between the source node and the destination node. Although ACO-based routing algorithms solve many problems such as multiconstraints and multiobjective routing, in addition to path optimization, these algorithms usually produce high overhead, since, in order to converge to an optimized solution, they need a colony-like behavior, i.e. a huge number of ant-like packets.

In this paper, we introduce a new geographic routing algorithm that combines a modified greedy forwarding and a recovery technique based on ACO.

### 3 Notations and System Models

We assume that all nodes are aware of their location through an hardware device such as GPS or any other location mean.

We model the wireless sensor network as a directed graph  $G = (V, E)$ , composed of a finite set  $V$  of sensors, called also nodes, and a finite set  $E$  of links. There exists a wireless link between nodes  $u$  and  $v$  ( $uv \in E$ ) if  $U$  and  $v$  are within transmission range of each other, i.e.  $|Uv| \leq R$ , where  $|Uv|$  represents the Euclidean distance between  $U$  and  $v$ . The physical set of nodes which are in the transmission range of node  $U$  is noted  $N(U)$  and called the neighborhood of node  $U$ .  $N(U) = \{v \in V \text{ such as } |Uv| \leq R\}$ . We note  $|N(U)|$  the number of neighbors of  $U$ . We also define  $N_D(U)$  the subset of  $N(U)$  in which each node is nearer from node  $D$  than  $U$  itself, i.e. such that:  $N_D(U) = \{v \in N(U) \text{ such as } |vD| \leq |UD|\}$ . The directed link from a node  $U$  to a node  $v$  is noted  $\vec{Uv}$ .

In the following, we call ‘‘current node’’ the node that has a packet to route and we use *NextNode* to refer to the next hop in the path of a packet. We note  $C(A, r)$  the circle  $C$  of radius  $R$  and center at  $A$ .

## 4 GRACO

### 4.1 GRACO Principles

GRACO with ACO based recovery routing protocol (GRACO) is a geographic routing algorithm that combines two modes: a greedy mode and an ACO-based recovery mode. A data packet is first routed using a modified greedy routing that accounts for the pheromone trails. If a pheromone trail exists for a given destination, it is used to select the next node. Otherwise, the next hop is selected using plain greedy forwarding strategy. However, if the packet reaches a stuck node, the node launches an ACO-based recovery and sends some exploratory ants (Fant) to find a path. The stuck node waits for a Bant to come back, if so, a path is established and data packets can be sent. The data packet will be routed using the same strategy as the Fant until arriving to the unstuck node,

and then switch to greedy forwarding again until its destination or another stuck node, in which case, the same mechanisms applies again. Algorithm 1 depicts the GRACO behavior. Both steps are now detailed in the following sections.

---

**Algorithm 1.** GRACO(N,D) - Run at each node  $U$  upon reception of a message  $M(N, D)$

---

```

1: if  $U=D$  then
2:   EXIT #  $U$  is the final destination of  $M$ . The routing has succeeded.
3: else
4:   if  $U \neq N$  then
5:     Discard  $M$  - EXIT #  $U$  is not needed in the routing of  $M$ 
6:   else
7:     #  $U$  is the recipient of the packet, in charge of forwarding it to  $D$ .
8:     if  $existPh(U, D) = \mathbf{true}$  then
9:        $NextNode \leftarrow getNextNodePH(U, D)$  # There already exists a pheromone
          trail to  $D$ 
10:    else
11:      if  $N_D(U) \neq \emptyset$  then
12:         $NextNode \leftarrow N$  st  $||UD| - |ND|| = \max_{v \in N_D(U)} ||UD| - |vD||$ 
13:      else
14:        # Greedy mode fails,  $U$  launches the recovery mode.
15:         $NextNode \leftarrow Recovery(U, D)$ 
16:      end if
17:    end if
18:    Return  $NextNode$ 
19:  end if
20: end if

