Analysis of the Impact of Hello Protocol Parameters over a Wireless Network Self-Organization

Tahiry Razafindralambo
INRIA ARES / CITI INSA Lyon
Bt L. De Vinci - 21 av. Jean Capelle
69621 Villeurbanne - FRANCE
tahiry.razafindralambo@insa-lyon.fr

Nathalie Mitton
INRIA - LIFL/CNRS - USTL POPS
Parc de la haute borne - 50, avenue Halley
59650 Villeneuve d'Ascq - FRANCE
nathalie.mitton@inria.fr

ABSTRACT

The neighborhood discovery and its maintenance are very important in wireless networks for any applications, especially for routing and every self-$*$ algorithm. Neighbor nodes are usually discovered thanks to the use of the HELLO protocol. This makes this HELLO protocol very important for wireless networks especially for self-organizing the network. Most of layer-3 protocols assume an ideal MAC layer. In such a case, HELLO protocol parameters have no impact over the self-organization. But this is not the case when considering realistic MAC and physical layers. In this paper, we investigate the impact of the parameters of such a protocol over a self-organization structure when considering realistic a MAC layer. We analyze theoretically and by simulations, the joint effect of the HELLO protocol parameters and of the MAC layer characteristics over several network self-organizations.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Wireless communication

General Terms

Algorithms

Keywords

Clustering, MAC, Algorithms, Analysis, Simulations, Wireless Networks

1. INTRODUCTION

Ad hoc networks or wireless sensor networks (wireless multihop networks) are composed of devices that communicate via wireless interfaces. They require no fixed infrastructure and no human intervention. Many layer-3 protocols have been proposed in the literature for wireless ad hoc and sensor networks. They range from routing protocols such [17, 8] to clustering protocols [10, 15, 12] by going through localization protocols [20]. All that protocols assume that nodes have and maintain exact knowledge about specially nearby network nodes (called their neighbors). This knowledge is acquired thanks to small beacon messages (the well-known HELLO messages). The principle is rather simple and may be described as follows. Each network node regularly sends such a HELLO message to advertise its presence. Due to the broadcast nature of radio communications, each sufficiently close host receives this message and may infer that it is a neighbor of the sender. All nodes maintain a neighborhood table, and any localized protocol may make decisions based on this table. Self-organization often relies on a specific partition of the network, called *clustering*: the terminals are gathered into clusters according to some criteria, each cluster is identified by a special node called *cluster-head*. Many clustering algorithms have been proposed in the literature. Most of them are based on a metric which allows each node to elect its cluster-head. This metric can be for instance the node identifier [9, 1], the node degree [10, 6, 15], a mobility value [2], a balanced sum of all of them [4] or a link density value [12]. Among all these clustering heuristics, oldest ones [9, 10, 2, 4] build overlapping 1-clusters, *e.g.* where every node is at most 1 hop away from its cluster-head. However, overlapping 1-clusters offer a structure very weak towards mobility and link instability. Later, protocols building k-clusters have been proposed. Some of them just extend a protocol building 1-clusters like in [6]. Therefore, clusters are still overlapping and remain unstable. Most of these protocols constrain clusters in terms of radius [1, 6] or number of nodes [16]. As far as we know, only two of them let clusters adapt the underlying topology [15, 12] and produce unbounded clusters. This makes them more robust towards the topology changes.

Nevertheless, to the best of our knowledge, none of these layer-3 protocols, either for routing or clustering, have taken into account the importance of the reliability of these neighborhood tables on which they all rely. Indeed, even in a static environment, some HELLO packets may be lost because of collisions and delay occurring at the MAC layer level. Therefore, the choice of the MAC layer, the HELLO packet frequency and the time data are kept in memory have importance in order to provide the nodes with a view of their neighborhood as close to the physical one as possible. So, neighborhood tables are not always reliable and since layer-3 protocols mainly rely on them, routing or *clustering* results are also impacted.

In this paper, we are interested in investigating the impact of realistic MAC and physical layers over a network self-organization. We analyze how the parameters of the HELLO protocol, the choice of the MAC layer protocol and the metric used for clustering may overcome or worsen the effects of the propagation errors. To do so, we study two clustering algorithms proved self-stabilizing, which only differ by the metric used: DDR [15] and the density-based

MSWiM'07, October 22 - 26, 2007 - Chania, Crete Island, Greece

algorithm [12]. We study their behavior over two different MAC layer approaches: IEEE 802.11[1] and MadMac [18]. We provide a theoretical analysis based on stochastic geometry and queue theory to establish the more suitable parameters for the HELLO protocol (HELLO message frequency and data lifetime). We show that these parameters can strongly impact the behavior of the clustering protocols. We highlight that in a network without data communications, changing the MAC protocol does not affect the chosen parameters of the HELLO protocol, contrarily to the metric used for self-organizing the network, which can help to smooth down the link failures occurring at the MAC layer.

The remaining of the paper is organized as follows. Section 2 briefly details the MAC protocols and the clustering algorithms we use. Section 3 provides a theoretical analysis of the HELLO parameters. Section 4 describes our simulations and shows the results. At last, Section 5 concludes by giving future works.

2. BACKGROUND

2.1 Models and Notations

For the sake of simplicity, let's first introduce some notations. We classically model a multi-hop wireless network, by a graph $G = (V, E)$ where V is the set of mobile nodes ($|V| = n$) and $e = (u, v) \in E$ represents a bidirectional wireless link between a pair of nodes u and v if and only if they are within communication range of each other. When two nodes share a bi-directional link, we say that they are neighbors. $d(u, v), u, v \in V^2$ denotes the distance in G (in number of hops) between nodes u and v. We note $\mathcal{H}(u)$ the cluster-head of node u and $\Gamma_k(u)$ the set of k-neighbors of u, *e.g* the set of nodes v such that $d(u, v) \leq k$. $\delta(u) = |\Gamma(u)| = |\Gamma_1(u)|$ is the degree of u.

