2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing

Hovering Information - Self-Organising Information that Finds its Own Storage

Alfredo A. Villalba Castro
Centre Universitaire d'Informatique
University of Geneva, Switzerland
alfredo.villalba@cui.unige.ch

Giovanna Di Marzo Serugendo
School of Computer Science
and Information Systems
Birkbeck, University of London, UK
dimarzo@dcs.bbk.ac.uk

Dimitri Konstantas
Centre Universitaire d'Informatique
University of Geneva, Switzerland
dimitri.konstantas@cui.unige.ch

Abstract

A piece of Hovering Information is a geo-localized information residing in a highly dynamic environment such as a mobile ad hoc network. This information is attached to a geographical point, called the anchor location, and to its vicinity area, called anchor area. A piece of hovering information is responsible for keeping itself alive, available and accessible to other devices within its anchor area. Hovering information uses mechanisms such as active hopping, replication and dissemination among mobile nodes to satisfy the above requirements. It does not rely on any central server. This paper presents the hovering information concept and discusses results of simulations performed for two algorithms aiming to ensure the availability of a piece of hovering information at its anchor area.

1 Introduction

Hovering information [7] is a concept characterising self-organising information responsible to find its own storage on top of a highly dynamic set of mobile devices. The main requirement of a single piece of hovering information is to keep itself stored at some specified location, which we call the anchor location, despite the unreliability of the device on which it is stored. Whenever the mobile device, on which the hovering information is currently stored, leaves the area around the specified storage location, the information has to hop - "hover" - to another device.

Current approaches in this area (cf. Section 6) try to either define a virtual structured overlay network on top of this environment offering a stable virtual infrastructure, or propose a system-based approach offering services such as information dissemination and storage. In these approaches, the mobile nodes decide when and to whom the information is to be sent. Here we take the opposite view; it is the information that decides upon its own storage and dissemination. This opens up other possibilities, not available for traditional MANET services, such as different pieces of hovering information all moving towards the same location and (re-)constructing there a coherent larger information for a user, e.g. TV or video streaming on mobile phones.

Hovering information is a *self-organised* user-defined information which do not need a central server to exist. Individual pieces of hovering information each use local information, such as direction, position, power and storage capabilities of nearby mobile devices, in order to select the next appropriate location. Hovering information benefits from the storage space and communication capacities of the underlying mobile devices. It is not residing in a centralized server, and is not bound to any mobile operator.

This paper presents the hovering information concept as well as a preliminary algorithm allowing single pieces of hovering information to get attracted to their respective anchor locations. A complete formal description of the hovering information model is described in [6].

Section 2 discusses potential applications of this concept. Section 3 presents the hovering information concept. Section 4 discusses the Attractor Point algorithm that we have designed where the information is "attracted" by the anchor location and a general Broadcast-based algorithm we implemented in order to allow comparisons. Section 5 reports on simulation results related to availability and additional metrics such as number of messages exchanged or pieces of hovering information replicated. Finally Section 6 compares our approach to related works, and Section 7 discusses some future works.

978-0-7695-3158-8/08 $25.00 © 2008 IEEE
DOI 10.1109/SUTC.2008.62

2 Applications

When deployed over mobile devices, hovering information is an infrastructure free service that supports a large range of applications. Among others we can cite: *urban security* - users (citizens, policemen, security) post and retrieve comments or warnings related to dangers in their urban environment; *self-generative art* - users of a learning art experience centre provide collective inputs self-assembled together into a piece of art (painting, music, etc) generated by a computer according to some rules; *intravehicular networks* - drivers insert tags into the environment related to road conditions or accidents; *emergency scenarios* - emergency crew use hovering information to locate survivors or coordinate their work. More generally, hovering information is a technical way to support *stigmergy-based applications.* Stigmergy is an indirect communication mechanism among individual components of a self-organising system. Communication occurs through modification brought to local environment. The use of ant pheromone is a well known example of stigmergy. Users that communicate by placing hovering information at a geo-referenced position, which is later on retrieved by other users is also an example of stigmergy. The hovering information concept, using an infrastructure free storage media, naturally supports stigmergy-based applications that need to be deployed on an ad hoc manner (e.g. unmanned vehicles or robots).