```

---

## 4.2 Ph-assisted Greedy Forwarding

The greedy mode of GRACO consists of a variant of the plain greedy forwarding (GF). GF is enhanced with the use of pheromone trails from older recoveries. Consider a node  $S$  that wants to send a data packet to  $D$ . Before applying GF,  $S$  checks whether it has recorded pheromone trails to  $D$  in order to avoid returning to the same stuck node. If so, that means the greedy failed to progress a previous data packet during a previous attempt for the same destination  $D$  and a recovery mode was been launched. For that reason,  $S$  will use these pheromone trails to send the packet instead of the GF. Otherwise, If there is no pheromone trail for  $D$ ,  $S$  proceeds by using the greedy method. In the example presented in Fig. 1,  $S$  sends a data packet to  $D$ , the packet is routed using greedy forwarding until arriving to  $N_4$  where it finds pheromones to  $D$ , then it uses a pheromone based forwarding which helps the packet to avoid the stuck node  $N_5$ . Thus, GRACO's greedy mode is a Ph-assisted greedy forwarding.

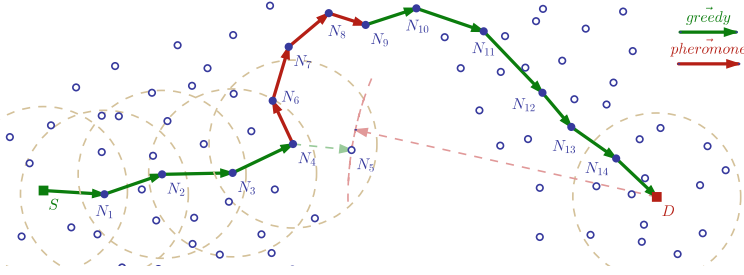


Fig. 1. Ph-assisted greedy forwarding

### 4.3 ACO Based Recovery

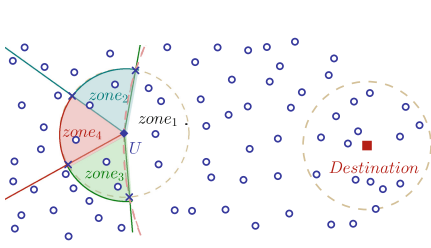
Similarly to other ACO based algorithms, the ACO recovery uses two types of ants to solve a problem: Fants (Forward-ants) to discover the environment, and Bants (Backward-ants) to “mark” the solutions found. In addition, GRACO relies on a concept of zones which plays an important role that we introduce in the following.

**Zones.** Using the concept of zones [20], a node divides its neighborhood into 4 zones based on its position and the position of D. Consider a destination node  $D$ , each node  $U$  partitions its neighbors into two main zones: the positive progress zone, later called  $zone_1$ , and a negative progress zone. As shown in Fig. 2,  $zone_1$  is represented by the intersection of the two circles  $C_1(U, R)$  and  $C_2(D, |DU|)$ . It gathers nodes in  $N_D(U)$ . Then, the negative progress zone is then partitioned into 3 sub-zones:  $zone_2$ ,  $zone_3$  and  $zone_4$ .

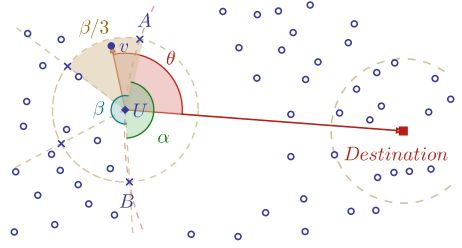
Let  $\alpha$  be the positive progress angle and  $\beta$  the negative progress angle,  $\alpha$  is the angle  $(\overrightarrow{UB}, \overrightarrow{UA})$  where  $A$  and  $B$  are the points of intersection of two circles  $C_1(U, R)$  and  $C_2(D, |DU|)$ , and  $\beta = 2\Pi - \alpha$  (Fig. 3). For a node  $v \in N(U)$ ,  $\theta$  is the angle  $(\overrightarrow{UD}, \overrightarrow{Uv})$ . Node  $v$  belongs to:

