# GrAnt: Inferring Best Forwarders from Complex Networks' Dynamics through a Greedy Ant Colony Optimization

Ana Cristina Kochem Vendramin, Anelise Munaretto, Myriam Regattieri Delgado, Aline Carneiro Viana

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

# *GrAnt: Inferring Best Forwarders from Complex Networks' Dynamics through a Greedy Ant Colony Optimization*

Ana Cristina Kochem Vendramin — Anelise Munaretto — Myriam Regattieri Delgado — Aline Carneiro Viana

**N° 7694**

July 2011

Thème COM

*Rapport de recherche*

ISSN 0249-6399 ISRN INRIA/RR--7694--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SACLAY - ÎLE-DE-FRANCE

# GrAnt: Inferring Best Forwarders from Complex Networks' Dynamics through a Greedy Ant Colony Optimization

Ana Cristina Kochem Vendramin , Anelise Munaretto , Myriam Regattieri Delgado , Aline Carneiro Viana

Thème COM — Systèmes communicants
Équipes-Projets Hipercom

Rapport de recherche n° 7694 — July 2011 — 23 pages

**Abstract:** This paper presents a new prediction-based forwarding protocol for the complex and dynamic Delay Tolerant Networks (DTN). The proposed protocol is called GrAnt (Greedy Ant) as it uses a greedy transition rule for the Ant Colony Optimization (ACO) metaheuristic to select the most promising forwarder nodes or to provide the exploitation of good paths previously found. The main motivation for the use of ACO is to take advantage of its population-based search and of the rapid adaptation of its learning framework. Considering data from heuristic functions and pheromone concentration, the GrAnt protocol includes three modules: routing, scheduling, and buffer management. To the best of our knowledge, this is the first unicast protocol that employs a greedy ACO which: (1) infers best promising forwarders from nodes' social connectivity, (2) determines the best paths to be followed to a message reach its destination, while limiting the message replications and droppings, (3) performs message transmission scheduling and buffer space management. GrAnt is compared to Epidemic and PROPHET protocols in two different scenarios: a working day and a community mobility model. Simulation results obtained by ONE simulator show that in both environments, GrAnt achieves higher delivery ratio, lower messages redundancy, and fewer dropped messages than Epidemic and PROPHET.

**Key-words:** opportunistic forwarding, adaptive forwarding, contact prediction, mobility, delay tolerant networks

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université
4, rue Jacques Monod, 91893 ORSAY Cedex
Téléphone : +33 1 72 92 59 00

# GrAnt: Inférant les meilleur relays à partir de dynamic de réseaux complexes en utilisant une optimisation par colonies de fourmis Greedy

**Résumé :**

Cet article porte sur la proposition d'un protocole d'acheminement pour les réseaux complexes et dynamiques du type tolérants aux délais (DTN), qui est basé sur l'estimation de possibilités futures de contact. Le protocole proposé est appelé GrAnt (Greedy Ant) car il utilise une règle de transition greedy pour la méta-heuristique d'optimisation par colonies de fourmis (ACO). Cette méta-heuristique donne à GrAnt la possibilité de sélectionner les relais les plus prometteuses ou d'exploiter les bons chemins préalablement trouvé. La motivation principale pour l'utilisation de l'ACO est de profiter de son mécanisme de recherche basé sur population et de son apprentissage et adaptation rapide. En utilisant des simulations basées sur des modèles synthétiques de mobilité, nous montrons que GrAnt est en mesure d'adapter conformément son acheminement dans des différents scénarios et possède une meilleure performance comparée à des protocoles comme Epidemic et PROPHET, en plus de la génération de faible surcharge.

**Mots-clés :** routage opportuniste, acheminement adaptive, mobilit, rseaux tolrants aux dlais, estimation de contactes

# 1 Introduction

The Internet has been widely used to interconnect a variety of communication devices around the world. To provide such communication the Transmission Control Protocol/Internet Protocol (TCP/IP) [1] is used. The proper functioning of TCP/IP depends on specific environments such as wired networks, where there are continuous end-to-end connections, low-delay paths, and low error rates. In such environments selecting only a single route between a source and a destination node is sufficient to achieve an acceptable communication performance. However, new communication networks are emerging such as mobile ad-hoc networks, satellite networks, and sensor networks. Unlike the Internet, these networks are subject to constant changes in their topology due to users' mobility, obstacles and/or limited resources resulting in frequent network partitions, long and variable end-to-end delays, and high error rates [2]. Given the ubiquity of devices able to operate in these networks, it is necessary to provide interoperability between them anywhere and anytime even in the absence of a network infrastructure. To accommodate these challenges, where the assumptions necessary for the proper functioning of TCP/IP protocols are not found, it was arisen the Delay Tolerant Network (DTN) [2] [3] [4].

A DTN is a complex and dynamic environment whose objective is to support the mobility of users among heterogeneous networks. In DTNs a continuous path between a source and a destination node can not be assumed. To achieve messages delivery, a store-carry-forward communication model is employed in intermediate DTN nodes. The nodes may need to store messages from others in their buffer for long periods of time and to carry them along its path until some forwarding opportunity arises [5]. Considering these challenges, it is necessary to maintain an updated view of the DTN dynamics by periodically gathering and analyzing neighbors information and to select more than one path to forward each message while limiting the redundant message overhead.

Given that the adaptation in nature is a permanent and continuous process and that the dynamic and complex environment of DTNs is favorable for applying population-based paradigms, we propose the use of Ant Colony Optimization (ACO) in DTNs. ACO simulates the decision-making process of ant colonies as they forage to find the most efficient trails from their nests to food sources [6] [7]. It is an adaptive and self-organizing mechanism capable of performing a rapid search in a large and dynamic space while providing a diversity of solutions to the DTN forwarding problem. In this way, this paper presents a new greedy ACO-based forwarding protocol for DTN, so-called GrAnt, that directs the traffic to the most promising nodes with the aim of improving the messages delivery while limiting the message replications and droppings. The term greedy suggests the use of deterministic decision rules instead of the probabilistic ones commonly used in the ACO paradigm. In particular, GrAnt brings schemes for: gathering updates from network dynamics, determining the best paths to be followed to a message reach its destination, message transmission scheduling, and buffer space management. Through simulations, we study the performance of GrAnt when compared to Epidemic and PROPHET protocols. The simulation results show that GrAnt is able to make good enough decisions to guarantee high reliability and acceptable delay in significantly different mobility models. Specifically, GrAnt is able to outperform Epidemic and PROPHET: more successfully delivered messages, lower messages redundancy, and fewer dropped messages.

The rest of this paper is organized as follows: Section 2 reviews the existing DTN protocols and describes the ACO metaheuristic. Section 3 describes the proposed GrAnt protocol. Section 4 presents the simulation environment followed by the performance evaluation of the protocols. Finally concluding remarks are given in Section 5.

# 2 Background

The state-of-the-art of DTNs routing protocols is based on local and on two-hop information to achieve message delivery [8] [9] [10]. In this paper, a greedy *Ant Colony Optimization (ACO)* is used to direct DTN traffic through a small subset of good forwarders. To the best of our knowledge this is the first unicast work that employs a complete ACO search (biased by pheromone and heuristics values) and analyzes the most relevant information that can be gathered from DTN nodes. The proposed protocol is then called GrAnt. Before describing the GrAnt protocol, we go through the related work in the area, discussing the most representative results on both DTN and ACO-based routing.

## 2.1 DTN routing

Considering that, in a DTN, no guarantee that a fully connected path between any two nodes exists at any time, transfers of messages custody needs to be provided by nodes. Until a forwarder opportunity arises a node may need to store multiple messages in its buffer. It is also possible that only one contact is available at a time and it has not enough resource to receive all custodies. In these scenarios, typical of DTNs, the new forwarding protocols need to consider the following challenges. First, due to limited duration of each contact, it is important to determine which and in what order the messages should be forwarded when an opportunity arises. Second, if more than one contact is available at any given time, the most promising contact(s) to where each message should be forwarded to, has to be determined. If, we consider infinite buffer and network bandwidth, the greater the number of each message forwarding, the better the chance of that message be delivered to its destination. Nevertheless, resources are usually scarce in DTNs, making it necessary to determine in a dynamic way, the number of message's copies that should be forwarded to custodians. Finally, if a buffer achieves its storage capacity and a new message has to be received, it is important to correctly determine which message should be dropped to accommodate the new one, while limiting the impact on the reliability of the dropped messages.

Existent DTN routing protocols can be summarized into three categories [11]: (1) *Flooding or Controlled flooding-based* [12] [13]: forward messages to all or almost all encountered nodes without predicting which ones are good message forwarders. This category of protocols can be effective only when the nodes' buffer is infinite which is not feasible in wireless networks [14]. According to [15], the high degree of messages replication in DTNs is not conforming to the limited contact time between the nodes. Congestion can occur when a large amount of data is stored in the nodes' buffer and it is expected to transfer them when an opportunity arises; (2) *Prediction-based* [8] [9] [16] [17] [18]: try to predict which nodes are useful for delivering messages based on historical encounters between nodes, node's context information, node location-visiting pattern, and social information; (3) *Scheduling protocols* [19] [20]: rely on the complexity task of controlling the trajectory of special nodes to improve the rate of messages delivery. So, it is mainly used in sparse networks [11].