2.2 Periodic HELLO Protocol

The HELLO protocol was first described in OSPF version 2 [14]. It works as follows. Each node regularly sends a HELLO message to signal its presence to close nodes and maintains an internal neighborhood table. The frequency of these messages is noted f_{HELLO}. When a node u receives such a message from a node v, u adds v to its neighborhood table, or updates the timestamp of the entry if v was already there. HELLO messages may include several kinds of information (see Section 3) but must contain at least the identifier of the sender. Since a neighbor may move or disappear, deprecated entries of the table are regularly removed thanks to a timer. An entry is deprecated when its associated timestamp is too old (it is higher than a threshold AGE_{MAX}): this happens when a neighbor has not signaled itself recently, or when the message HELLO it sent has not been received.

2.3 MAC Layers

The two MAC layers we study are IEEE 802.11 and MadMac[2]. We study the network self-organization over IEEE 802.11 because it is the widespread technology in the field of wireless *ad hoc* networks. Most of current wireless cards use the IEEE 802.11 MAC protocol. Nevertheless, IEEE 802.11 has originally been designed to be used with a base station. Thus its use in *ad hoc* environment leads to some anomalies [5, 11]. Therefore, some alternative MAC layers have been proposed to IEEE 802.11 like MadMac [18]. In order to fairly study the impact of the MAC layer over the HELLO protocol and then over the self-organization, we need also to evaluate the network behavior over an alternative MAC layer to IEEE 802.11.We thus evaluate the network self-organization over a MadMac layer too. MadMac has been shown to be fairer and less aggressive than 802.11.

802.11 and MadMac are both based on CSMA access method. Before transmitting a data packet, each node listens whether the radio channel is free. If so, the transmission can occur, otherwise, it is deferred. The main difference between MadMac and 802.11 relies in the time a node has to wait before triggering a transmission. This time is much higher in MadMac when nodes are overloaded or undergone collisions, that makes the protocol less aggressive. It is worth noting that in both protocols, the broadcast packet are not acknowledged at the MAC layer.

2.4 Self-Organization

In this section, we briefly describe the clustering algorithms we study.For more details, please refer to the corresponding references.

DDR and the density-based algorithms are very similar. They use the same construction algorithm and only differ in the metric they use to take clustering decision. The algorithm they use has been theoretically proved to be self-stabilizing [13]. It builds non-overlapping k-clusters in a distributed fashion by needing only local information (up to 2 hops away). Furthermore, the resulting clusters are not constrained regarding their size neither in terms of number of nodes nor in terms of cluster diameter. They simply adapt the underlying topology. These features allow us to analyze whether a metric can smooth lower layers errors down. Our idea is that when building overlapping clusters or/and 1-clusters, the structure will be impacted faster and it will be more difficult to distinguish the contribution of such or such metric.

DDR uses the node degree as the metric decision. The density-based clustering algorithm is based on a metric called *density*. This link density (noted $\rho(u)$) considers the ratio between the number of links and the number of nodes in $\Gamma(u)$:

$$\rho(u) = \frac{|\{e = (v, w) \in E \mid w \in \{u\} \cup \Gamma(u) \; and \; v \in \Gamma(u)\}|}{\delta(u)}$$

Both algorithms run as follows. Each node periodically computes its degree (for DDR) or its density value (for the density-based algorithm) and broadcasts it to its 1-neighbors *e.g.,* using HELLO packets. Each node is thus able to compare its metric (degree or density) to its 1-neighbors' and decides by itself whether it joins one of them (the one with the highest value) or it wins and elects itself as cluster-head. In case of ties, the node with the lowest identifiers[3] wins. In this way, two neighbors can not be both cluster-heads. If node u has joined node w, we say that w is node u's parent in the clustering tree (noted $\mathcal{P}(u) = w$). A node's parent can also have joined another node and so on. A cluster then extends itself until it reaches another cluster.

3. MOTIVATIONS AND ANALYSIS

3.1 HELLO Packets Format

Each protocol of the literature assumes an ideal MAC layer and consider the Unit Disk Graph model [7], *i.e.* nodes u and v are neighbors if and only if the Euclidean distance from u to v (noted $|uv|$) is lower than R, where R is the transmission range. Links are thus bi-directional. But, when considering a realistic MAC layer, links are not always bi-directional and/or may flap. In our study,

[1] http://www.ieee802.org/11/

[2] In this section we only give a simplified definition of these protocols. Please refer to 802.11 standard and [18] for more details.

[3] Identifiers are supposed unique within the network.

since clustering algorithms are designed for bi-directional links, nodes need to only consider bi-directional links. To make every node aware of the bi-directionality of links, HELLO messages must contain the neighborhood tables of the node. Indeed, if node u receives a HELLO message from node v which does not include node u in the neighbor list, link (u, v) is uni-directional.

Moreover, including the neighborhood table in the HELLO messages in mandatory to compute the link density. Indeed, this allows each node to know the neighbors of its neighbors and thus its 2-neighborhood. By this way, a node discovers links between two of its neighbors and then is able to compute its density value. Last, in order to build clusters, HELLO packets sent by node u must also contain the degree or the density value of u, u's parent ($\mathcal{P}(u)$) and u's cluster-head ($\mathcal{H}(u)$). Table 1 gives the format of a HELLO packet. This format is thus the same for both clustering protocols.

3.2 Motivations

When considering an ideal MAC layer as every clustering scheme of the literature does, parameters such the HELLO packet frequency and the time data is stored do not have any importance. Nevertheless, in a realistic environment, we have to take them into account and measure their impact. Indeed, even in a static environment, some HELLO packet may be lost because of interferences, fading, shadowing and collisions occurring at the MAC layer level. Therefore, the HELLO packet frequency (f_{HELLO}) and the time an entry can remain in the neighborhood table without been refreshed (AGE_{MAX}) have importance in order to provide a node u with a view of its neighborhood as close to the physical one as possible.