- $zone_2$  if  $\frac{\alpha}{2} < \theta \leq \frac{\alpha}{2} + \frac{\beta}{3}$
- $zone_4$  if  $\frac{\alpha}{2} + \frac{\beta}{3} < \theta \leq \frac{\alpha}{2} + 2 \cdot \frac{\beta}{3}$
- $zone_3$  if  $\frac{\alpha}{2} + 2 \cdot \frac{\beta}{3} < \theta \leq \frac{\alpha}{2} + \beta$

**Pheromone Initialization.** Before starting the route establishment phase, the amount of pheromone deposited on the links must be initialized. Each node  $U$  assigns a pheromone trail to each of its outgoing links, i.e., an initial pheromone value  $\phi_{0j}$  is assigned for each neighbor  $v$  in  $N(U)$  as  $j \in \{1, 2, 3, 4\}$  and  $zone_j$  being the zone in which node  $v$  lies, knowing that  $\phi_{01} \geq \phi_{02}(= \phi_{03}) \geq \phi_{04}$ . The motivation is that, in most cases, shortest paths pass through the neighbors whose directions are closer to the direction of the destination. Thus, the initial pheromone values being bigger as the zone is closer to the destination, direction



**Fig. 2.** Different zones of  $U$  for the destination node  $D$

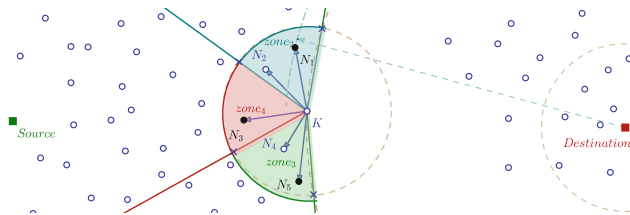


**Fig. 3.** Illustration of  $\alpha$ ,  $\beta$  and  $\theta$

will favor ants to choose these zones. As a result, this phase leads to a faster convergence to a shortest path most of the time. The ACO-based recovery strategy assumes that each node maintains a PhTable, a table of pheromone values assigned to its outgoing links for different destinations. Whenever a node receives a packet for a specific destination  $D$ , it searches in its PhTable for pheromone for  $D$ . If such pheromone trail exists, it will be used to choose the next hop. Otherwise, a pheromone initialization process is launched.

The pheromone initialization process attributes pheromone trails to all the outgoing links of a node. However, the not updated pheromone trails will be evaporated as the time passes. If a link is unused, the pheromone level on it should be completely evaporated by the end, thus, the corresponding PHTable entree is deleted. In this way, the pheromone evaporation process minimises the amount of data stored in the nodes.

**Route Establishment.** The route establishment phase is accomplished using two types of ants: the Fant and the Bant. The main role of the Fant is to explore the neighborhood in order to find an alternative path. Furthermore, it drops, on its way, a pheromone track to the stuck node to be used later by the Bant. The Bant establishes the route to the destination found by the Fant by dropping pheromone trails when it goes back to the stuck node. The route establishment starts at the stuck node  $K$  by diffusing (broadcasting) an Fant to the neighbors. The Fant will be considered and forwarded only by 3 neighbors that are selected