3 Hovering Information Concept

3.1 Mobile Nodes and Hovering Information

Mobile nodes represent the storage and motion media exploited by pieces of hovering information. A *mobile node* n is defined as a tuple:

$$n = (id, loc, speed, dir, r_{comm}),$$

where id is its mobile node identifier, loc is its current location (a geographic location), $speed$ is its current speed in m/s, dir is its current direction of movement (a geographic vector) and r_{comm} is its wireless communication range in meters.

A piece of hovering information is a piece of data whose main goal is to remain stored in an area centred at a specific location called the *anchor location*, and having a radius called the *anchor radius*. A *piece of hovering information* h is defined as a tuple:

$$h = (id, a, r, n, data, policies, size),$$

where id is its hovering information identifier, a is its anchor location (geographic coordinate), r is its anchor radius in meters, n is the mobile node where h is currently hosted (hosting node), $data$ is the data carried by h, $policies$ are the hovering policies of h and $size$ is the size of h in bytes. Policies stand for hovering policies stating how and when a piece of hovering information has to hover.

We consider that identifiers of pieces of hovering information are unique, but replicas (carrying same data and anchor information) are allowed on *different* mobile nodes.

We also consider that there is only one instance of a hovering information in a given node n, any other replica resides in another node.

Figure 1 shows a piece of hovering information (blue hexagon) and two mobile nodes (yellow circles). One of them hosts the hovering information whose anchor location, radius and area are also represented (blue circle). The anchor area is the disc whose center is the anchor location, and radius is the anchor radius. The communication range of the second mobile node is also shown.

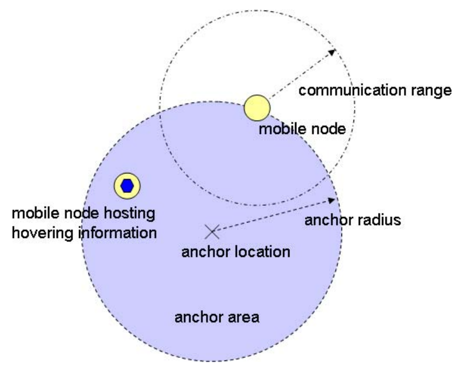

Figure 1. Mobile Nodes and Hovering Information

A hovering information system is composed of mobile nodes and pieces of hovering information. A hovering information system at time t is a snapshot (at time t) of the status of the system. We denote by $\mathcal{N}_t$ the set of mobile nodes at time t. Mobile nodes can change location, new mobile nodes can join the system and others can leave. New pieces of hovering information can appear (with new identifiers), replicas may appear or disappear (same identifiers but located on other nodes), hovering information may disappear or change node.

Figure 2 shows two different pieces of hovering information h_1 (blue) and h_2 (green), having each a different anchor location and area. Three replicas of h_1 are currently located in the anchor area (in three different mobile nodes n_2, n_3 and n_4), while two replicas of h_2 are present in the anchor area of h_2 (in nodes n_2 and n_5). It may happen that a mobile node hosts replicas of different pieces of hovering information, as it is the case in the figure for the mobile node

n_2 that is at the intersection of the two anchor areas. The arrows here also represent the communication range possibilities among the nodes.

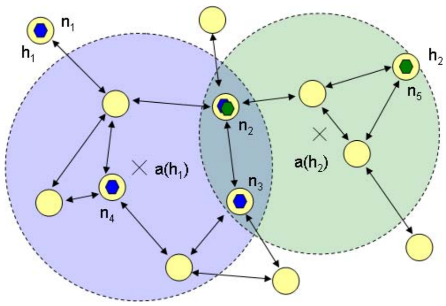

Figure 2. Hovering Information System at time t

3.2 Properties - Requirements

Survivability. A hovering information h is alive at some time t if there is at least one node hosting a replica of this information. The survivability along a period of time is defined as the ratio between the amount of time during which the hovering information has been alive and the overall duration of the observation. The survivability of h between time t_c (creation time of a piece of hovering information) and time t is given by:

$$SV_H(h,t) = \frac{1}{t - t_c} \sum_{\tau=t_c}^{t} sv_H(h,\tau),$$

where $sv_H(h,\tau)$ takes value 0 or 1 whether h is survival or not at time τ.