We are interested in the first two categories of protocols which can operate in various environments with different nodes mobility models. Specifically, Epidemic controlled flooding-based and PROPHET prediction-based protocols are used for performance gain comparison with GrAnt. In Epidemic [12], when two nodes meet, they exchange their current messages' vectors and request to each other the messages it haven't seen previously. To limit the resources utilization, a hop-count field can be set in each message. When the buffer reaches its maximum capacity and a new message is received, the oldest message is dropped. In Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) [8], vectors are exchanged indicating the probability of each node to deliver their messages. Messages are forwarded to nodes having higher delivery probabilities that also has a transitive property. When the buffer reaches its maximum capacity and a new message is received, the oldest message is dropped. When there is more than one message to be transmitted at a time, the message whose destination is more likely to be encountered will be transmitted first. If the messages have equal probabilities, the oldest one will be forwarded first. Differently from Epidemic and PROPHET, GrAnt takes advantage of the rapid adaptation of ACO learning framework to conduct a global search and gather relevant information from DTN nodes. In this way, GrAnt is able to analyze the utility of each contact as a message forwarder and limit the messages replications and droppings. Criteria like message priorities, number of previous message forwarding, and nodes' utility are considered when ordering the messages for forwarding and discarding.

## 2.2 Ant Colony Optimization Metaheuristic

Studies in the literature suggest the modeling of life and of natural and biological intelligence to solve complex computational problems in various fields, from engineering to biology. Such studies are inspired by the fact that individuals develop adaptive traits increasingly complex when they are in groups, than when they are alone. Since there is no comprehensive knowledge from the environment, simple individuals interact with each other and with their environment in the search for a problem solution. The intelligence emerges as a result of the pattern of simple interactions between them in time [21].

*Ant Colony Optimization* (ACO) metaheuristic is an example of an artificial swarm intelligence system which is inspired by the collective behavior of social insects [6] [7]. In ACO algorithms, usually an artificial ant collects information about a problem, stochastically makes its own decision, and constructs solutions in a stepwise way. The behavior that emerges is a group of relatively "not intelligent" ants that interact through simple rules and dynamically self-organize maintaining their positions around the shortest trails: Ants leave their nest without information about the location of food sources, move randomly at initial steps, and deposit a substance called pheromone on the ground. The pheromone marks a trail, representing a solution for the problem, that will be positively increased to become more attractive in subsequent iterations and to serve as a history of the best ants' previous movement.

### 2.2.1 ACO for Routing

The collective behavior of ants through a distributed learning of the best trails have resulted in the successful implementation of ACO in dynamic and combinatorial problems, particularly in the area of communication networks. Communication networks are becoming more complex and it is desirable that they can self-organize, self-configure and self-adapt to constant changes in their topology, traffic load, and services diversity. When designing routing protocols for these environments it is important that they provide the following properties: robustness, scalability, low computational cost, distributed search, ability to observe the network dynamics and quickly adapt to them. A system that is inherently self-organized like ACO has these properties.

In a network routing problem, artificial ants are mobile agents that incorporate intelligence when moving from one node to another to search a candidate path between a source and a destination node. According to the classification provided by [22], ACO routing algorithms for Mobile Ad-Hoc Networks (MANETs) can be differentiated with respect to: (1) How ants are created and how destinations are chosen: algorithms can adopt a *proactive* or *reactive* behavior as discussed later; (2) Which kind of information the ants can gather in each path: only the identities of visited nodes or more specific information about them; (3) Which information the ants can use to choose the next hop: pheromone concentration and/or information about crossed nodes that can be incorporated in heuristic functions, generally indicating an explicit influence toward useful local information; (4) How much pheromone the ants deposit in a path: a constant or a variable amount of pheromone depending on information gathered or depending on local parameters.

The ACO search phase in MANETs is normally reactive; It is initialized on-demand when it is necessary to establish a multihop path. In typical reactive algorithms when a node needs to send a message to a destination it will first consult its routing table to see if there is any known route to that destination. If there is not an entry, small control messages, called *Forward Ants (FAs)*, will be created and sent towards the destination via one or more of its neighbors. If more than one path exists, one will be chosen according to a transition rule: the *Ant System (AS)*'s probabilistic transition rule [6] or the *Ant Colony System (ACS)*'s pseudo-random-proportional rule [23]. In the AS, an ant decides which is its next state to move to in a randomly way with a probability distribution depending on the pheromone concentration and heuristic function, i.e. as higher the pheromone and heuristic values on a link, the higher the probability it will be selected. The ACS transition rule is composed by two sub-rules: a deterministic/greedy transition rule that provides the exploitation of priori and accumulated knowledge by choosing, in a greedy way, the best available solution (higher pheromone and heuristic values); and a probabilistic transition rule like AS. The first sub-rule is chosen with a pre-defined probability $q_0$ and the second sub-rule with probability $1 - q_0$. When the destination is reached, a *Backward Ant (BA)* is created and it stores all the information gathered by its corresponding FA. The BA is sent back to the source node along the reverse path followed by FA depositing a pheromone quantity on it. The routing table of the visited nodes is changed as a function of the followed path. ACO also considers a mechanism, called evaporation [7], that regulates the amount of pheromone deposited on paths by ants. If pheromone is not limited, the system tends to a rapid convergence. Over time, the pheromone evaporates in old paths, preventing the ants to follow it.

Reactive ACO algorithms have been extensively studied in MANETs [22] [24]. However, routing in DTN is more challenge due to very frequent partitions and long end-to-end delay. In [22] the authors introduce the ACO into DTN routing. Nevertheless, they choose the next node toward the destination based only on pheromone concentration (a constant value deposited by ants), that is, no information about the nodes in a path is considered. In addition, no pheromone evaporation is used which eventually will conduct to the usage of the same best nodes already found. Ants are sent in a constant rate and this sending process is stopped when the first good next hop is found (considering a pre-defined threshold that will characterize its quality as acceptable or not). No information about the number of message's copies is provided and the algorithm was not compared to existent DTN protocols.

Another problem associated with reactive algorithms is that initially a source node can take a long time to find a path toward a destination. This is especially true if the source is far from the target like in DTNs. Moreover, due to the DTNs dynamics, the choice of the best path is not the main objective. So, it is important to send the data message along with the ants' control message, to maintain a large number of paths, preventing the ants to always use the same intermediates nodes toward destinations, and to consider the quality of nodes to better direct the network traffic. Keeping this issues in mind, we propose a Greedy Ant (GrAnt) protocol to infer best forwarders toward destinations and thus, to better direct the DTN traffic as described in the next Section.

# 3 The GrAnt Protocol

This paper presents a Greedy Ant protocol, called GrAnt, as a solution to the problem of finding a set of nodes for routing each message. To adapt to the large variations that a DTN suffers in its topology and to reduce the latency in message delivery, traditional ACO protocols had to be modified. The following actions are incorporated into GrAnt (see Section 3.2 for more details):

- *Action 1:* to increase reliability in dynamic networks like DTNs, it is important to allow redundant paths as a tentative to avoid the convergence of the algorithm to only one or very few paths. Instead of using a time-based pheromone evaporation like traditional ACO, GrAnt performs an event-driven evaporation, which only happens if a node detects that a new path to a destination has just been found. This evaporation process seems more suitable to lossy networks;

- *Action 2:* since a source node can take a long time to find a path toward a destination, Forward Ants (FAs) are encapsulated into data messages;

- *Action 3:* unlike a classical ACO, we do not fix the number of FAs to find a path to an unknown destination. Instead, this number is completely dynamic according to the utilities of the already established messages' forwarders and the success of the messages delivery. That is, new FAs are created and sent only while the reception of its respective message is not known and when better forwarders appear;

- *Action 4:* instead of a probabilistic choice, we use a greedy ACO transition rule (where the best option is always chosen) to forward the messages to the most promising nodes or to provide the exploitation of good forwarders already found, considering heuristic functions and pheromone concentration. The exploration of the search space proposed by classical ACO is still provided by the dynamics of the DTNs.

In general, the GrAnt protocol works as follows (see Fig. 1 for its execution overview in a small network). An FA $k$ is sent together with a data message $m$ toward a destination $d$ (see Fig 1(a)). The path to $d$ is constructed based on the knowledge acquired by this FA (see Section 3.1) which dictates the forwarding decision at a node and tries to infer the capability of good next forwarders to $d$ (see Section 3.2.1). While being forwarded, each FA $k$ collects the quality information ($Q_x$) of every node $x$ composing the path to $d$ (see Fig. 1(b) and Section 3.2.1). Once the destination is found, a Backward Ant (BA) is created and sent through the reverse path indicated by the FA. The BA stores the total quality ($Q^k_{path}$) of the path found by the FA and deposits a pheromone (proportional to the total quality) at links between nodes composing the reverse path (from $d$ to the source) (see Fig. 1(c). If subsequent messages are forwarded to the same destination, the already deposited pheromone will be reinforced at those links and will help the forwarding of future FAs to the same destination (see Fig. 1(d) and Section 3.2.2).