Sending too few HELLO messages leads to obsolete tables while sending too many of them may saturate bandwidth to the detriment of data traffic. Estimating the value of AGE_{MAX} is also a difficult task. In most of the literature [17, 8], a node v is generally removed from the neighborhood table of a node u if node u has not received any HELLO packet from node v for $3 * \frac{1}{f_{HELLO}} s$. If AGE_{MAX} is too short, node u consider less neighbors than in reality whereas if the time out is too long, nodes remain in the neighborhood table of node u while they might have disappeared or moved away from yet a long time. In both cases, node u has a wrong view of its environment. Moreover, in wireless ad hoc networks, nodes generally have limited memory size. Thus, if AGE_{MAX} is too high, node memory may quickly be saturated. Hence, both AGE_{MAX} and f_{HELLO} are very important parameters.

3.3 Clustering Analysis in the Ideal Case

In the ideal case, at each round, every node has successfully received a HELLO packet from every of its neighbors. For the ideal case, we consider the Unit Disk Graph model [7]. Therefore, in a static environment and by considering an ideal MAC layer, parameters such AGE_{MAX} and f_{HELLO} do not have any impact on the clustering scheme.

At the bootstrap step, every node is its own parent and its own cluster-head. Its density value is 1, its degree is 0. Upon reception of a HELLO packet, a node u updates these values. Its density and degree are re-computed. Its parent is chosen among its neighbors and node u takes as cluster-head the cluster-head of its parent.

At the end of the first round, every node knows its neighborhood. At the end of the second round, every node learns its 2-neighborhood. From it, it is able to detect whether links are bidirectional and deduces its density value and its degree. At the end of the third round, every node knows the density of their neighbors and is thus able to elect its parent. It will then updates it cluster-head as being the cluster-head of its parent. Before getting the right cluster-head value, each node has to wait $k - 1$ more rounds where $k = d(u, \mathcal{H}(u))$. Table 2 sums up these informations.

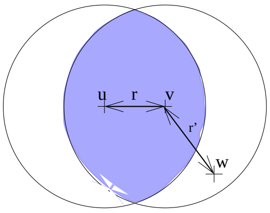

Figure 1: Scenario 1

3.4 Neighbor Discovery Analysis

In this section, we provide a theoretically analysis in order to better understand the relation between each parameter of the HELLO protocol. For it, we fist need to compute the probability for a node to be discovered by its neighbors. Since, we run simulation under the NS2 simulator, we compute this probability under the assumptions of NS2 model. Nevertheless, the model we propose and introduce in Section 3.4.2 holds for any model, as soon as the probability of detection is computed according to the assumptions of the model.

3.4.1 Probability of detection

We first compute the probability for a node to be discovered by its neighbors and deduce from it the mean degree of a node in our simulation environment. First, let's introduce some notations. Let's denote by B_v, the ball centered in v and of radius R. Let $\mathcal{S}_t$ be the set of emitting node at time t. Let $P_v(u)$ be the power received at node v from node u. The analysis we perform in this paper are evaluated over the NS2 simulator[4] in its 2.27 version. Therefore, let's describe how NS2 simulates collisions and interferences at a node. In NS2, $P_v(u)$ is such that $P_v(u) = \frac{P_e}{|uv|^\alpha}$ where P_e is the sending power and $|uv|$ is the Euclidean distance between nodes u and v. Every node of the network sends a message with the same emission power P_e. A node v correctly receives a message from node u at time t if one of the following conditions holds:

- $B_v \cap \mathcal{S}_t = \{u\}$ u *is the only sending node in* $\Gamma(v)$.
- $\forall w \in B_v \cap \mathcal{S}_t$, $\frac{P_v(u)}{P_v(w)} > K$ where K is a constant.

Let's consider the scenario illustrated on Fig. 1. Nodes u, v and w are such that $|uv| = r < R$ (so $u \in \Gamma(v)$), $w \in \Gamma(v) \cap \overline{\Gamma(u)}$. In such case, if both nodes u and w send a message to v, node v will successfully receive the message from A iff

$$\frac{P_v(u)}{P_v(w)} > K \Leftrightarrow \frac{|wv|}{|uv|} > \sqrt[\alpha]{K} \tag{1}$$

Since HELLO packets are broadcast[5] and of small size, they are sent without activating the RTS/CTS mechanism. Moreover, a broadcast packet is not acknowledged so when a broadcast packet is lost because of collisions or interferences, the sending node has no clue about it and will not send the packet again. When a node needs to send a message, it draws a random backoff time in $[0, 32]$. Thus, the probability that two neighbor nodes draw the same backoff value is: $\bar{\mathbb{P}}_{11} = \frac{1}{33}$.

In order to avoid a totally synchronized network, in NS2, a node triggers the transmission of its HELLO packet within the time interval $[1/f_{HELLO} - 0.25, 1/f_{HELLO} + 0.25]$ s. If two nodes u and

[4] http://www.isi.edu/nsnam/ns/

[5] In IEEE 802.11, broadcast packet are sent without RTS/CTS.

Id(u)	$\delta(u)$ or $\rho(u)$	$\mathcal{P}(u)$	$\mathcal{H}(u)$	Neighbor List

Table 1: Format of a HELLO packet

	What a node u learns from its neighbors	What it can then compute
Round 1	uni-directional 1-neighbors	neighborhood table
Round 2	bi-directional 1-neighbors + 2-neighbors	its bi-directional degree + its density
Round 3	bi-directional 1-neighbors + 2-neighbors + its neighbors' density/degree	its parent
Round $k-1$	its parent's cluster-head	its cluster-head

Table 2: Sum up of exchanged information in the ideal case

v trigger their respective transmissions at time t_u and t_v, where $t_v > t_u$, node v will hear u's message on the carrier and will not send its own message (CSMA principle). Before triggering the HELLO packet transmission the time in NS2 is sampled every $0.05s$. Therefore the probability that two nodes choose the exact same time is: $\bar{\mathbb{P}_{12}} = \frac{1}{11}$. All this implies that any two HELLO packets collide with the probability $\bar{\mathbb{P}_1}$ (probability that two neighbor nodes send their HELLO packet exactly at the same time):

$$\bar{\mathbb{P}_1} = \bar{\mathbb{P}}_{11} \times \bar{\mathbb{P}}_{12} = \frac{1}{363} \qquad (2)$$