Availability. A hovering information h is available at some time t if there is at least a node in its anchor area hosting a replica of this information. The availability of a piece of hovering information along a period of time is defined as the rate between the amount of time along which this information has been available during this period and the overall time. The availability of h between time t_c (creation time of a piece of hovering information) and time t is given by:

$$AV_H(h,t) = \frac{1}{t - t_c} \sum_{\tau=t_c}^{t} av_H(h,\tau),$$

where $av_H(h,\tau)$ takes value 0 or 1 whether h is available or not at time τ.

Accessibility. A hovering information is accessible by a node n at some time t if the node is able to get this information. In other words, if it exists a node m being in the communication range of the interested node n and which contains a replica of the piece of hovering information. The accessibility of a piece of hovering information h is the rate between the area covered by the hovering information's replicas and its anchor area. The accessibility of h between time t_c (creation time of a piece of hovering information) and time t is given by:

$$AC_H(h,t) = \frac{1}{t - t_c} \sum_{\tau=t_c}^{t} ac_H(h,\tau),$$

where $ac_H(h,\tau)$ is the rate between the area covered by the hovering informations replicas and its anchor area. The interested reader can refer to [6] for a full set of definitions.

Let us notice that an available piece of hovering information is not necessarily accessible and vice-versa, an accessible piece of hovering information is not necessary available. Figure 3 shows different cases of survivability, availability and accessibility. In Figure 3(a), hovering information h (blue) is not available, since it is not physically present in the anchor area, however it is survival as there is a node hosting it. In Figure 3(b), hovering information h is now available as it is within its anchor area, however it is not accessible from node n_1 because of the scope of the communication range. Finally, in Figure 3(c), hovering information h is survival, available and accessible from node n_1.

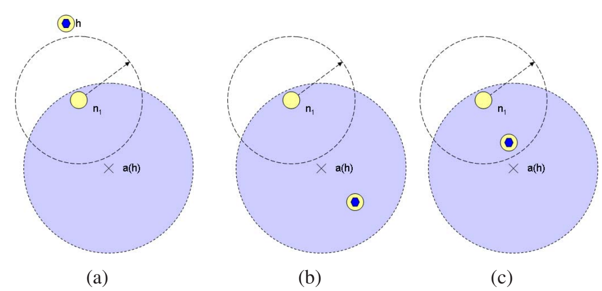

Figure 3. Survivability, Availability and Accessibility

4 Algorithms for Hovering Information

4.1 Assumptions

We make the following assumptions in order to keep the problem simple while focusing on measuring availability and resources consumption. **Unlimited memory:** All mobile nodes have an unlimited amount of memory able to store any number of hovering information replicas. The proposed algorithms do not take into account remaining memory space or the size of the hovering information. **Unlimited energy:** All mobile nodes have an unlimited amount of energy. The proposed algorithms do not consider failure of nodes or impossibility of sending messages because of low level of energy. **Instantaneous processing:** Processing time of the algorithms in a mobile node is zero. We do not consider performance problems related to overloaded processors or execution time. **In-built geo-localization service:** Mobile nodes have an in-built geo-localization service such as GPS which provides the current position. We assume that this information is available to pieces of hovering information. **Neighbours discovering service:** Mobile nodes are able to get a list of their current neighbouring nodes at any time. This list contains the position, speed, and direction of the nodes. As for the other two services, this information is available to pieces of hovering information.

4.2 Safe, Risk and Relevant Areas

In this paper we consider that all pieces of hovering information have the same hovering policies: active replication and hovering in order to stay in the anchor area (for availability and accessibility reasons), hovering and caching when too far from the anchor area (survivability), and cleaning when too far from the anchor area to be meaningful (i.e. disappearance). The decision on whether to replicate itself or to hover depends on the current position of the mobile node in which the hovering information is currently stored. Therefore, we distinguish three different areas: safe area, risk area and relevant area.

A piece of hovering information located in the *safe area* can safely stay in the current mobile node, provided the conditions on the node permit this: power, memory, etc. This area is defined as the disc having as centre the anchor location and as radius the safe radius (r_{safe}).