To direct the DTN traffic to the most promising contacts, GrAnt uses information about opportunistic social connectivity between nodes. Firstly, we characterize which type of knowledge can be inferred from the DTN dynamics and later we detail how such knowledge is gathered and used by GrAnt.

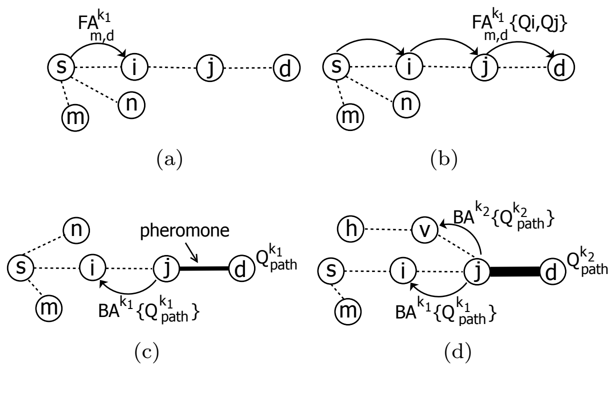



Figure 1: Overview of the GrAnt protocol execution.

## 3.1 Characterizing connectivity-based interactions of nodes

In research on complex and challenged networks, special attention has been devoted to the computational analysis of social networks [25] [26].

We believe that the degree of sociability among nodes becomes a key factor in determining opportunities for communication because in DTNs the contacts are based on social relations between people, instead of being randomly established. Therefore, some questions about contacts and their influence raised in [27] are considered by GrAnt: How many individuals know each member of a network? Whose individuals have the greatest number of contacts? Are these the most influential individuals? What is the probability that two randomly selected individuals will know each other? What is the probability that the paths between two individuals needs intermediate individuals? In an attempt to consider these questions, measures of centrality are used in complex networks, focusing on the position and role of certain nodes within a network [28].

The GrAnt protocol characterizes the utility of each node as a message forwarder, by considering its centrality and its social proximity with other nodes.

#### 3.1.1 Node's Centrality

There are several ways to measure the centrality of a node in a network. The most useful centrality measures are [28]: degree centrality, betweenness centrality, and closeness centrality. The GrAnt protocol takes profit of the nodes' degree centrality and proposes a variant of the betweenness centrality, called here as *betweenness utility*.

**Degree Centrality:** The more popular a node is (i.e, the node has a high *degree centrality*), the more opportunity it will have to choose the best messages forwarders. To obtain its popularity each node will store the total number of contacts established per unit of time divided by the total number of nodes $(n-1)$ in the network. We claim that nodes with accumulated good centrality values in the past can be good forwarders in the future. Thus, in this paper, we consider the past information giving higher importance to the most recent degree centrality of nodes to predict its future degree centrality value according to the exponentially weighted moving average (EWMA) as in Eq. 1.

$$DC_i(t+\Delta t) = \alpha \times DC_i(t-\Delta t) + (1-\alpha) \times DC_i(t), \tag{1}$$

where $DC_i(t+\Delta t)$ is the predicted degree centrality (popularity) for node $i$ at time $(t+\Delta t)$. $DC_i(t-\Delta(t))$ and $DC_i(t)$ are the popularity of node $i$ at the past time $(t-\Delta t)$ and at the current time $t$, respectively. The coefficient $\alpha$ represents the degree of weighting decrease, a constant smoothing factor between 0 and 1. EWMA is used for its simplicity as the nodes' popularity needed to be determined online as new contacts are established. As future work, it will be necessary to conduct a detailed study of the degree centrality pattern of nodes over network lifetime.

**Betweenness Utility:** The traditional betweenness centrality measures evaluate the frequency that a node appears on the shortest paths linking any two other nodes. In this way, a node with a high betweenness centrality is a better candidate to facilitate interactions among the nodes it links [28]. Nevertheless, as we are interested in different paths to each destination, we compute the nodes' betweenness utility in a slightly different way. In particular, to have a high betweenness utility in relation to a destination $d$, a node $i$ must appear with high frequency in paths between any source node and $d$: $(BetwU_{i,d})$. In this way, no shortest-path verification is required and, different from related works such as [9], no list of all previous encountered contacts by a node is exchanged. In fact, in GrAnt, neighbors only exchange local information concerning their degree centrality and their betweenness utility in relation to each destination $d$.

### 3.1.2 Social Proximity between Nodes

Another important information to be inferred from DTNs nodes is their social proximity with other nodes. We use a new metric that considers the prediction of the contact duration combined with the contacts frequency (as in Eq. 2).

$$Social_{i,j} = \frac{\lambda_{i,j} \times d_{i,j}(t + \Delta t)}{T}, \tag{2}$$

where $\lambda_{i,j}$ is the contacts frequency (i.e. the number of times $i$ and $j$ established a contact over the time window $T$) and $d_{i,j}(t+\Delta t) = \beta \times d_{i,j}(t - \Delta t) + (1-\beta) \times d_{i,j}(t)$ is the predicted contact duration between nodes $i$ and $j$ for time $(t + \Delta t)$, with a weighting coefficient $\beta$ between 0 and 1. $d_{i,j}(t + \Delta t)$ presents a minimum handling of events and brings a more updated view of the contact duration between nodes.

Once characterized the social connectivity between DTN nodes, we describe the three GrAnt's modules responsible for gathering and employing such knowledge to make their decisions: (1) *Routing:* determine which route(s) a message must follow to eventually reach its destination (see Section 3.2); (2) *Scheduling:* decides the order in which messages are transmitted (see Section 3.3); (3) *Buffer Management:* indicates which messages can be discarded from the buffer when it reaches its occupancy limit (see Section 3.4).

## 3.2 Routing Module

The GrAnt routing module falls under the category of prediction-based protocols as it observes the nodes' behavioral patterns to ensure a good message delivery rate, fewer dropped messages, and a low cost in terms of the transmitted message replicas. It is composed by a *path search phase* and a *backward phase* as detailed in the next subsections.

### 3.2.1 Path Search Phase by Forward Ants

The path search phase of GrAnt involves two functions aiming to infer the best messages forwarders: a *message forwarding* and a *path's quality measuring.*

**Message Forwarding.** The GrAnt message forwarding determines which route(s) a message must follow to eventually reach its destination. The forwarding decision is performed by adopting a greedy transition rule, which takes advantage of every good contact opportunity and provides a more efficient decision about the next forwarder. The transition rules consider two metrics very popular in ACO paradigms: the pheromone ($\tau_{(i,j),d}$) at link $(i, j)$ in the path to a destination $d$ and the heuristic function ($\eta_{(x),d}$) associated to a node $x$ in the path to $d$.

An FA $k$ at a node $i$ decides whether to forward or not a message $m$ to a new contact $j$ based on three conditions that consider the nodes' utility which may change according to whether node $i$ is the source of $m$ or an intermediate node that received $m$' custody. The pseudo-code of Fig 2 describes the conditions for forwarding $m$ from node $i$ to $j$, $\forall j \in N_i$ ($N_i$ is the set of neighbors of node $i$):

1- This condition forwards $m$ to a high quality path previously found. Therefore, it will always verify if there is pheromone on the link $(i, j)$ towards a destination $d$ ($\tau_{(i,j),d}$, see Section 3.2.2) and if the utility of the new contact $j$ ($U_j$) is better than node's $i$ utility ($U_i$). If the two conditions are true, the message $m$ and its forwarder ($j$) will be stored in the tuple <*message, forwarder*>, as shown in lines 7 and 18 of Fig. 2, and the conditions 2 and 3 will not be tested. The nodes' utility, which describes how good a node can perform as a message forwarder, is calculated here in a different way at a source node and at an intermediate

```
1  Given a message m in the buffer of node i
2  for connections j
3    //τ_(i,j),d(t) is the pheromone on link (i,j) to destination (d) in time t
4    η_(x),d(t)= Social_x,d + BetwU_x,d; // Heuristic Function of a node x={i or j}
5    if node i is the source of the message m
6       if (τ_(i,j),d(t) > δ_init and Social_j,d > Social_i,d )   //cond 1
7          new Tuple<Message, Forwarder>(m, j);
8       else if  (U^m_best_fwd ≠ ϕ) and (τ_(i,j),d(t)×η_(j),d(t) > U^m_best_fwd)  //cond 2
9            U^m_best_fwd = τ_(i,j),d(t)×η_(j),d(t));
10           best_fwd = j;
11      else if (Social_j,d >  Social_i,d) //cond 3
12           U^m_best_fwd = τ_(i,j),d(t)×η_(j),d(t));
13           best_fwd = j;
14     endif
15   else // (intermediate node)
16      // τ_(i,d),d(t) is the pheromone concentration on link (i,d) in time t
17      if (τ_(i,j),d(t) > δ_init  and  η_(j),d(t) > η_(i),d(t))    //cond 1
18          new Tuple<Message, Forwarder>(m, j);
19      else if  (U^m_best_fwd ≠ ϕ) and (τ_(i,j),d(t)×η_(j),d(t)  > U^m_best_fwd)  //cond 2
20           U^m_best_fwd = τ_(i,j),d(t)×η_(j),d(t));
21           best_fwd = j;
22      else if (τ_(i,j),d(t)×η_(j),d(t)  > τ_(i,d),d(t)×η_(i),d(t))    //cond 3
23           U^m_best_fwd = τ_(i,j),d(t)×η_(j),d(t));
24           best_fwd = j;
25     endif
26   endif
27 endFor
28 if  (U^m_best_fwd ≠ ϕ)   new Tuple<Message, Forwarder>(m, best_fwd);
```