**Fig. 4.** Fant is considered by the closest node in each zone.

using the concept of zone and then the distance to the destination. In fact, in each zone,  $K$  selects the neighbor with the best progress to the destination. The IDs of the selected neighbors are stored in the diffused Fant. Whenever a neighbor receives the Fant, it check if it is in the list of the selected neighbors, if so it forwards the Fant, otherwise it ignores the packet. In the example presented in Fig. 4, the stuck node selects 3 neighbors,  $N_1$ ,  $N_3$  and  $N_5$ , each one is the closer to destination in its zone. Whenever a node  $U$  receives a Fant, it checks whether it is closer to the destination than the Stuck node  $K$  or not. If so ( $|UD| < |KD|$ ),  $U$  sends back a Bant, the recovery can end. Otherwise,  $U$  forwards the ant. Similar to other ACO routing algorithms [19], a node forwards a Fant to the next node using a stochastic decision which is based on the values of pheromone trails to select the next hop. Suppose that a Fant is currently residing in node  $U$ .  $U$  has  $k$  neighbors  $v_1, v_2, \dots, v_k$ . We will note  $\Phi_i$  the amount of pheromone assigned to  $v_i$  (or the link  $\overrightarrow{Uv_i}$ ). The neighbors of  $U$  will be partitioned into 4 zones. Consider  $\Phi_{zone_i}$  is the maximum pheromone amount assigned to the neighbors of  $zone_i$ ,  $\Phi_{zone_i} = \max \{\Phi_j \mid v_j \in zone_i\}$ . In order to forward a Fant, first of all,  $U$  selects a zone with probability  $P_{zone_i}$ :

$$P_{zone_i} = \frac{\Phi_{zone_i}}{\sum_{j=1}^4 \Phi_{zone_j}} \quad (1)$$

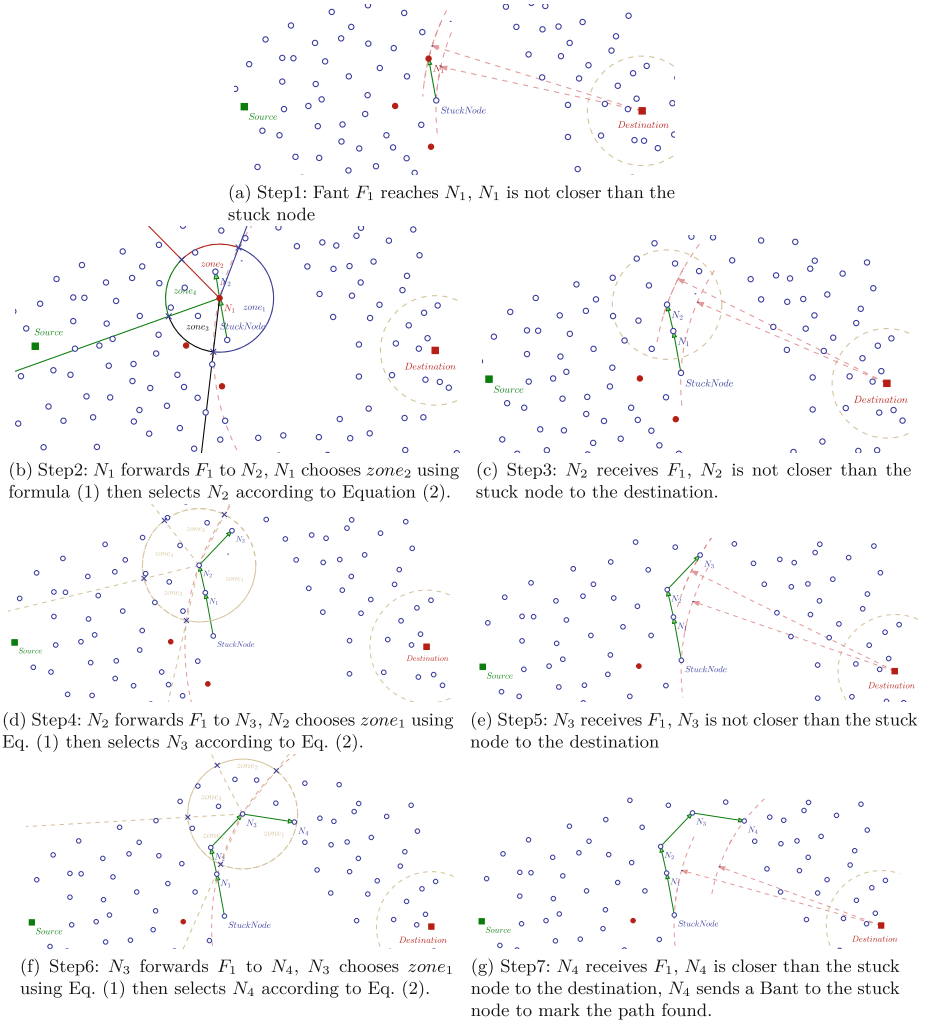
$U$  chooses a node  $v_i$  in this selected zone with probability  $P_i$ :