A piece of hovering information located in the *risk area* should actively seek a new location on a mobile node going into the direction of the safe area. It is in this area that the hovering information actively replicates itself in order to survive and stay available in the vicinity of the anchor location. This area is defined as the ring having as centre the anchor location and bound by the safe and risk radii (r_{risk}).

The *relevant area* limits the scope of survivability of a piece of hovering information. This area is defined as the disc whose centre is the anchor location and whose radius is the relevant radius (r_{rele}).

The *irrelevant area* is all the area outside the relevant area. A piece of hovering information located in the irrelevant area can disappear; it is relieved from survivability goals.

Figure 4 below depicts the different types of radii and areas discussed above centred at a specific anchor location a. The smallest disc represents the safe area, the blue area is the anchor area, the ring limited by the risk radius and the safe radius is the risk area, and finally the larger disc is the relevant area.

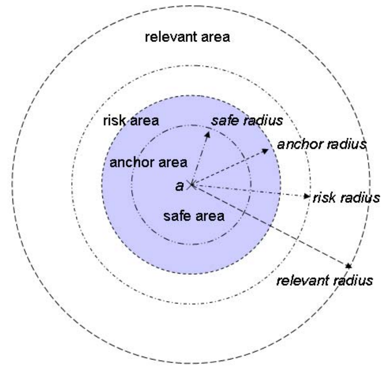

Figure 4. Radii and Areas

The values of these different radii are different for each piece of hovering information and are typically stored in the Policies field of the hovering information. In the following algorithms we consider that all pieces of hovering information have the same relevant, risk and safe radius.

4.3 Replication

We describe two algorithms simulating two variants of replication policies: the Attractor Point and Broadcast-based algorithms. Both algorithms are triggered periodically each T_R (replication time) seconds and only replicas of h being in the risk area are replicated onto some neighbouring nodes (nodes in communication range) which are selected according to the replication algorithm.

4.3.1 Attractor Point Algorithm

The anchor location of a piece of hovering information acts constantly as an attractor point to that piece of hovering information and to all its replicas. Replicas tend to stay as close as possible to their anchor area by replicating from one mobile node to the other.

Periodically and for each mobile node (see Algorithm 1), the position of the mobile node (line 2) is retrieved together with the list and position of all mobile nodes in communication range (lines 3 and 4). Hovering information replicas verify wether they are in the risk area and need to be replicated (line 8). The number of target nodes composing the multicast group is defined by the constant k_R (replication factor). The distance between each mobile node in range and the anchor location is computed (line 9). The k_R mobile nodes with the shortest distance are chosen as the target nodes for the multicast (lines 10). A piece of hovering information in the risk area then multicasts itself to the k_R

Algorithm 1 Attractor Point Replication Algorithm

1: **procedure** REPLICATION
2:     $pos \leftarrow$ NODE-POSITION
3:     $N \leftarrow$ NODE-NEIGHBOURS
4:     $P \leftarrow$ NEIGHBOURS-POSITION(N)
5:     **for all** $replica \in REPLICAS$ **do**
6:         $anchor \leftarrow$ ANCHOR-LOCATION$(replica)$
7:         $dist \leftarrow$ DISTANCE$(pos, anchor)$
8:         **if** $(dist \geq r_{safe})$ and $(dist \leq r_{risk})$ **then**
9:             $D \leftarrow$ DISTANCE$(P, anchor)$
10:             $M \leftarrow$ SELECT-KR-CLOSESTS(N, D, k_R)
11:             MULTICAST$(replica, M)$
12:         **end if**
13:     **end for**
14: **end procedure**

mobile nodes, in communication range, closest to its anchor location (line 11). Figure 5 illustrates the behaviour of the Attractor Point algorithm. Consider a piece of hovering information h in the risk area. It replicates itself onto the nodes in communication range that are the closest to its anchor location. For a replication factor $k_R = 2$, nodes n_2 and n_3 receive a replica, while all the other nodes in range do not receive any replica.