Figure 2: Pseudo-code of the Routing and Forwarding Module of GrAnt Protocol.

node. If the node $i$ is the source of $m$ it will have the highest betweenness utility (i.e., $BetwU_{i,d}$, as in Section 3.1.1) among all other nodes, thus, the utilities of $i$ and $j$ consider only their social proximity with $d$ ($Social_{i,d}$ and $Social_{j,d}$, as in Eq. 2), see line 6. However, if $i$ is an intermediate node the utilities consider the social proximity and the betweenness utility metrics. These two metrics are incorporated in an heuristic function as in Eq. 3, see line 17. Where $x$ is the node whose metrics are being analyzed;

$$\eta_{(x),d}(t) = BetwU_{x,d} + Social_{x,d}, \tag{3}$$

2- This condition tries to find out a new best forwarder for $m$ among the current contacts $j$ of node $i$, considering that a best forwarder $best\_fwd$, with its utility stored in a variable called $U^m_{best\_fwd}$, has been previously found. Thus, the variable $U^m_{best\_fwd}$ (lines 9 and 20) and the variable that represents the best current forwarder $best\_fwd$ for the message $m$ (lines 10 and 21) are updated with $U_j$ and $j$, respectively, when there is a previously value for $U^m_{best\_fwd}$ (it is not empty) and the relation $U_j > U^m_{best\_fwd}$ holds. If so, the condition 3 is not tested. It is important to point out that the variable $U^m_{best\_fwd}$ is another important contribution of GrAnt protocol: in a dynamic way, it limits the number of redundant copies of $m$. The utility of $j$ and $best\_fwd$ consider the product of the heuristic function (as in Eq. 3) and the pheromone concentration (see Section 3.2.2);

3- This condition is used for initialize/update the values of $U^m_{best\_fwd}$ and $best\_fwd$. If none of the two former conditions are satisfied, it verifies if the relation $U_j > U_i$ holds. If so, the variables $U^m_{best\_fwd}$ and $bestForwarder$ for $m$ are (re)initialized with $U_j$ and $j$, respectively. Here, the utilities of $i$ and $j$ consider: the $Social_{i,d}$ and $Social_{j,d}$ metrics if $i$ is a source node (line 11); the product of heuristic function and pheromone concentration if $i$ is an intermediate node (line 22).

After analyzing the current contacts' utility and inferring the best message's forwarder (according to the variable $bestForwarder$), GrAnt creates a tuple *<message, forwarder>* (line 28), that represents the pair of message $m$ and its designated forwarder $bestForwarder$, and updates a variable called $Forw_m$ that indicates the number of times a message $m$ was forwarded. $Forw_m$ is indexed by the message identification and it is incremented by one at each forwarding. Both variables are used by the scheduling and buffer management modules (see Sections 3.3 and 3.4).

**Path's Quality Measuring.** The path's quality measuring of GrAnt is initialized on-demand, when a message has to be delivered to its destination. Once a path search is requested in a source node, FAs are created, encapsulated into the data message and sent toward the destination via one or more intermediate nodes. The search for new paths continues until the node meets the destination or becomes aware of the successfully delivery of the corresponding message to its destination (see Fig. 3).

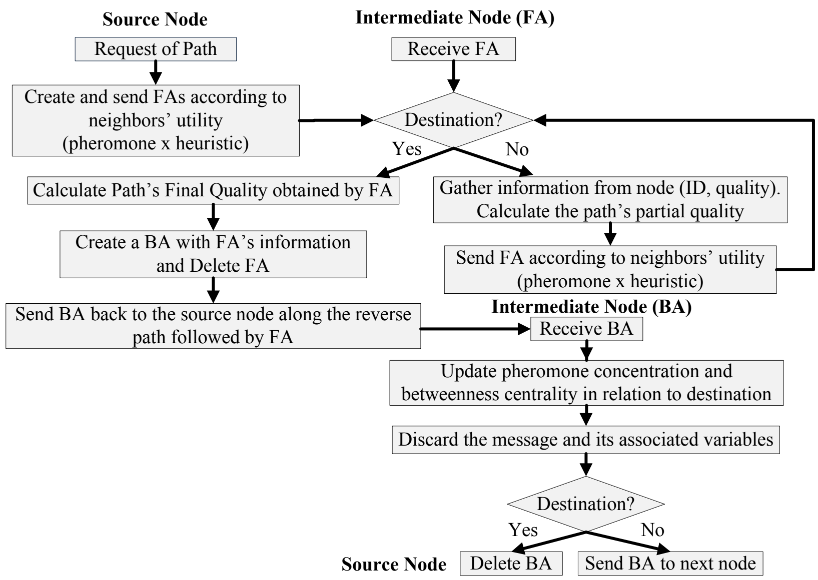



Figure 3: Path Search Phase and Backward Phase of GrAnt Protocol.

Throughout its path search, the FA carries the following information: $ID$ of the node that originated the message, $ID$ of the message destination, the $ID$ of the nodes through which the FA passed, and the predicted future quality of each of these nodes. In particular, the nodes quality will equally update the *partial quality* $q_{path^k_{s,d}}$ of

$path^k_{s,d}$, i.e. the path being constructed by the forwarding ant $k$ containing the nodes between the source $s$ and the destination $d$, as in Eq. 4.

$$q_{path^k_{s,d}}(t) = \sum_{\forall i \in path^k_{s,d}} Q_i(t + \Delta t), \tag{4}$$

Where $q_{path^k_{s,d}}(t)$ is the partial quality of the path in time $t$. In this paper the predicted future quality $Q_i(t + \Delta t)$ is the degree centrality $DC_i(t + \Delta t)$ (see Eq. 1) of a node $i$ belonging to the $path^k_{s,d}$ at time $t + \Delta t$.

When the FA $k$ reaches the destination, the *total quality* of the constructed path $Q_{path^k_{s,d}}(t)$ is calculated considering the average quality of its nodes and the reciprocal of the number of hops (i.e., $nHops$) composing it, as shown in Eq. 5. It is worth mentioning that smaller is the number of hops in a path, less network resource will be consumed and less interference will be generated.

$$Q_{path^k_{s,d}}(t) = \frac{q_{path^k_{sd}}(t)}{nHops} + \frac{1}{nHops}, \tag{5}$$

### 3.2.2 Backward Phase

After calculating the quality of the new path, a Backward Ant (BA) will be created with the information obtained by the FA and, then, the FA is deleted. The BA is sent back to the node that originated the message through the reverse path selected by the FA. Finding the complete reverse path performed by the FA may be hard or high delay constrained. Nevertheless, nodes closer to the destination will have a high probability of being visited by the BA. This is due to the high priority assigned to BA messages which indicates that they will always be forwarded first and dropped last (see Sections 3.3 and 3.4). According to the buffer management module (Section 3.4), BA messages that do not find their sources will be discarded when their TTL expires.

The reception of a BA $k$ sent from a node $j$ to each neighbor $i$ produces two effects: (1) increases by one the node $i$'s betweenness utility (i.e., $BetwU_{i,d}$) to the destination $d$ of the message (see Section 3.1.1) and (2) updates the pheromone on the link $(i, j)$ toward $d$ (i.e., $\tau_{(i,j),d}$) according to the value resulting from Eq. 6:

$$\tau_{(i,j),d}(t) = \begin{cases} (1-\rho) \times \tau_{(i,j),d}(t - \Delta t) + Q_{path^k_{s,d}}(t), & \text{if } i \in path^k_{s,d} \\ (1-\rho) \times \tau_{(i,j),d}(t - \Delta t), & \text{if } i \notin path^k_{s,d} \end{cases} \tag{6}$$

where $\tau_{(i,j),d}(t - \Delta t)$ is the pheromone of link $(i, j)$ last updated at time $(t - \Delta t)$ and $Q_{path^k_{s,d}}(t)$ is the pheromone concentration deposited by the just received BA $k$ on link $(i, j)$ (where $i, j \in path^k_{s,d}$). We consider for computation reasons, that, for the first time the pheromone to a destination $d$ is deposited in a link $(t - \Delta)$, $\tau_{(i,j),d}(t - \Delta t)$ equals to $\delta_{init}$ (see Section 4). Finally, an evaporation process (i.e., $(1-\rho)\times\tau_{(i,j),d}(t-\Delta t)$ in Eq. 6) is necessary for the ants "to forget" previous pheromone values deposited to a destination in a link. This evaporation has the effect of reducing the influence of the search history and is only triggered when a new pheromone to a destination is deposited. This event-driven evaporation process (instead of a time-driven one) is another contribution of this work.