We are interested in the number of neighbors actually detected in NS2. Therefore, we do not consider the whole network but only a ”typical point” located at the origin of the plane and its 1-neighborhood. Our model is similar to the one implemented in NS2. Let be a Poisson point process on $B(0,R)$ of intensity $\lambda > 0$. We consider a point 0 at the origin for which we study the detected neighborhood (Palm distribution [19]). Let u be a point at distance r ($r \leq R$) from the origin. We fix the two points 0 and u and we distribute the Poisson point process in $B(0,R)$ independently of these two points. From it, we study the probability $\mathbb{P}(r)$ that node u is detected by node 0. Let's introduce a random variable X which counts the number of the points of the Poisson Point Process lying in $B(0,R)$. Then, the number of neighbors $\mathbb{E}^o_{detected}$ detected by node 0 after one exchange of HELLO packet is:

$$\mathbb{E}^o_{detected} = \int_0^R \lambda 2\pi r \mathbb{P}(r) dr \qquad (3)$$

We have :

$$\mathbb{P}(r) = \mathbb{P}_1 * \mathbb{P}_2(r) \qquad (4)$$

where $\mathbb{P}_1(r)$ is the probability that node u and node 0 do not send a message at the same time, $\mathbb{P}_2(r)$ is the probability that node u is not disturbed by a communication in node 0's neighborhood. As already mentioned, according to NS2 assumptions (2), $\mathbb{P}_1$ is such that:

$$\mathbb{P}_1 = 1 - \bar{\mathbb{P}_1} = \frac{362}{363} \qquad (5)$$

As no RTS/CTS neither ACK mechanisms are used, probabilities for drawing a backoff time are independent and the same for each node. $\mathbb{P}_2(r)$ is the probability that node u is not disturbed by a communication in node 0's neighborhood. This holds if no other node $w \in \Gamma(0)$ emits at the same time than u or if at least a node $w \in \Gamma(0)$ emits at the same time than node u but $P_0(w)$ is not strong enough to disturb node u. The probability there is another neighbor of node 0 which emits at the same time than node u will depend on X. Thus:

$$\mathbb{P}_2(r) = \sum_{n=0}^{\infty} \mathbb{P}_{2_n}(r) \mathbb{P}[X = n] \qquad (6)$$

where $\mathbb{P}_{2_n}(r)$ is the probability that a node at distance r from 0 is detected by 0 knowing that $X = n$.

As shown by (1), if a node w emits at the same time than node u, node u will not be detected only if w is such that $|w0| > \sqrt[\alpha]{K} * |u0|$. So, that means that, if $d(u,0) > \frac{R}{\sqrt[\alpha]{K}}$, node u can be detected only if no other node in $\Gamma(0)$ emits at the same time than itself. So, if $X = n$, for $r > \frac{R}{\sqrt[\alpha]{K}}$: $\mathbb{P}_{2_n}(r) = \mathbb{P}_1^n$. Otherwise, if $r < \frac{R}{\sqrt[\alpha]{K}}$, node u is detected even if k nodes in $B(0,R)$ send a message at the same time, as soon as each of these k nodes w is such that $|0w| > \sqrt[\alpha]{K} * r$. We thus have, if $X = n$, for $r < \frac{R}{\sqrt[\alpha]{K}}$:

$$\begin{aligned} \mathbb{P}_{2_n}(r) &= \sum_{k=0}^{n} \binom{n}{k} \mathbb{P}_1^{n-k} (1 - \mathbb{P}_1)^k \left(\frac{\pi R^2 - \pi(\sqrt[\alpha]{K} r)^2}{\pi R^2} \right)^k \\ &= \left((1 - \mathbb{P}_1) \left(1 - \frac{(\sqrt[\alpha]{K} r)^2}{R^2} \right) + \mathbb{P}_1 \right)^n \end{aligned}$$

Then, we have :

$$\mathbb{P}_{2_n}(r) = \begin{cases} \left((1 - \mathbb{P}_1) \left(1 - \frac{(\sqrt[\alpha]{K} r)^2}{R^2} \right) + \mathbb{P}_1 \right)^n & \text{if } r < \frac{R}{\sqrt[\alpha]{K}} \\ \mathbb{P}_1^n & \text{otherwise.} \end{cases}$$

Since nodes are distributed according to a Poisson Point Process of intensity λ:

$$\mathbb{P}[X = n] = \frac{(\lambda \pi R^2)^n}{n!} e^{-\lambda \pi R^2}$$

We can now compute $\mathbb{P}_2(r)$. If $r < \frac{R}{\sqrt[\alpha]{K}}$,

$$\begin{aligned} \mathbb{P}_2(r) &= \sum_{n=0}^{\infty} \left((1 - \mathbb{P}_1) \left(1 - \frac{(\sqrt[\alpha]{K} r)^2}{R^2} \right) + \mathbb{P}_1 \right)^n \frac{(\lambda \pi R^2)^n}{n!} e^{-\lambda \pi R^2} \\ &= e^{-\lambda \pi (\sqrt[\alpha]{K} r)^2 (1 - \mathbb{P}_1)} \end{aligned}$$

And if $r > \frac{R}{\sqrt[\alpha]{K}}$, $\mathbb{P}_2(r) = \sum_{n=0}^{\infty} \mathbb{P}_1^n \frac{(\lambda \pi R^2)^n}{n!} e^{-\lambda \pi R^2} = e^{-\lambda \pi R^2 (1 - \mathbb{P}_1)}$. Thus:

$$\mathbb{P}_2(r) = \begin{cases} e^{-\lambda \pi (\sqrt[\alpha]{K} r)^2 (1 - \mathbb{P}_1)} & \text{if } r < \frac{R}{\sqrt[\alpha]{K}} \\ e^{-\lambda \pi R^2 (1 - \mathbb{P}_1)} & \text{otherwise.} \end{cases} \qquad (7)$$