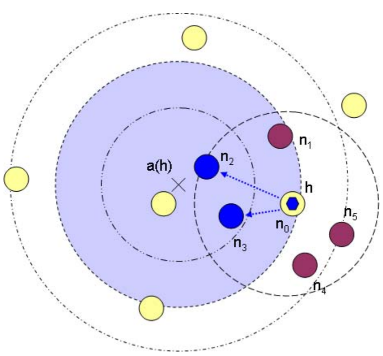

Figure 5. Attractor Point Algorithm

4.3.2 Broadcast-based Algorithm

The Broadcast-based algorithm (see Algorithm 2) is triggered periodically (each T_R) for each mobile node. After checking the position of the mobile node (line 2); pieces of hovering information located in the risk area (line 6) are replicated and broadcasted onto all the nodes in communication range (line 7). We expect this algorithm to have the best performance in terms of availability but the worst in terms of network and memory resources consumption. Figure 6 illustrates the behaviour of the Broadcast-based algorithm. Consider the piece of hovering information h in the risk area, it replicates itself onto all the nodes in communication range, nodes n_1 to n_5 (blue nodes).

Algorithm 2 Broadcast-based Replication Algorithm

1: **procedure** REPLICATION
2:     $pos \leftarrow$ NODE-POSITION
3:     **for all** $replica \in REPLICAS$ **do**
4:         $anchor \leftarrow$ ANCHOR-LOCATION$(replica)$
5:         $dist \leftarrow$ DISTANCE$(pos, anchor)$
6:         **if** $(dist \geq r_{safe})$ and $(dist \leq r_{risk})$ **then**
7:             BROADCAST$(replica)$
8:         **end if**
9:     **end for**
10: **end procedure**

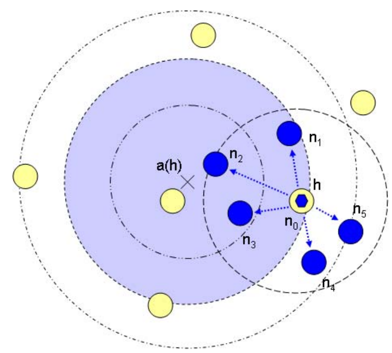

Figure 6. Broadcast-based Algorithm

4.4 Caching and Cleaning Modules

Each node is assumed to have an unlimited amount of memory. Therefore, when replicas are sent from one node to another, they are simply stored in the nodes memory. However, if a node receives two or more replicas of the *same* piece of hovering information h, the first replica to arrive is stored in the memory, and any subsequent one is ignored. Therefore, at most one replica of each piece of hovering information is present in a given node n.

Periodically - each T_C seconds - and for each node, replicas that are too far from their anchor location are removed, i.e. those replicas that are in the irrelevant area. Although the amount of memory is unlimited and replicas could stay forever in the nodes' memory, we remove the replicas that are too far away from their anchor location, this represents the cases where the replica considers itself too far from the anchor area and not able to come back anymore. This avoids as well the situation where all nodes have a replica.

4.5 Metrics

In order to evaluate and compare the above algorithms, the following values have been measured.

Messages complexity. The message complexity at a given time t is the number of messages sent between time 0 and time t by all nodes n of the system ($\mathcal{N}_t$):

$$MSGS(t) = \sum_{\tau=0}^{t} \sum_{n \in \mathcal{N}_\tau} msgs_n(\tau),$$

where $msgs_n(\tau)$ represents the number of messages sent at time τ by node n.

Replication complexity. The replication complexity measures, for a given piece of hovering information h, the maximum number of replicas having existed in the whole system at the same time.

$$REP_h(t) = \max_{\tau=t_c}^{t}\left(\sum_{n \in \mathcal{N}_\tau} mem_n(\tau)\right),$$

where $mem_n(\tau)$ is 0 if there is no replica of h in n, and 1 if there is a replica.

Concentration. The concentration of a given piece of hovering information h is defined as the rate between the number of replicas of h present in the anchor area and the total number of replicas of this hovering information in the whole environment.

5 Evaluation

We evaluated the behaviour of the two above described algorithms under different scenarios by varying the number of nodes. In these experiments, we considered only one piece of hovering information. For this given piece of hovering information h, we measured the availability of h, the corresponding message complexity, the corresponding replication complexity and the concentration of h.