Additionally, the BA serves as an acknowledgment of the message received by the destination, allowing the nodes which still maintain the message to discard it. A node that encounters another node that has already received a BA for a given message, will delete the corresponding message and its associated variables. Upon its reception by the source node, the BA will be destroyed. Thus, full paths are obtained for each destination using information gathered by the ants during the search phase. In next iterations, those knowledge will guide the FAs during its path to each destination.

In summary, the adopted strategies by GrAnt allow the maintenance of a set of alternative paths to each destination. At the beginning, only lower quality contacts will be available, however due to the dynamics of nodes in DTNs, after some time, a higher percentage of FAs will provide a faster discovery of new paths and/or the intensification of already existing ones.

### 3.3 Scheduling Module

Eventually, messages will be waiting at each node $i$ to be forwarded to one or more available contacts until they reach their final destinations. As the duration of contacts may be short and often insufficient to exchange messages, it is important to assign forwarding priorities to messages, according to their importance, and to define some scheduling module. In this paper, the scheduling module performs according to the messages state: (1) their number of forwarding; (2) their designated forwarders; (3) their priority class. After assigning a forwarder to each message, according to the forwarding decision described in Section 3.2.1, tuples <*message, forwarder*> are created and a classification process is triggered to determine the order of their transmission. Initially, the tuple <*message, forwarder*> will be ordered according to their messages' priority. Two priority classes are considered: (1) expedited or high priority class and (2) normal priority class. Data and FAs messages belong to the normal priority class and BAs belongs to the expedited priority class. If all tuples belong to the same priority class, the scheduling module will consider the number of times each message $m$ was sent to other nodes (i.e., according to $Forw_m$). The tuple with the message with lower $Forw_m$ value will be transmitted first. If the tuples have the same $Forw_m$, the tuple with the message whose destination are more likely to be encountered by the designed message forwarder $j$ (i.e., higher $Social_{i,j}$) will be transmitted first. Finally, if the tuples belong to the same priority class and have the same $Forw_m$ and $Social_{i,j}$ values, the oldest one will be scheduled first.

### 3.4 Buffer Management Module

The buffer management module allows GrAnt (1) to reactively indicate which messages can be discarded when the buffer reaches its capacity and a new message needs to be stored, and (2) in a preventive way, to remove old or already delivered messages from the buffer. The following actions are taken: (1) regularly it checks the Time To Live (TTL) of messages and discards those whose TTL has expired; (2) discards messages that were successfully received by the destination node; (3) when the buffer is full, the tuples with the messages with lower priority will be dropped first. If the tuples' messages belong to the same priority class, the tuple with a higher $Forw_m$ value will be dropped, because it assumes that such message has a higher probability of having been delivered to its final destination. If the tuples' messages have the same $Forw_m$ value, the one whose destination is less likely to be encountered (i.e., according to the $Social_{i,j}$ value of the forwarder in the tuple that represents the node with the current message's custody) will be dropped. Finally, if the tuples' messages belong to the same priority class and have the same $Forw_m$ and $Social_{i,j}$ values, the oldest one will be dropped.

## 4 Performance Evaluation

This section describes the simulation experiments we have conducted to assess both the performance and the accuracy of the GrAnt protocol in different mobility environments. The simulation is carried out through the Opportunistic Network Environment (ONE) Simulator [29]. We investigate how good are the metrics incorporated in the GrAnt protocol (e.g., quality of nodes and pheromone evaporation), how GrAnt performs as a forwarding protocol, and how its performance compares with other protocols. As explained in Section 2.1, we are interested in the controlled flooding-based and prediction-based protocols which can operate in various environments with different nodes mobility models. So, Epidemic and PROPHET protocols are used for performance gain comparison with GrAnt. To evaluate their reliability and cost, we use four metrics:

- *Message delivery ratio:* the ratio of packets delivery to destinations;
- *Message redundancy ratio:* reflects the number of messages replicas propagated into the network. It is expressed as $Redundancy = (M_{transmitted} - M_{delivery})/M_{delivery}$ where $M_{transmitted}$ represents the number of messages transmitted to nodes and $M_{delivery}$ is the number of messages delivered to their destination;
- *Number of dropped messages* due to a buffer overflow;
- *Average message delivery delay:* the average time between a message is generated and delivered (including buffering delays).

Finally, to better reflect realistic mobility scenarios, we consider a *community-based* and an *activity-based* scenario. The simulation parameters are summarized in Table 1. In particular, our evaluations neither assume infinite buffers nor infinite bandwidth. The considered scenarios, the parameters analysis and the performed investigations are presented in Sections 4.1, 4.2 and 4.3, respectively.

Table 1: Simulation Parameters

| Simulation Parameters | POIs Scenario | WD Scenario |
|---|---|---|
| Duration / Warm up period (sec.) | 800000 / 5000<br>800000 / 5000 | 800000 / 5000<br>800000 / 5000 |
| Number of Simulations | 10 | 10 |
| Number of Nodes | 120 (50 source nodes) | 339 (80 source nodes) |
| Time of Nodes' Quality Update (sec.) | 5000 | 5000 |
| Area (meters) | 8800 $x$ 7800 | 10000 $x$ 8000 |
| Nodes Speed (m/s) | 0.5-1.5 | 0.8-1.4 (pedestrian), 7-10 (cars and buses) |
| Wait Time at destination (sec.) | 100-200 | 100-200, 10-30 (buses dwell time in points) |
| Message TTL (min.) | 600 | 1800 |
| Rate of Message Generation (sec.) | 50-90<br>50-90 | 100-150<br>100-150 |
| Message Size (KB) | 500 | 500 |
| Nodes Buffer (MB) | 4 | 10 |
| Pheromone Threshold $\delta_{init}$ | 0.01 | 0.01 |
| $\alpha$ and $\beta$ (degree of weighting decrease) | 0.3<br>0.3 | 0.3<br>0.3 |

## 4.1 Mobility scenarios

It is important that the movement models used in evaluations be realistic. In realistic scenarios, users do not move completely random but rather move in a predictable way based on repetitive patterns of behavior such that if a node has visited a location several times before, it is likely that it will visit that location again. So, in this paper we evaluate the performance of GrAnt in two different scenarios to better reflect that reality: a *community movement model* and a *working day movement model.*

**Community Movement Model.** The community-based scenario is divided into five communities or Points of Interest (PoIs) that simulates a group of people in their community that will eventually meet each other and will exchange data. It uses the "Shortest Path Map-Based Movement" movement model [29] that employs Dijkstra's algorithm for finding the shortest path between two random PoIs. There are four groups of nodes, each one with different destination selection probabilities and with thirty nodes that are placed in a random PoI. There is a small probability of these people go to other PoIs different from their home community, but there is a great probability that they meet each other at PoIs in common. Table 2 shows the destination selection probabilities assigned to each node group, where PoI 5 represents the point of common interest to all groups.

Table 2: Destination Selection Probabilities

| Node Group | Destination POI1 | Destination POI2 | Destination POI3 | Destination POI4 | Destination POI5 |
|---|---|---|---|---|---|
| 1 | 0.6 | 0.05 | 0.05 | 0 | 0.3 |
| 2 | 0.05 | 0.6 | 0 | 0.05 | 0.3 |
| 3 | 0 | 0.05 | 0.6 | 0.05 | 0.3 |
| 4 | 0.05 | 0 | 0.05 | 0.6 | 0.3 |

**Working Day Movement Model.** The Working Day (WD) movement model [29] [30] represents an activity-based scenario that simulates the daily lives of people who go to work in the morning, spend the day working, at the end of the day may go to a public place for leisure activities with friends and at night go to their houses. The scenario is divided into meeting points, bus, houses, offices, and roads. Social relationships are formed when a group of people are doing the same activity in the same location. The nodes use the same time to awaken in the morning, leave their houses and use different means of transport (walking, bus or car) to go working. Different nodes have different locations where they go to leisure activities. So, in this scenario, differently from the destination selection probabilities of POIs scenario, each node has its own routine, always following the same sequence of activities. Eight groups were created from A to H. Three of them (i.e., E, F, G) were created to simulate movement between A and other groups, and one group (i.e., H) was created to simulate movement between all groups. The assignment of nodes per groups is the following: A has 50 nodes; B has 15 nodes; C has 30 nodes; D has 30 nodes; E (i.e., A and B) has 30 nodes; F (A and C) has 50 nodes; G (A and D) has 50 nodes; H (all groups) has 70 nodes.