$$P_i = \frac{\Phi_i}{\sum_{j=1}^{|N(D)|} \Phi_j} \quad (2)$$

Using the concept of zones, the ants are not completely blind, as in the usual ACO based algorithms, they will be directed, but not forced, to the right direction. Besides the PHTable mentioned before, each node maintains also a back routing table, the BRTable, to store information that will be used later by Bants to go back to the stuck node. Whenever a Fant arrives to a node from one of its neighbors, an entry is added to the BRTable. Figure 5 shows an example of a Fant  $F$  trying to bypass a void.  $F$  is launched at the stuck node, and forwarded to  $N_1, N_2, N_3$  until reaching unstuck node  $N_4$ . To summarize, Algorithm 2 shows the forwarding strategy of the Fant until arriving to an unstuck node.

When a Fant reaches an unstuck node  $N$ , a Bant is sent back to the stuck node. An unstuck node is a node closer to destination than the stuck node. After adding an entry to the BRTable,  $N$  extracts the information from the Fant, creates a Bant and deletes the Fant. Subsequently,  $N$  sends the Bant to the stuck node  $K$ . The role of the Bant is to drop pheromone on the path found in order to establish a track from  $K$  to  $N$ . Unlike the traditional ACO based algorithm, the Bant will not return to the stuck node using the same path of the corresponding Fant, but it will use the pheromone dropped by other Fants





**Fig. 5.** An example of a Fant searching for a path around a void

that used at least one node of its path. In fact, the Bant will not only search for a path in the BRTTable, but also it will have the ability to choose the best path found (in our case the shortest). In the example presented in Fig. 6, the Fants find two valid paths to an unstuck node, the first one presented in green is a 4-hop-length path and the second one in pink is a 16-hop-length path. A Bant, in this example, chooses to use the green path to go back to K since it is the shortest path to K. On its way back to the stuck node, the Bant updates pheromone trails in the PHTables. Consider a Bant arrives to node A from node

---

**Algorithm 2.** ForwardFant(K,D) - Run at each node  $U$  upon reception of a Fant for  $D$  issued by stuck node K

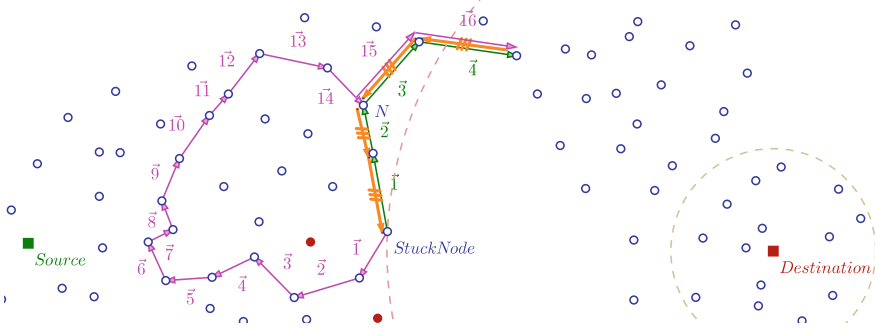
---

```

1: update_BRTTable( $U$ )
2: if  $|UD| < |KD|$  then
3:   send_Bant( $K, D$ ) # Node  $U$  allows a progress compared to the stuck node and
   can stop the recovery.
4: else
5:   if existPh( $U, D$ ) = false then
6:     initialisePh( $U, D$ )
7:   end if
8:    $NextNode \leftarrow$  getNextNodePH( $U, D$ )
9: end if
10: Return  $NextNode$ 

```

---



**Fig. 6.** Bant returns to the stuck node

B, it will update pheromone track of the link  $\overrightarrow{AB}$  in the PHTable of A:

$$\Phi_{\overrightarrow{AB}} = \Phi_B = \Phi_B + \Delta\Phi$$

where  $\Delta\Phi$  is the amount of pheromone added to reinforce the path to D, this value depends on the quality of the path. Algorithm 3 sums up the Bant forwarding.