We performed simulations using the OMNet++ network simulator (distribution 3.3) and its Mobility Framework 2.0p2 (mobility module) to simulate nodes having a simplified WiFi-enabled communication interfaces (not dealing with channel interferences).

5.1 Simulation Settings and Scenarios

The generic scenario consists of a surface of 500m x 500m with mobile nodes moving around following a Random Way Point mobility model with a speed varying from 1m/s to 10m/s without pause time. In this kind of mobility model, a node moves along a straight line with speed and direction changing randomly at some random time intervals. Table 1 summarises the values used for the generic scenario.

Blackboard	500mx500m
Mobility Model	Random Way Point
Nodes speed	1m/s to 10 m/s
Communication range (r_{comm})	121m
Replication time (T_R)	10s
Cleaning time (T_C)	60s
Replication factor (k_R)	1, 2, 4 and 8
Anchor radius (r)	50m
Safe radius (r_{safe})	30m
Risk radius (r_{risk})	70m
Relevant radius (r_{rele})	200m

Table 1. Simulation settings

Based on this generic scenario, we defined 10 specific scenarios with varying number of nodes: from 20 to 200 nodes, increasing the number of nodes by 20 each time. We have performed 20 runs for each scenario. One run lasts 3'600 simulated seconds. All the results presented here are the average of the 20 runs for each scenario, and the errors bars represent a 95% confidence interval. We investigated four variants of the Attractor Point algorithm, with four different replication factors, namely 1, 2, 4 and 8. All the simulations ran on a linux cluster of 32 computation nodes (Sun V60x dual Intel Xeon 2.8GHz, 2Gb RAM).

5.2 Results

Availability. Figure 7 shows the average of the availability performance over the 20 runs. As expected, the Broadcast-based algorithm outperforms the Attractor Point algorithm which tends to behave like the first one as the replication factor (k_R) increases, since the Broadcast-based algorithm is a particular case of the Attractor Point algorithm when k_R is big enough. For the Broadcast-based and the Attractor Point (with a k_R greater than 4) algorithms, we observe that an 80% of availability can be expected as soon as the number of mobile nodes in the environment reaches 100 nodes. This represents a density of 3.1 nodes per anchor area. The maximum availability value, nearly 96%, is reached by the Broadcast-based and the Attractor Point (with a k_R of 8) algorithms when the population of mobile nodes is 200, while the Attractor Point (with a k_R of 4) reaches nearly 93% of availability for 200 nodes

Messages Complexity. Figure 8 shows the average number of messages sent. As expected, the Broadcast-based algorithm sends a higher number of messages when compared to the Attractor Point algorithm. This phenomenon is amplified when the number of nodes increases. In the worst case (200 nodes), the number of sent messages, in average, by the Broadcast-based algorithm is nine times higher than the number of messages sent when the Attractor Point algorithm is used with a k_R of 1. This messages complexity