For both scenarios, each time a message is generated one source node is randomly selected and it has a destination node randomly selected from 10 of its friends. In each simulation, messages are generated according to the message generation rate. All results show the average values over 10 simulation runs. More details about the simulation parameters for POIs and WD are given in Table 1.

## 4.2 GrAnt parameters analysis

Here, we investigate how GrAnt parameters affect its performance and which are the best values to be used in the simulation to asses good delivery rate. In particular, we analyzed the social proximity between nodes, the nodes' quality, and the pheromone evaporation rate. We use the POIs and WD scenarios with a buffer of 4 MB and 10 MB, respectively. Both scenarios have an interface with a communication range of 10 meters and a transmission speed of 250 Kbps.

**Social proximity metric's evaluation.** We first analyze how good our EWMA metric presented in Eq. 2 is to compute the social proximity between nodes ($Social_{i,j}$) during a time window $T$, when compared to the two metrics presented in [10]: (1) Normalized Contact Duration (NCD)= $\int_0^T \zeta_{i,j}(t) \times d_{i,j}/T$, where $\zeta_{i,j}(t) = 1$ if $i$ and $j$ are in contact and zero otherwise and $d_{i,j}$ is the encounter duration between $i$ and $j$;

(2) Inter-Encouter Time (IET) $= \int_0^T \theta_{i,j}(t) \times d_{i,j}/T$, where $\theta_{i,j}(t)$ is the time length, starting from time $t$, till the time when $i$ and $j$ encounter each other and $d_{i,j}$ is the encounter duration between them. We use the degree centrality metric to evaluate the quality of each constructed path and an evaporation rate of 0.1. The results showed that by using our social proximity metric, GrAnt delivered 62.30% (against 61.84% of NCD and 55.08% of IET) and 59.83% (60.10% in NCD and 55.67% of IET) of the messages in POIs and WD scenario, respectively. Our metric performed better in terms of message delivery in the POI scenario and in WD scenario the NCD metric delivered more messages. Considering the average message delivery rate for both scenarios, our metric was the the best one.

**Nodes Quality metric's evaluation.** Next, we evaluate the performance of the metric $Q_i(t)$ when considering the popularity of node $i$ presented in Eq 1 (see Section 3.1.1) compared to the use of a common used metric that represents the node's percentage of free buffer (i.e., a node is considered having a good quality if it has more free space in buffer). These metrics consider the values during a time ($t$) and ($t - \Delta(t)$) using the EWMA as in Eq. 1. For the two mobility scenarios, the popularity metric provided better results (62.30% of delivered messages in POIs and 59.83% in WD) than the buffer metric (61.84% in POIs and 59.77% in WD).

**Pheromone evaporation evaluation.** Finally, we obtained the following results when evaluating the pheromone evaporation rates $\rho = [0.9; 0.7; 0.3; 0.1]$. The results concerning the message delivery rate are the following: when $\rho = 0.9$, 61.96% in POIs and 59.65% in WD; when $\rho = 0.7$, 62.04% in POIs and 59.83% in WD; when $\rho = 0.5$, 62.03% in POIs and 59.98% in WD; when $\rho = 0.3$, 62.16% in POIs and 59.78% in WD; and when $\rho = 0.1$, 62.30% in POIs and 59.83% in WD. Here, one can see that in POIs scenario an evaporation rate of 0.1 achieved the best results. In a WD scenario, however, the evaporation rate of 0.5 followed by the rates of 0.1 and 0.7 provided the best results. Considering the average message delivery rate for both scenarios, the evaporation rate of 0.1 performed better as it maintains more solutions, and consequently, will be used in the remaining section.

## 4.3 GrAnt performance analysis

The performance gain of GrAnt was compared to Epidemic and PROPHET, both with a hop-count field of 11, over different buffer sizes (see Fig. 4), message's TTLs (see Fig. 5), and simulation time (see Table 3) for POIs and WD scenarios. We do not consider a hop-count field in GrAnt because the variable $U^m_{best\_fwd}$ (one of the contribution of our protocol) is able to dynamically limit the number of messages forwarding (see Section 3.2.1). Unlike GrAnt, Epidemic and PROPHET do not make use of such optimized resource. So, to be fair on the protocols comparisons we set a hop-count field for both Epidemic and PROPHET as these algorithms perform very poor without it.

Each graph of Figs. 4 and 5 contains six curves for GrAnt, Epidemic and PROPHET, two for each of them. The dash curves with empty points show the results where the nodes have one interface with a communication range of 10 meters and a transmission speed of 250 Kbps and the solid points' curves show the results where the nodes have two interfaces: one with a communication range of 10 meters and a transmission speed of 250 Kbps; and the other with a communication range of 100 meters and a transmission speed of 10 Mbps.

In Figs. 4(a), 4(b), 4(c), and 4(d), it can be seen that higher is the buffer size and communication range, more messages are delivered to their destinations and less messages redundancy is generated by the three protocols. The only exception is in the POIs scenario, see Fig. 4(c), in which the GrAnt protocol had a small increase in the redundancy ratio when the buffer size increased. Nevertheless, for all buffer sizes, for the two scenarios, and for both communication ranges, the GrAnt protocol

Table 3: Protocols' Performance Gain over Different Simulation Time

| Simulation Time (seconds) | GrAnt (Msg. Deliv./Msg. Redun.) | PROPHET (Msg. Deliv./Msg. Redun.) | Epidemic (Msg Deliv./Msg. Redun.) |
|---|---|---|---|
| 400,000 | 60.10%/7.18(POI)<br>56.04%/14.57(WD) | 33.20%/25.16(POI)<br>29.83%/107.18(WD) | 36.71%/24.02(POI)<br>23.48%/208.53(WD) |
| 800,000 | 62.30%/6.99(POI)<br>59.83%/13.81(WD) | 33.64%/25.33(POI)<br>29.84%/108.91(WD) | 37.01%/24.41(POI)<br>24.23%/209.12(WD) |
| 1200,000 | 62.94%/6.94(POI)<br>62.35%/13.53 (WD) | 33.77%/25.34 (POI)<br>30.69%/108.28 (WD) | 37.18%/24.43(POI)<br>25.12%/206.85(WD) |
| 1600,000 | 63.39%/6.93(POI)<br>64.33%/13.43(WD) | 33.90%/25.33 (POI)<br>31.22%/107.97 (WD) | 37.41%/24.38(POI)<br>25.78%/204.49(WD) |
| 2000,000 | 63.66%/6.92(POI)<br>64.35%/13.42 (WD) | 34.05%/25.29 (POI)<br>31.21%/108.39 (WD) | 37.49%/24.41(POI)<br>25.91%/205.52(WD) |

provided the best results in terms of messages delivery ratio and redundancy ratio. For example, in POIs scenario and for a buffer size of 8MB, GrAnt protocol delivered 80.12% of the messages (against 55.13% of Epidemic and 48.03% of PROPHET) with a messages redundancy of only 7.16 (16.24 of Epidemic and 18.12 of PROPHET) when using a network interface of 10 m and 92% of messages (66.98% of Epidemic and 47.23% of PROPHET) with a redundancy of only 9.20 (49.51 of Epidemic and 48.26 of PROPHET) when hosts use two interfaces (10m and 100m). In WD scenario, GrAnt delivered 57.55% of the messages with a redundancy of only 13.96 (10m) and 97.8% with 18.12 of redundancy (10 and 100m) followed by PROPHET which delivered 27.8% of the messages with 115.97 of redundancy (10m) and 79.35% of messages with 144.27 of redundancy (10m e 100m) and Epidemic which delivered 22.44% of messages (redundancy of 223.58) and 73.24% (redundancy of 322.13). PROPHET and Epidemic protocols performed worst than GrAnt in both scenarios because they do not have a process to limit dynamically the number of messages forwarding and to choose the best candidate forwarders like GrAnt does with its variable $U^m_{best\_fwd}$.

In respect to the average delay of the delivered messages, for POIs (see Fig. 4(e)) and WD scenarios (see Fig. 4(f)), PROPHET and Epidemic protocol provided a higher delay as the buffer size increased. Nevertheless, in GrAnt protocol the results were different: there was a low delay variation for both scenarios with an interface of 10m and a decrease in average delays as the buffer size increased in both scenarios with two network interfaces. In POIs and WD scenarios with a communication range of 10m GrAnt provided the lowest delays for buffer sizes of 8M to 16MB. In buffer sizes of 4MB and 6MB PROPHET presented the best results. In POIs scenarios with two network interfaces for the buffer sizes of 4MB to 10MB PROPHET provided the lowest delay and GrAnt was better for buffer of 14MB and 16MB. In WD scenario with two interfaces, Epidemic provided the lowest average delays for buffer of 4MB to 8MB and PROPHET presented the best results for buffer sizes of 10MB to 16MB. The average message delay is the only metric where GrAnt could not provide the best values for all buffer sizes and scenarios. This is justified by the fact that GrAnt instead of propagating a message to all or almost all encountered nodes, it carefully analyzes the nodes utilities and forwards a message to a new node only if it is more promising than the already established custodian node(s) for that message. In this way, by increasing the buffer size, more messages are stored and they will stay more time in the nodes' buffer.