## 5 Simulations and Results

In order to evaluate the performance of the GRACO algorithm, we compare it to the GFG algorithm under the WSNNet simulator [21]. We generate random topologies in a region of  $300 \times 300$  m region with 250 to 900 nodes and  $R = 25$  m. To evaluate the impact of voids recovery to performance, we run simulations on different void diameters. We evaluate the algorithm according to end-to-end delay, data delivery cost, delivery rate and hop count. The simulation results of GRACO are given with 95% confidence intervals.

---

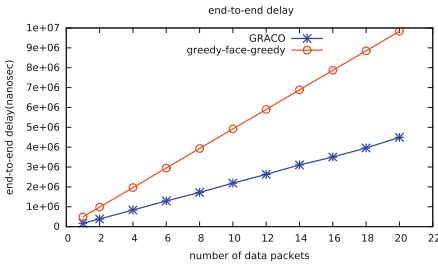
**Algorithm 3.** ForwardBant( $K,D$ ) - Run at node  $U$  upon reception of a Fant for  $K$

---

- 1: **Returns**  $NextNode$  # the next hop
  - 2:  $update\_PHtable(U)$
  - 3: **if**  $U = K$  **then**
  - 4:   kill Bant # the Bant reaches the stuck node.
  - 5: **else**
  - 6:    $NextNode \leftarrow getNextNodeBRtable(U, D)$
  - 7: **end if**
- 

## 5.1 End-to-End Delay

The end-to-end delay is the time interval between a given source sends a packet and the destination receives it. In case of a set of data packets, it represents the delay between the source sends the first data packet and the destination receives the last one. Sources and destinations are randomly chosen in a way they are in different sides of a void. Figure 7 shows the end-to-end delay varying with the number of data packets sent. We can see that the end-to-end delay of GFG algorithm is always higher than GRACO. This is mainly due to the fact that GFG tries to create a path around the void every time a packet tries to cross the void, even if the packet has the same source and destination as the previous one. However, GRACO launches the recovery only the first time when the packet is trying to bypass the void, and then the next packets will find pheromone traces for the paths found in the previous recovery phase. When the number of data packets sent increases, the difference between the delay produced by GFG and the one produced by GRACO become very large. Thus, GRACO delivers the data packets faster than GFG.



**Fig. 7.** End-to-End delay varying with the number of data packets sent



**Fig. 8.** Delivery rate varying with the number of data packets sent

## 5.2 Data Delivery Cost

Data delivery cost is the number of packets sent in the network to deliver a data packet from a source to a destination, including recovery cost. Figure 9a

presents the data delivery cost varying with the number of data packets sent. For the particular case of a single packet, the data delivery cost used by GRACO is higher than GFG. This is explained by the number of packets sent during the recovery phase in GRACO is bigger than GFG in one recovery mode. However, when the number of data packets sent between the same source and the same destination increases, GRACO becomes cheaper than GFG since GRACO launches the recovery mode only when the first data packet is sent, the next ones will use the routes already found. As for GFG, it launches the recovery mode every time a data packet is sent. Thus, the cost on GRACO for multiple packets between the same source and destination is the same as for only one packet, in contrary with GFG, the cost used to deliver multiple data packet will be the cost of sending one packet multiplied by the number of data packets sent. In Fig. 9b, it can be seen that the cost per packet decreases when the number of data packets increases. On the other hand, the cost of GFG to deliver a single packet remains constant.