Finally, we can compute the mean number of neighbors a node 0 detects:

$$\begin{aligned} \mathbb{E}^o_{detected} &= \textstyle\int_0^R 2\lambda \pi r \mathbb{P}_1 \mathbb{P}_2(r) dr \\ &= \textstyle\int_0^{\frac{R}{\sqrt[\alpha]{K}}} 2\lambda \pi r \mathbb{P}_1 e^{-\lambda \pi (\sqrt[\alpha]{K} r)^2 (1 - \mathbb{P}_1)} dr \\ &\quad + \textstyle\int_{\frac{R}{\sqrt[\alpha]{K}}}^R 2\lambda \pi r \mathbb{P}_1 e^{-\lambda \pi R^2 (1 - \mathbb{P}_1)} dr \\ &= \mathbb{P}_1 \left(\tfrac{1 - e^{-\lambda \pi R^2 (1 - \mathbb{P}_1)}}{(1 - \mathbb{P}_1)(\sqrt[\alpha]{K})^2} + \lambda \pi R^2 e^{-\lambda \pi R^2 (1 - \mathbb{P}_1)} (1 - \left(\tfrac{1}{\sqrt[\alpha]{K}} \right)^2) \right) \end{aligned}$$

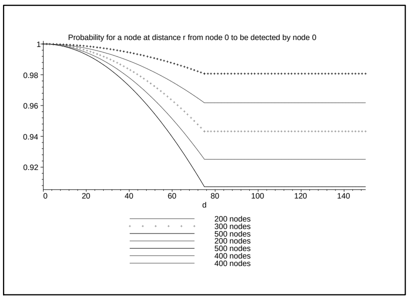

Figure 2: Probability for a node u to be detected by node 0 as a function of $d = |u0|$

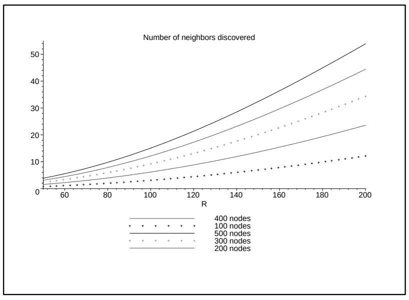

Figure 3: Average number of neighbors detected per node

and the probability $\tilde{\mathbb{P}}$ for a node to be detected by its neighbor:

$$\tilde{\mathbb{P}} = \frac{\mathbb{E}^{o}_{detected}[\lambda]}{\lambda \pi R^2}$$
$$= \mathbb{P}_1 \left(\frac{1-e^{-\lambda\pi R^2(1-\mathbb{P}_1)}}{(1-\mathbb{P}_1)(\sqrt[\alpha]{K})^2(\lambda\pi R^2)} + e^{-\lambda\pi R^2(1-\mathbb{P}_1)}(1 - \left(\frac{1}{\sqrt[\alpha]{K}}\right)^2) \right) (8)$$

Fig. 2 and 3 plot the theoretical results for $K = 16$ and $\alpha = 4$ (default parameters of NS2). One can notice that the higher the nuber of neighbors to discover, the higher the imprecision.

Nevertheless, like already claimed, links can be uni-directional because of propagation errors. The node degree $\tilde{\delta}$ only counts bi-directional links and thus is the number of nodes detected by node 0 and which have also detected 0.

$$\tilde{\delta} = \int_0^R 2\lambda\pi r \mathbb{P}(r)^2 dr$$
$$= \mathbb{P}_1^2 \left(\frac{1-e^{-2\lambda\pi R^2(1-\mathbb{P}_1)}}{(1-\mathbb{P}_1)(2\sqrt[\alpha]{K})^2} + \lambda\pi R^2 e^{-2\lambda\pi R^2(1-\mathbb{P}_1)}(1 - \left(\frac{1}{\sqrt[\alpha]{K}}\right)^2) \right)$$

Fig. 4 compares the node degree in an ideal case with the one observed theoretically with MadMac or IEEE 802.11. Results clearly show that a realistic MAC layer impacts the neighbor discovery and the bi-directionality of wireless links.

3.4.2 Evaluating the parameters

In this section, based on the probability detection and the mean effective node degree provided by Section 3.4.1, we try to theoretically capture the dynamic evolution of the neighborhood table of each node based on HELLO packet parameters f_{HELLO} and AGE_{MAX}. To do so, we model the neighborhood table of a node with a M/M/c/c Markov chain by considering the Erlang formulas for birth and death process with c servers [3]. Let λ_f be the frequency of HELLO packet reception at a node (note that when propagation is error-free, $\lambda_f = f_{HELLO}$), and μ_v the removal rate of information for node 0 neighborhood (*i.e.* $1/AGE_{MAX}$).

A M/M/c/c chain captures the evolution of informations contained in the neighborhood table of a typical node. We consider

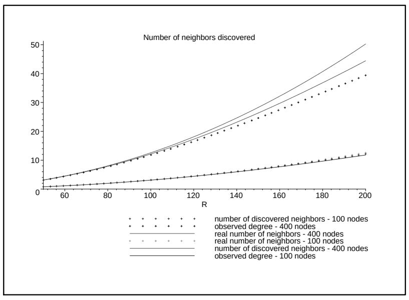

Figure 4: Comparison of the number of real neighbors and the number of the ones discovered by a node in NS2

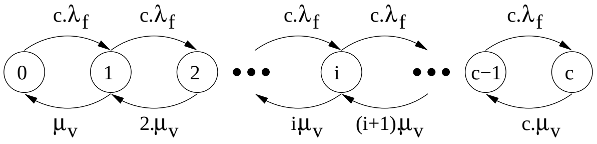

Figure 5: M/M/c/c queue

node 0. The arrival rate of a HELLO packet is represented by a Poisson Process of parameter λ_f. The departure rate of the information contained in a HELLO packet in the neighborhood table follows an exponential law with parameter μ_v. We chose an exponential law for the sake of simplicity and tractability. The steady state probability distribution of a M/M/c/c Markov chain is similar to a M/G/c/c probability distribution[3]. In the latter one, the departure distribution function can be a constant, that is similar to the regular removal of entries in the neighborhood table. Our system contains c servers (the information received from each neighbor of node 0 is considered by node 0 like a client in the system). We assume that the number of physical neighbors that has to be discovered is known. When the system is full (every node is known), each new arrival does not bring any new information, so the packet is not useful. Thus, our system has a queue of length 0. Even if the number of neighbors is known, we do not consider a finite population size. As there is no preemption in the queue policy management, we make the assumption that each received packet (when the system is not full) brings additional information to the observed node. This assumption holds because we cannot distinguish the nodes. This model assumes that there is no preemption when a packet is received. In other words, when node 0 receives a packet from node u, the information is stored in the node 0's table during $1/\mu_v$ and can not be refreshed. Fig. 5 shows the graphical representation of our chain. Such a chain represents the state of the neighborhood table of node 0. If the system (node 0) is in state i, that means that node 0 has discovered i nodes out of its c neighbors. Note that as in every Markov chain, the probability of simultaneous events is 0, thus receiving simultaneous HELLO packets is impossible. At steady state, the probability to be in state k is given by:

$$\pi^{(k)} = p^k . \binom{c}{k} . \pi^{(0)}$$

where, $p = \dfrac{\lambda_f}{\mu_v}$ and $\pi^{(0)} = \dfrac{1}{1 + \sum_{i=1}^{c} p^i . \binom{c}{i}}$.

We are interested in the probability for the system to remain in state c, *i.e.*, the probability $\pi^{(c)}$. $\pi^{(c)}$ is the probability for a given node to have a complete knowledge of its neighborhood at a given point of time. $\pi^{(c)}$ is such that:

$$\pi^{(c)} = \frac{c^c}{c!} \cdot \frac{p^c}{\sum_{i=0}^{c} p^i \cdot \frac{1}{c!}}$$

λ_f is the correct packet arrival rate at node 0 (rate of successful reception of a HELLO packet) and can be written as:

$$\lambda_f = f_{HELLO} \times \tilde{\mathbb{P}}$$

where $\tilde{\mathbb{P}}$ is the probability of successful reception at node 0 (see Section 3.4.1). Fig. 6 and 7 plot the probability for the Markov chain to be in state $\pi^{(i)}$, depending on the number of neighbors. The different curves show the influence of p (and so of the HELLO protocol parameters). As expected, an increasing value of p increases the probability to be in state $\pi^{(c)}$. We can also see that even for a high value of p, the probability for a node to have the full knowledge of its neighborhood is not 1. For a higher degree, the probability to have a complete knowledge is the same as with a low degree for a fixed value of p. These values constitute a lower bound, because of the non preemptive update of information in the Markov chain, but they give a good intuition of the evolution of the neighborhood table and of the impact of the HELLO protocol parameters. One has also to notice that an increasing degree also decreases the value of $\tilde{\mathbb{P}}$ and thus reduces the value of p. Therefore, in order to maintain the same probability $\pi^{(c)}$, when the degree changes, one has to increase the HELLO packet frequency and/or to increase the value of AGE_{MAX}.

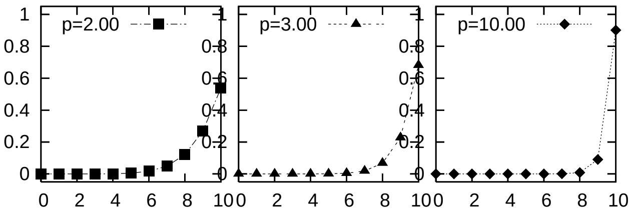

Figure 6: $c = 10$

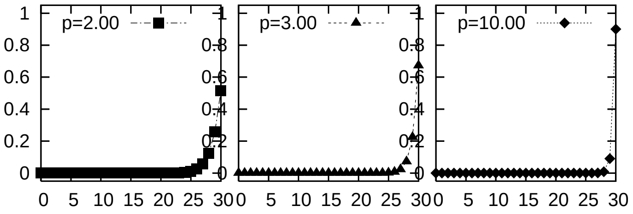

Figure 7: $c = 30$

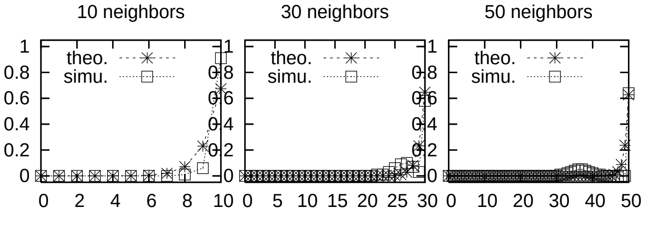

Figure 8: Comparison between analysis and simulation for $f_{HELLO} = 1$ and $\mu_v = 1/3$

Fig. 8 shows the comparison between theoretical results and simulations by plotting the probability $\pi^{(i)}$ for $f_{HELLO} = 1$ and $\mu_v = 1/3$, considering the probability $\tilde{\mathbb{P}}$ (8). For the simulations, we run 100 simulations of 50 seconds each and we observed the evolution of the neighborhood table of a given node. Figures show the accuracy of the proposed model for different intensities and for the observed node. From these results, we can say that to behave correctly, protocols relying on the neighborhood discovery have to deal with the inconsistent state of the neighborhood table at each point of time. It is worth noting that this inconsistent state is worsen by mobility and increasing the value of p is not suitable for mobility handling. The next step of this analysis is to include mobility handling.

4. RESULTS

4.1 Simulation Model and Parameters

All simulations are performed over the NS2 simulator in its 2.27 version. The simulator has been modified to reflect the 802.11b DSSS parameter. We have also modified the communication range of each node to reflect 802.11b card based on the specification of wireless card. Nodes are randomly deployed using a Poisson Point Process in a $1000m \times 1000m$ square with various levels of intensity λ. In such processes, λ represents the mean number of nodes per surface unit. The communication range R is set to $150m$ in all tests. Simulation are run during $30s$. The results obtained are within a 95% - confidence interval. No mobility pattern is used.