In relation to the number of dropped messages (see Figs. 4(g) and 4(h)), when the buffer size increased, fewer messages were dropped. For all buffer sizes and for both POIs and WD scenarios, the performance of GrAnt was better than PROPHET and Epidemic. For example, in POIs scenario, for a buffer size of 8 MB, GrAnt dropped only 13% (10m) and 14% of the messages (10 and 100m) while PROPHET dropped 37% (10m) and 85% (10 and 100m), and Epidemic dropped 20% (10m) and 89% (10 and 100m). In WD scenario the number of messages dropped by PROPHET and Epidemic was higher: PROPHET dropped 81% (10m) and 97% (10 and 100m), Epidemic dropped 86% (10m) and 99% (10 and 100m) while GrAnt dropped only 26% (10m) and 16% of the messages (10 and 100m). GrAnt outperforms Epidemic and PROPHET as it limits the number of messages replication in the network in the tentative of avoiding buffers overflow and, thus, the number of messages dropped.

Table 3 shows the delivery message ratio and the messages redundancy ratio achieved by the three protocols along the simulation time in POIs and WD scenarios with a communication range of 10m. It can be seen that when the simulation time increases, better message delivery ratios with less redundancy are obtained by GrAnt in both scenarios. This is justified by the fact that as more information are gathered by GrAnt, better choice it can make among the candidate forwarders. PROPHET and Epidemic did not exhibit the same behavior; they also deliver more messages along the time but at a cost of more redundancy generated.

As can be seen in Fig. 5, the performance gain of the three protocols was not very sensible to the messages' TTL variation. In POIs scenario, GrAnt achieved a small increase in the number of delivered messages, as the custodians nodes had more time to try to encounter each message destination (see Fig. 5(a)) and a reduction in the redundancy ratio (see Fig. 5(c)) as the TTL value increased up to 2400 minutes. Epidemic and PROPHET were less susceptible than GrAnt regarding the messages' TTL increasing as they do not depends on the time to obtain more updated information about the nodes to chose among candidate message forwarders. Both protocols, in the scenario with only one interface and from TTL of 2400 minutes, maintained the same performance. In the scenario with two interfaces the protocols had their metrics stabilized with a TTL of 1800 (in PROPHET) and 1200 (in Epidemic). In WD scenario with one interface as the TTL increased the three protocols delivered more messages as can be seen in Fig. 5(b) (only Epidemic delivered fewer messages from TTL of 3000), generated less redundancy (see Fig. 5(d)) and increased the average delay in the delivery of messages (see Fig. 5(f)). Regarding the number of dropped messages, for both scenarios and communication range, as can be seen in Figs. 5(g) and 5(h), the three protocols dropped more messages as the TTL increased. In summary, for both POIs and WD scenarios, with one or two interfaces and in all TTL variations (see Fig. 5), GrAnt delivered more messages, generated less redundancy and dropped fewer messages than Epidemic and PROPHET. In POIs scenario, with one interface and TTL of 2400 minutes, for example, GrAnt delivered 65.67% of the messages (against 32.63% of PROPHET and 36.28% of Epidemic) with only 6.66 of redundancy ratio (26.07 of PROPHET and 24.85 of Epidemic) and only 52% of dropped messages (92% of PROPHET and Epidemic). The only drawback of GrAnt is its small increase in the delay of the messages delivery (19,573 seconds) as compared to Epidemic (15,705) and PROPHET (11,666) due to its process of choosing the best forwarders. In WD scenario, GrAnt delivered 65.19% of the messages (30.35% of PROPHET and 35.63% of Epidemic), generated only 13.17 of redundancy (108.31 of PROPHET and 196.28 of Epidemic), dropped only 23% of the messages (82% of PROPHET and 87% of Epidemic) and delivered the messages with an average delay of 55,239 (57,488 of PROPHET and 67,171 of Epidemic).

# 5 Conclusions and Future Works

In this paper, we proposed the prediction-based GrAnt (Greedy Ant) protocol that uses a greedy version of the ACO metaheuristic to conduct local and global searches in a highly dynamic and complex environment by analyzing and gathering information on nodes' utility. The main motivation for the use of ACO is to take advantage of the rapid adaptation of its learning framework. The GrAnt protocol includes three modules, named Routing/Forwarding, Scheduling, and Buffer Management, aiming to maximize the number of successfully delivered messages and minimize the resource usages along each path. Simulations have shown that GrAnt outperforms PROPHET and Epidemic in both activity-based and community-based scenarios. In a working day movement model, for example, GrAnt is able to achieve higher message delivery ratio (gain of 114,79% when compared to PROPHET and 82,96% when compared to Epidemic), to generate lower messages redundancy (87,84% less than PROPHET and 93,29% less than Epidemic), and to drop fewer messages (72,95% less than PROPHET and 73,56% less than Epidemic). This is due to GrAnt's capability of dynamically restricts the number of messages forwarding to the most promising nodes. By making use of useful information about the candidate messages forwarders like their degree centrality, betweenness utility and their social proximity with other nodes, GrAnt is able to better direct the network traffic.

As future work, we intend to improve the prediction process of variables like the degree centrality and contact duration with more sophisticated mechanisms. Additionally, we intend to consider the pheromone concentration updating during the path search and analyze the resulting performance of GrAnt.

# References

[1] J. F. Kurose, K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 5th Edition, Pearson Addison Wesley, 2010.

[2] F. Warthman, Delay-tolerant networks (dtns) - a tutorial, Tech. rep., Warthman Associates (2008).

[3] K. Fall, A delay-tolerant network architecture for challenged internets, in: Proc. of the Conference on Applications, technologies, architectures, and protocols for computer communications, 2003, pp. 27–34.

[4] K. Fall, S. Farrel, Dtn: An architectural retrospective, IEEE Journal on Selected Areas in Communications 26 (5) (2008) 828–836.

[5] V. C. et al., Delay-Tolerant Network Architecture, iETF RFC-4839, DTN Research Group (2007).

[6] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics,Part B: Cybernetics 26 (1) (1996) 29–41.

[7] M. Dorigo, G. di Caro, Ant Colony Optimization Meta-Heuristic, New Ideas in Optimization, McGraw-Hill, 1999.

[8] A. Lindgren, A. Doria, O. Schelén, Probabilistic routing in intermittently connected networks, Lectures Notes in Computer Science 3126 (2004) 239–254.

[9] E. M. Daly, M. Haahr, Social network analysis for routing in disconnected delay-tolerant manets, in: Proc. of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2007, pp. 32–40.

[10] T. Zhou, R. R. Choudhury, K. Chakrabarty, Diverse routing: Exploiting social behavior for routing in delay-tolerant networks, in: Proc. of the International Conference on Computational Science and Engineering, 2009, pp. 1115–1122.

[11] D. Hua, X. Du, Y. Qian, S. Yan, A dtn routing protocol based on hierarchy forwarding and cluster control, in: Proc. of the International Conference on Computational Intelligence and Security, 2009, pp. 397–401.

[12] A. Vadhat, D. Becker, Epidemic routing for partially connected ad hoc networks, Tech. Rep. TR-CS-2000-06, Duke University (2000).

[13] T. Spyropoulos, K. Psounis, C. S. Raghavendra, Spray and wait: An efficient routing scheme for intermittently connected mobile networks, in: Proc. of the ACM SIGCOMM Workshop on Delay-Tolerant Networking, 2005, pp. 252–259.

[14] E. M. Daly, M. Haahr, The challenges of disconnected delay-tolerant manets, Ad Hoc Networks 8 (2) (2010) 241–250.

[15] M. P. Wittie, K. A. Harras, K. C. Almeroth, E. M. Belding, On the implications of routing metric staleness in delay tolerant networks, Computer Communications 32 (16) (2009) 1699–1709.

[16] X. Chen, J. Shen, J. Wu, A novel information model for efficient routing protocols in delay tolerant networks, in: Proc. of IEEE International Symposium on Parallel And Distributed Processing, 2009, pp. 1–8.

[17] J. M. Pujol, A. L. T. P. Rodriguez, Fair routing in delay tolerant networks, in: Proc. of the 28th Conference on Computer Communications Society, 2009, pp. 837–845.

[18] P. Hui, J. Crowcroft, Predictability of human mobility and its impact on forwarding, in: Proc. of the Third International Conference on Communications and Networking, 2008, pp. 543–547.

[19] W. Zhao, M. Ammar, E. Zegura, Controlling the mobility of multiple data transport ferries in a delay-tolerant network, in: Proc. of the 24th Annual Joint Conference on the IEEE Computer and Communications Societies, Vol. 2, 2005, pp. 1407–1418.

[20] B. Burns, O. Brock, B. N. Levine, Mv routing and capacity building in disruption tolerant networks, in: Proc. of the 24th Annual Joint Conference on the IEEE Computer and Communications Societies, 2005, pp. 398–408.

[21] S. Johnson, Emergence: The Connected Lives of Ants, Brains, Cities and Software, Penguin Books, 2002.