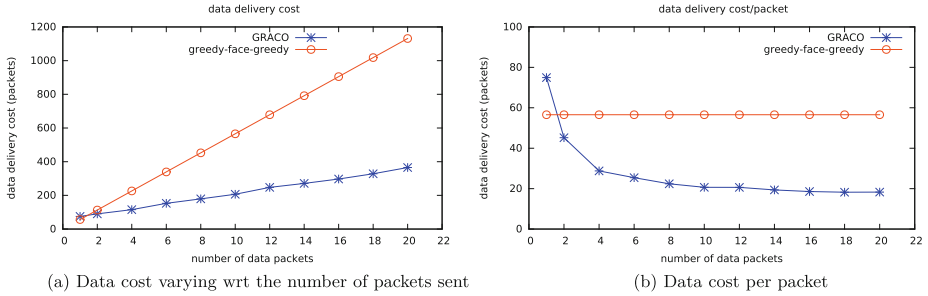


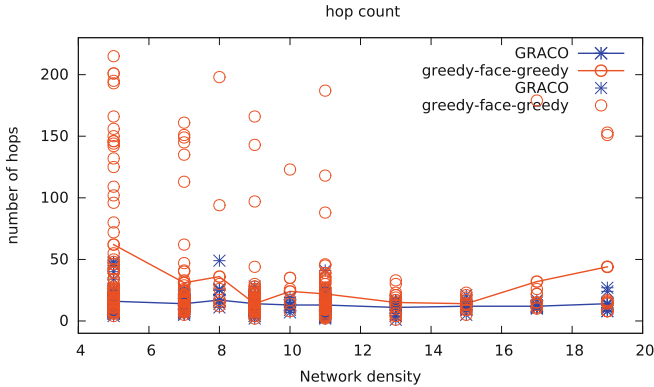
Fig. 9. Data cost

### 5.3 Hop Count

Besides the end-to-end delay and data delivery cost, we compared the length of paths found by GRACO and GFG. As presented in Fig. 10, the simulations show that GRACO creates shorter paths than the ones created by GFG. GFG may generate extremely long path in some cases [22] specially in low density networks as shown in Fig. 10. In the other hand, the ACO and the zone concept used in GRACO help the algorithm to create shorter paths than GFG, these paths are usually optimal or near optimal paths.

### 5.4 Packet Delivery Rate

Packet delivery rate is the ratio of data packets successfully received by their destinations to all data packets sent by the sources. From Fig. 8, we can see that GFG delivers all packets sent successfully. Although GRACO couldn't maintain its delivery rate at 100 %, it has a high delivery rate (more than 95 %). The main



**Fig. 10.** hop count

possible reasons for packet loss is that, in the case of multiple voids, GRACO could fail in cascading recoveries leading to sporadic data delivery failures.

## 6 Conclusion

In this paper, we presented GRACO, a new geographic routing protocol able to handle the void problem with very low added complexity. It combines a pheromone assisted greedy forwarding phase and an ACO based recovery phase to avoid holes and recover from local minima. The proposed routing protocol is fully localized, distributed, scalable, and does not require any graph structure or routes information to be maintained. Besides the efficient hole avoidance, simulations results show that GRACO delivers data packets faster and with better end-to-end delay, data delivery cost and hop count compared to GFG routing protocol. GRACO provides also a reliable delivery performance. GRACO still has some potential for future improvement. The pheromone update rule has to be improved in order to account for multiple recoveries in a single path. We intend to consider GRACO as a routing solution for Advanced Metering Infrastructure (AMI) communications in the context of smart grid. For this purpose, we plan to analyze communication requirements in this type of networks, and to implement an extension of GRACO that takes into account the QoS required.