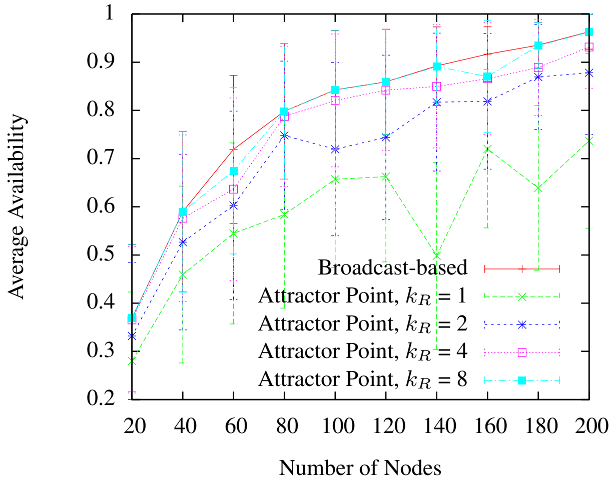

Figure 7. Availability

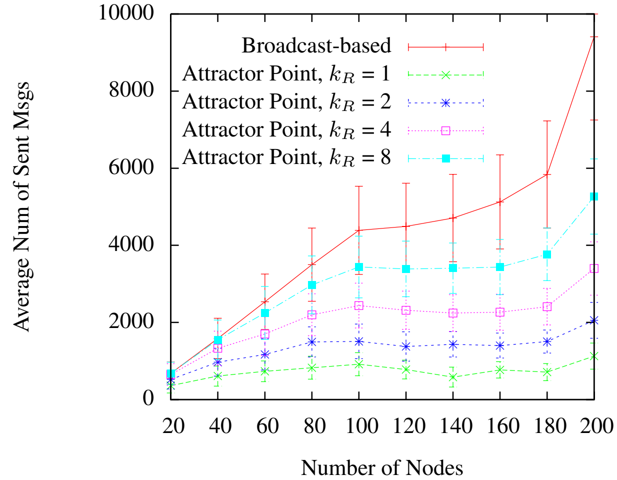

Figure 8. Messages Complexity

will be a decisive factor when applying the algorithm in a network dealing with interferences.

Replication Complexity. Figure 9 shows in average the maximum number of replicas of a single piece of hovering information having existed at the same time. Again, we observe that the Broadcast-based algorithm creates more replicas than the Attractor Point. The curves for Replication Complexity are very similar to those for Message Complexity (see Figure 8). This is explained as the number of sent messages is directly proportional to the number of existing replicas; since each replica can potentially send messages (replicate itself again).

Concentration. Figure 10 shows the concentration rate. We observe that the concentration rate is above 7% for the Attractor Point algorithm and increases with number of nodes up to 17% (depending on k_R). In the other hand, a maximal concentration rate of 7% is reached by the Broadcast-based algorithm. The Attractor Point algorithm concentrates 2 to 3 times more replicas than the Broadcast-based algorithm (depending on k_R and the number of nodes considered).

At the time of writing, additional simulations are running. They are aiming at computing the **accessibility** of hovering information under different scenarios.

6 Related Works

The Virtual Infrastructure project [2, 3] defines virtual (fixed) nodes implemented on top of a MANET. This project proposes the notion of an *atomic memory cells*, implemented on top of a MANET, which ensure their persistency by replicating their state in neighbouring mobile devices. This notion has been extended to the idea of *virtual mobile nodes* which are state machines having a fixed location or a well-defined trajectory. On top of this virtual infrastructure it should become easier to define distributed algorithms such as routing or leader election.

GeOpps [4] proposes a geographical opportunistic routing algorithm over VANETs (Vehicular Ad Hoc Networks). The algorithm selects appropriate cars for routing some information from a point A to a point B. The choice of the next hop (i.e. the next car) is based on the distance between that cars trajectory and the final destination of the information to route. This work focuses on routing information to some geographical location; it does not consider the issue of keeping this information alive at the destination, while this is the main characteristic of hovering information.

The work proposed by [5] aims to disseminate traffic information in a network composed by infostations and cars. The system follows the publish/subscribe paradigm. Once a publisher creates some information, a replica is created and propagated all around where the information is relevant. While the idea is quite similar to that of hovering information, keeping information alive in its relevant area, this study does not consider the problem of having a limited amount of memory to be shared by many pieces of information or the problem of fragmentation of information. It also takes the view of the cars as the main active entities, and not the opposite view, where it is the information that decides where to go.

The Ad-Loc project [1] proposes an annotation location-aware infrastructure-free system. Notes stick to an area of relevance which can grow depending on the location of interested nodes. Information is periodically broadcasted to neighbouring nodes. Nodes are the active entities exchanging information. The size of the area of relevance grows as