Both clustering algorithms are evaluated over a 802.11 MAC layer and a MadMac layer for each node distribution. Each result is compared to the results obtained with the use of an ideal MAC layer. The ideal MAC layer refers to a Unit Disk Graph model with no interferences, no packet collisions, each message sent is immediately and successfully received by each node within communication range. Each association of protocols MAC-clustering is evaluated for three values of f_{HELLO}: $1/f_1 = 1s$, $1/f_2 = 2s$, $1/f_3 = 3s$ (where when $1/f_{HELLO} = x$ a node sends a HELLO packet every x seconds). For each combination $(MAC - Clustering - f_{HELLO})$, we removed entries in data base older than AGE_{MAX} by using three values for AGE_{MAX}: $AGE_1 = \frac{1}{f_{HELLO}}$, $AGE_2 = \frac{2}{f_{HELLO}}$ and $AGE_3 = \frac{3}{f_{HELLO}}$.

Contrarily to the ideal MAC layer, links may become uni-directional when using a realistic MAC layer. Since the clustering algorithms require bi-directional links, a node will not consider uni-directional link when computing its density.

4.2 Model Validation

To validate our NS2 code, we use a simple scenario depicted in Fig. 9. In this scenario the lines represent the wireless links between nodes. Table 3 gives the value computed by the nodes during the simulation. Results validate our model.

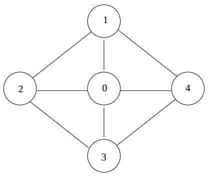

Figure 9: Validation scenario

4.3 Simulations Results

This section presents the simulation results of our proposed implementation.

	$\delta(u)$		$\rho(u)$		$\mathcal{H}(u)$		$\mathcal{P}(u)$	
Node	Theo.	Simul.	Theo.	Simul.	Theo.	Simul.	Theo.	Simul.
0	4	4	$8/4$	2.00	0	0	0	0
1	3	3	$5/3$	1.66	0	0	0	0
2	3	3	$5/3$	1.66	0	0	0	0
3	3	3	$5/3$	1.66	0	0	0	0
4	3	3	$5/3$	1.66	0	0	0	0

Table 3: Validation

4.3.1 *Neighbor discovery*

As claimed earlier, the use of realistic MAC and physical layers greatly impacts the neighbor discovery. In this section, we run simulations in order to validate the theoretical analysis provided in Section 3.4.1. Fig. 10 plots the mean node degree, *i.e.* the mean number of bi-directional neighbors for $R = 150m$ and different values of λ after one round of HELLO messages exchange. Results perfectly match.

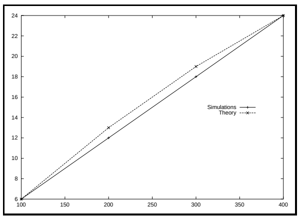

Figure 10: Mean node degree by theory and simulation as a function of λ for $R = 150$

4.3.2 *The influence of the timeout for data refresh*

We show how the time data are stored influences the clustering protocols. Fig. 11 and 12 plot the number of clusters for the two different clustering protocols as a function of the time with $f_{HELLO} = f_1$. Results are plot here for $\lambda = 100$ but the clustering behaviors are similar whatever the number of nodes.

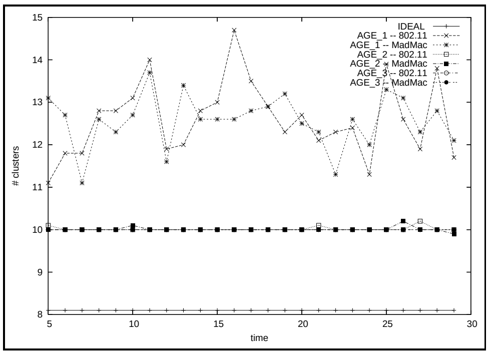

Figure 11: Number of clusters obtained for $\lambda = 100$ and $f_{HELLO} = f_1$ when using the density-based algorithm

We only present results for $f_{HELLO} = f_1$. But, results for $f_{HELLO} = f_2$ and $f_{HELLO} = f_3$ present the same oscillations. The lower the frequency, the more the system oscillates, the longer it needs to stabilize, as we will show later in Section 4.3.3. Results plotted here are thus the better behavior we can expect.

One can notice that depending on how fresh the data considered are, the clustering algorithms converge toward different values, but never aims at the one obtained in the ideal case. This can be explained by the theoretical analysis of Section 3.4. Indeed, as shown by this analysis, the probability to detect every neighbor node in only one round is very low, indeed, when $AGE_{MAX} = AGE_1$, none node has an exact view of its neighborhood at any time. And, to compute the degree or the density value, nodes need not only their own neighborhood but also the neighborhood of their neighbors. This explains that, when AGE_{MAX} is close to the frequency of HELLO packets, we observe a kind of oscillation in the number of clusters and algorithms have difficulty to self-stabilize. The neighborhood table and data used for computing the density/degree value computation changes frequently. We can see that this behavior appears for both MAC layer protocols. In fact, the delay introduced at the MAC layer modifies the status of each node. When assuming a perfect but not instantaneous transmission of messages, for example a TDMA scheduling where no collision occurs, every message transmission has the same delay and the jitter is equal to 0. This behavior does not appear here, transmissions are thus not perfect and some HELLO packets are lost. While considering a random access method, the jitter varies because of contention. In 802.11 and MadMac, this aspect is worsen by the random backoff algorithm used and this oscillation appears because there exists a high probability that a HELLO packet sent by node u is not received by all nodes in $\Gamma(u)$ before the end of the AGE_{MAX} period. Increasing the frequency of HELLO packet or increasing the AGE_{MAX} of data validity is thus the solution to this problem. Indeed, this amounts to increasing the value of μ_v in the analysis.