## References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. *IEEE Com. Mag.* **40**, 102–114 (2002)
2. Dvir, A., Carlsson, N.: Power-aware recovery for geographic routing. In: *Wireless Communications and Networking Conference (WCNC)* (2009)
3. Mitton, N., Razafindralambo, T., Simplot-Ryl, D.: Position-based routing in wireless ad hoc and sensor networks. In: Nikolettseas, S., Rolim, J.D.P. (eds.) *Theoretical Aspects of Distributed Computing in Sensor Networks*, pp. 447–477. Springer, Heidelberg (2010)

4. Takagi, H., Kleinrock, L.: Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. Com.* **32**, 246–257 (1984)
5. Hou, T.C., Li, V.O.K.: Transmission range control in multihop packet radio networks. *IEEE Trans. Com.* **34**, 38–44 (1986)
6. Kranakis, E., Singh, H., Urrutia, J.: Compass routing on geometric networks. In: *Canadian Conference on Computational Geometry* (1999)
7. Finn, G.G.: Routing and addressing problems in large metropolitan-scale internetworks. *ISI Research Report* (1987)
8. Jiang, Z., Ma, J., Lou, W., Wu, J.: An information model for geographic greedy forwarding in wireless ad-hoc sensor networks. In: *Conference on Computer Communications (Infocom)* (2008)
9. Sanavullah Shri Alarmelu, M.Y., Poonkuzhali, V.R.: An efficient void handling technique for geographic routing in manet: a survey. *J. Adv. Res. Comput. Sci. Softw. Eng.* **2**(12), 164–170 (2012)
10. Guan, X.: *Face Routing in Wireless Ad-hoc Networks*. Ph.D. thesis. University of Toronto (2009)
11. Gouvy, N., Mitton, N., Zheng, J.: Greedy Routing Recovery Using Controlled Mobility in Wireless Sensor Networks. In: *International Conference on Ad hoc Networks and Wireless (ADHOC-NOW)* (2013)
12. Dorigo, M., Di Caro, G.: The ant colony optimization meta-heuristic. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimisation*. McGraw-Hill, London (1999)
13. Dorigo, M., Stützle, T.: The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Glover, F., Kochenberger, G.A. (eds.) *Handbook of Metaheuristics*, pp. 250–285. Springer, US (2003)
14. Zungeru, A.M., Ang, L.M., Seng, K.P.: Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison. *J. Netw. Comput. Appl.* **35**, 1508–1536 (2012)
15. Uchhula, V., Bhatt, B.: Article: comparison of different ant colony based routing algorithms. *IJCA Spec. Issue MANETs* **2**, 97–101 (2010)
16. Sutariya, D., Kamboj, P.: A survey of ant colony based routing algorithms for manet. *Eur. Scientific J.* **3**, 82–91 (2014)
17. Huang, R., Guanghui, X.: Swarm intelligence-inspired adaptive routing construction in wsn. In: *Wireless Communications, Networking and Mobile Computing (WiCOM)* (2010)
18. Gunes, M., Sorges, U., Bouazizi, I.: Ara-the ant-colony based routing algorithm for manets. In: *International Conference on Parallel Processing Workshops* (2002)
19. Shirkande, S.D., Vatti, R.A.: Aco based routing algorithms for ad-hoc network (wsn, manets): a survey. In: *Communication Systems and Network Technologies (CSNT)* (2013)
20. Kamali, S., Opatrny, J.: Posant: a position based ant colony routing algorithm for mobile ad-hoc networks. In: *International Conference on Wireless and Mobile Communications (ICWMC)*, (2007)
21. Fraboulet, A., Chelius, G., Fleury, E.: Worldsens: development and prototyping tools for application specific wireless sensors networks. In: *Symposium on Information Processing in Sensor Networks (IPSN)*, April 2007
22. Seada, K.: Modeling and analyzing the correctness of geographic face routing under realistic conditions. *Elsevier Ad Hoc Netw. J. Spec. Issue Recent Adv. Wirel. Sens. Netw.* **5**, 855–871 (2007)



<http://www.springer.com/978-3-319-19661-9>

Ad-hoc, Mobile, and Wireless Networks  
14th International Conference, ADHOC-NOW 2015, Athens,  
Greece, June 29 -- July 1, 2015, Proceedings  
Papavassiliou, S.; Ruehrup, S. (Eds.)  
2015, XIII, 418 p. 151 illus., Softcover  
ISBN: 978-3-319-19661-9