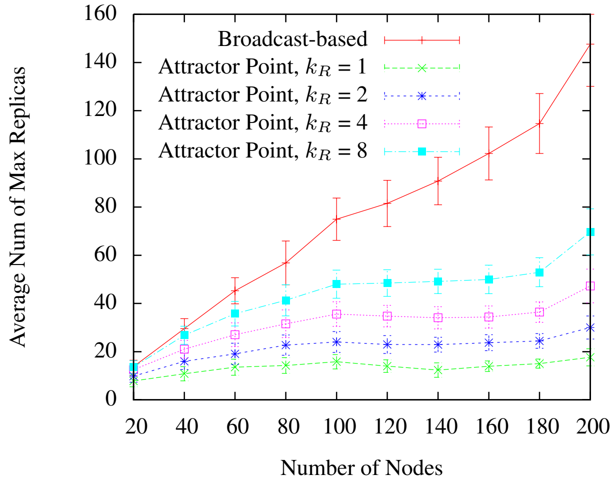

Figure 9. Replication Complexity

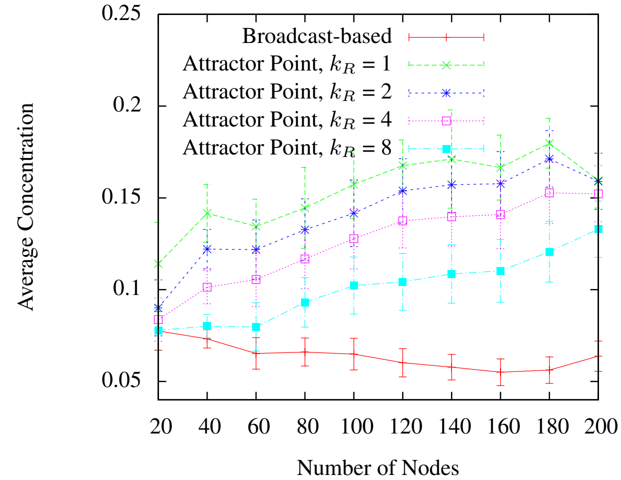

Figure 10. Concentration

necessary in order to accommodate the needs of users potentially far from the central location. The information then becomes eventually available everywhere.

7 Conclusion

In this paper we discussed the notion of hovering information, defined and simulated the Attractor Point algorithm which intends to keep the information alive and available in its anchor area. This algorithm multicasts hovering information replicas to the nodes that are closer to the anchor location. The performances of this algorithm have been compared to those of a Broadcast-based version. The results show that the Broadcast-based algorithm outperforms the Attractor Point algorithm in terms of availability but only from a very small factor. The proposed Attractor Point algorithm is much less bandwidth and memory greedy than the Broadcast-based algorithm and achieves higher levels of concentration of data in the anchor area.

Considering that these results constitute a proof of concept of the hovering information paradigm, future works will concentrate on releasing the assumption of unlimited memory and in considering not only one piece of hovering information but multiple distinct pieces all hovering in the same environment. We intend as well to take into account the speed and direction of the nodes when choosing the nodes that will host replicas. We have tested the Attractor Point algorithm under a Random Way Point mobility model and under ideal wireless conditions. This is not characteristic of real world behaviour. We will apply the Attractor Point algorithm to scenarios following real mobility patterns (e.g. crowd mobility patterns in a shopping mall or traffic mobility patterns in a city) with real wireless conditions (e.g. channel interferences or physical obstacles).

References

[1] D. J. Corbet and D. Cutting. Ad loc: Location-based infrastructure-free annotation. In *ICMU 2006*, London, England, Oct. 2006.

[2] S. Dolev, S. Gilbert, L. Lahiani, N. A. Lynch, and T. Nolte. Timed virtual stationary automata for mobile networks. In *OPODIS*, pages 130–145, 2005.

[3] S. Dolev, S. Gilbert, E. Schiller, A. A. Shvartsman, and J. Welch. Autonomous virtual mobile nodes. In *DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pages 62–69, New York, NY, USA, 2005. ACM Press.

[4] I. Leontiadis and C. Mascolo. Geopps: Opportunistic geographical routing for vehicular networks. In *Proceedings of the IEEE Workshop on Autonomic and Opportunistic Communications. (Colocated with WOWMOM07)*, Helsinki, Finland, June 2007. IEEE Press.

[5] I. Leontiadis and C. Mascolo. Opportunistic spatio-temporal dissemination system for vehicular networks. In *MobiOpp '07: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 39–46, New York, NY, USA, 2007. ACM Press.

[6] A. Villalba Castro, G. Di Marzo Serugendo, and D. Konstantas. Hovering information - self-organising information that finds its own storage. Technical Report BBKCS-07-07, School of Computer Science and Information Systems, Birkbeck, University of London, Nov 2007.

[7] A. Villalba Castro and D. Konstantas. Towards hovering information. In *Proceedings of the First European Conference on Smart Sensing and Context (EuroSSC 2006)*, pages 161–166, 2006.