[22] L. Rosati, M. Berioli, G. Reali, On ant routing algorithms in ad hoc networks with critical connectivity, Ad Hoc Networks 6 (6) (2008) 827–859.

[23] L. Gambardella, M. Dorigo, Solving symmetric and asymmetric tsps by ant colonies, in: Proc. of IEEE International Conference on Evolutionary Computation, 1996, pp. 622–627.

[24] L. Liu, G. Feng, Swarm intelligence based node-disjoint multi-path routing protocol for mobile ad hoc networks, in: Proc. of the International Conference on Information, Communications and Signal Processing, 2005, pp. 598–602.

[25] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, Cambridge University Press, 1994.

[26] D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, Journal of the American Society for Information Science and Technology 58 (7) (2007) 1019–1031.

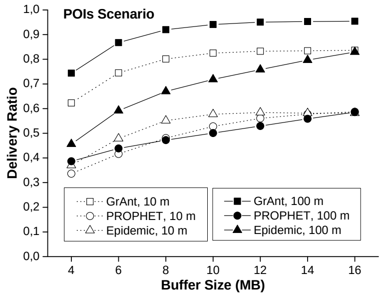
[27] I. S. Pool, M. Kochen, Contacts and influence, Social Networks 1 (1) (1978) 5–51.

[28] L. C. Freeman, Centrality in social networks: Conceptual clarification, Social Networks 1 (3) (1979) 215–239.

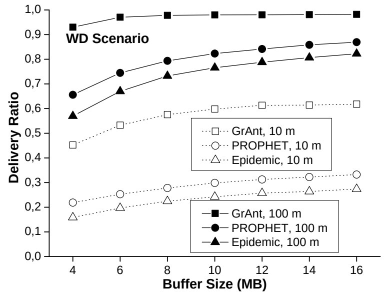
[29] A. Keränen, T. Kärkkäinen, J. Ott, Simulating mobility and dtns with the one, Journal of Communications 5 (2) (2010) 92–105.

[30] F. Ekman, A. Keränen, J. Karvo, J. Ott, Working day movement model, in: Proc. of 1st ACM/SIGMOBILE Workshop on Mobility Models for Networking Research, 2008, pp. 33–40.

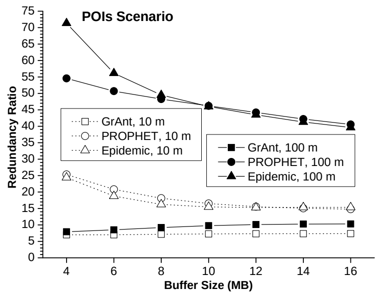
# Contents



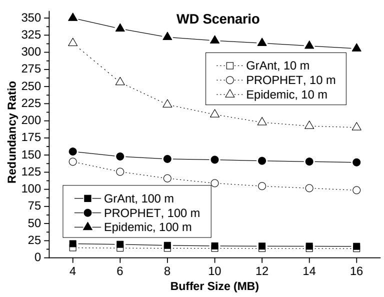

(a) Delivery - POIs scenario



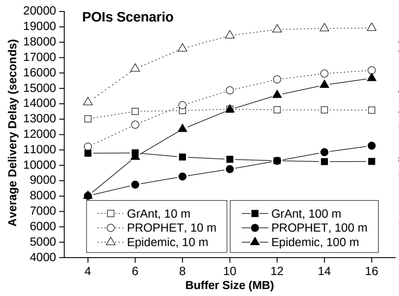

(b) Delivery - WD scenario



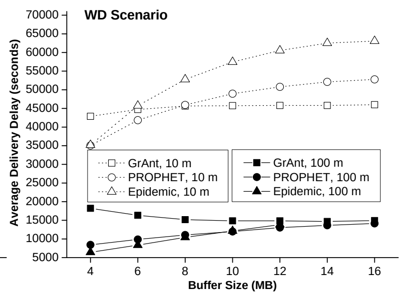

(c) Redundancy - POIs scenario



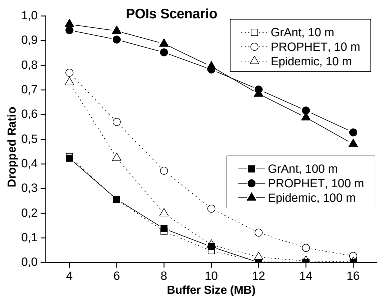

(d) Redundancy - WD scenario



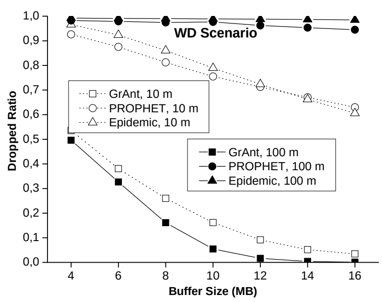

(e) Delay - POIs scenario




(f) Delay - WD scenario

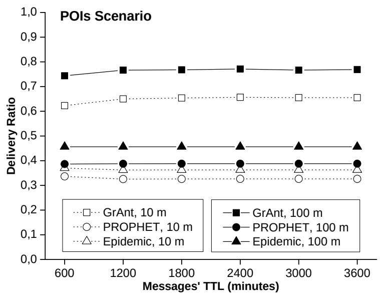



(g) Dropped - POIs scenario

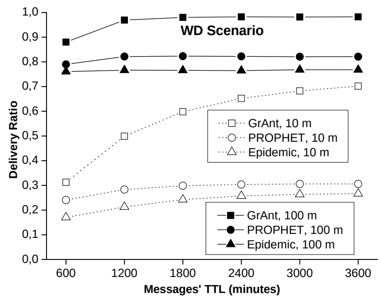



(h) Dropped - WD scenario

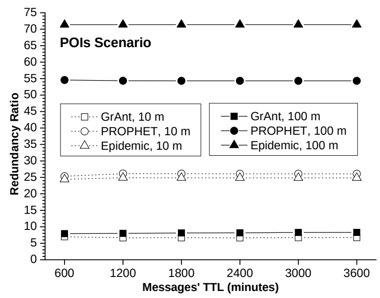
Figure 4: Protocols' Performance over different buffer sizes: Delivery Ratio in POIs (a) and WD scenario (b), Redundancy Ratio in POIs (c) and WD (d), Average Delivery Delay in POIs (d) and WD (e), Dropped Ratio in POIs (f) and WD (g)



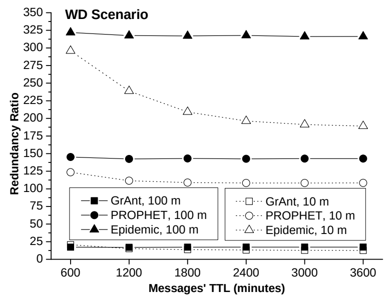

(a) Delivery - POIs scenario



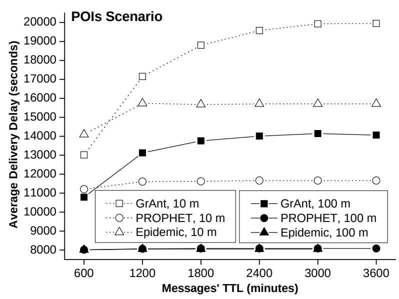

(b) Delivery - WD scenario



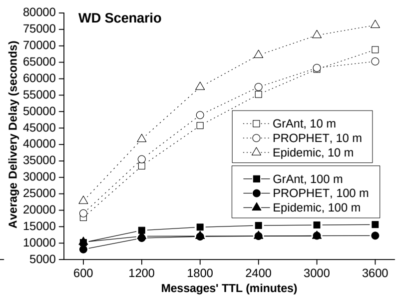

(c) Redundancy - POIs scenario



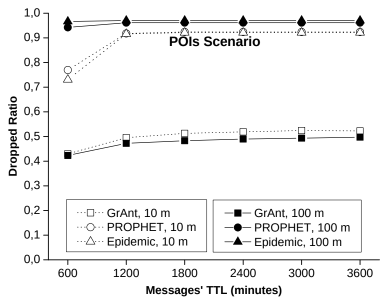

(d) Redundancy - WD scenario



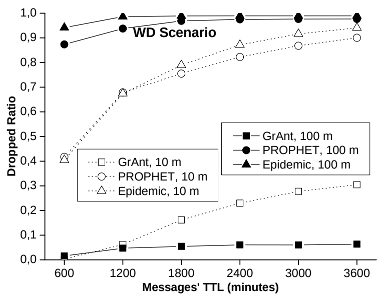

(e) Delay - POIs scenario




(f) Delay - WD scenario




(g) Dropped - POIs scenario




(h) Dropped - WD scenario



Figure 5: Protocols' Performance over different messages' TTL: Delivery Ratio in POIs (a) and WD scenario (b), Redundancy Ratio in POIs (c) and WD (d), Average Delivery Delay in POIs (d) and WD (e), Dropped Ratio in POIs (f) and WD (g)

Centre de recherche INRIA Saclay – Île-de-France
Parc Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 Orsay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399