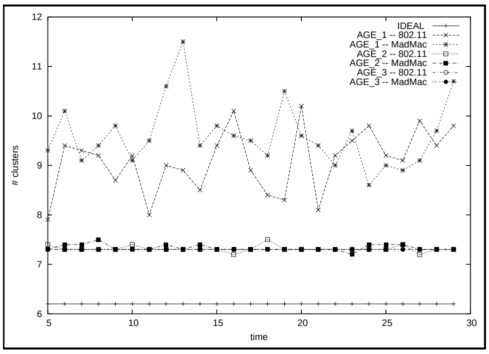

Figure 12: Number of clusters obtained for $\lambda = 100$ and $f_{HELLO} = f_1$ when using DDR

Fig. 11 and 12 show that having AGE_{MAX} equal to twice the f_{HELLO} (thus $p = 2\tilde{\mathbb{P}}$ from Section 3.4) is enough to avoid this oscillation problem, whatever the clustering protocol and whatever the network density. This confirms the theoretical results of Section 3.4. By increasing the AGE_{MAX} period, the clustering protocol behavior tends toward the ideal case. Moreover, we can notice that the density-based algorithm oscillates less strongly than DDR. This shows that the density metric allows to smooths the small topology changes down and so, that the metric used in layer-3 protocols may be useful to overcome the imperfections of the lower layers. We can also notice that oscillations are stronger when using 802.11 when $AGE_{MAX} \neq AGE_1$ as a MAC layer rather than MadMac, whatever the clustering algorithm. This is due to the fact that MadMac is fairer than 802.11 and that in average it allows more nodes to access the channel. If at round 1, node u cannot send its HELLO message, it is more likely to send its HELLO packet at round 2 with MadMac than with 802.11.

4.3.3 *The influence of HELLO frequency*

In this section, we evaluate the influence of the HELLO frequency. For it, we evaluate the different scenarii for several values of f_{HELLO} and by fixing AGE_{MAX}. Fig. 13 and 14 show the evolution in time of the number of clusters depending for AGE_3 and $\lambda = 100$. We can see that when the HELLO packet frequency decreases, the convergence time of the algorithm increases, which was expected since the more often data are sent, the faster each node updates its neighborhood table. We can see, as expected, that the time needed for the algorithm to converge is roughly proportional to the HELLO frequency. This behavior is the same whatever the MAC layer protocol and whatever the metric used for clustering. Because of page restriction, we only show here the behavior of the system for AGE_3 and $\lambda = 100$ but results are the same for other AGE_{MAX} values and other intensity values. These observations confirm the analytical results where the value of p impacts the network. Thus, for a fixed value of AGE_{MAX}, the HELLO packet frequency only influences the convergence time.

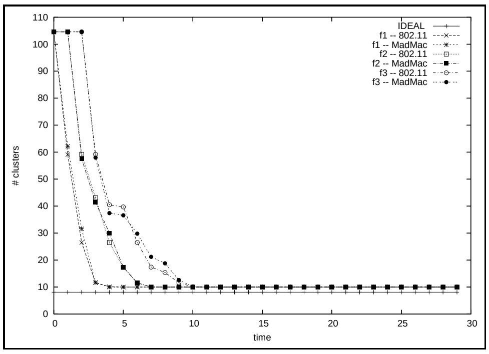

Figure 13: Number of clusters, for $\lambda = 100$ and AGE_3 when using the density-based algorithm

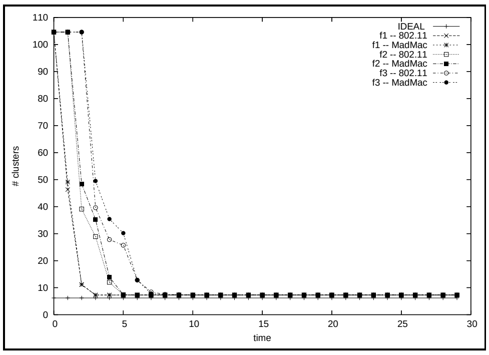

Figure 14: Number of clusters, for $\lambda = 100$ and AGE_3 when using the DDR algorithm

5. CONCLUSION AND FUTURE WORKS

In this paper, we analyzed the impact of a realistic protocol stack over a self-organization depending of several parameters. We showed that the choice of the MAC layer impacts only in small measures the self-organization even if using MadMac instead of IEEE 802.11 as MAC layer protocol allows to discover nodes a little bit quicker. We also showed that using the density metric instead of the degree smooths local link failures down and gives more stability to the self-organization. The main disturbance comes from the setting of the neighbor discovery protocol. Indeed, we showed theoretically and by simulation that even in the best case, a node never discovers all its neighbors. We give a theoretical way to set the HELLO protocol parameters according to the discovery probability required. The packet frequency f_{HELLO} and the time information is stored if not refreshed AGE_{MAX} are linked and the parameter to set up is the ratio of both of them. These first results show that studying the HELLO protocol parameters is important when the informations in the HELLO packet are necessary for some other protocols. The next step of this work is to optimize the HELLO protocol for a specific (clustering, MAC) combination.

As future works, we intend to lead similar analysis when considering node mobility and the presence of traffic. We present some preliminary results on the simple topology given in Fig. 9. In this scenario, we try to include a saturated data traffic from node 1 to 0. The results of a $30s$ simulation is given in Table 4 with a frequency of HELLO packets equal to $1.s^{-1}$ and a valid data age of $3s$.. Theoretical results are the same as in Table 3. It is worth noting that this does not mean that the clustering algorithm has converged correctly. At this point, nothing can be concluded except that when some traffic is present in the network the clustering algorithm can be strongly affect. More simulations and theoretical analysis are needed to conclude anything and to understand the network behavior in such cases. Indeed, introducing any kind of self-organization into a network aims at bringing more advantages to the network and just consists in a basis. If having some regular traffic totally disturbs the self-organization, we have to study in what measures the self-organization is useful to the network and does not harm it and what the limits of the self-organization are.

	$\rho(u)$		$\mathcal{H}(u)$		$\mathcal{P}(u)$	
Node	802.11	MadMac	802.11	MadMac	802.11	MadMac
0	1	2.00	0	0	1	0
1	1.66	1.66	1	0	1	0
2	1	1.66	0	0	0	0
3	1.66	1.66	3	0	3	0
4	1	1.66	0	0	1	0

Table 4: Node values

6. REFERENCES

[1] A. Amis, R. Prakash, T. Vuong, and D. Huynh. Max-Min d-cluster formation in wireless *ad hoc* networks. In *INFOCOM*, March 2000.

[2] P. Basu, N. Khan, and T. Little. A mobility based metric for clustering in mobile *ad hoc* networks. In *DISC*, 2001.

[3] S. K. Bose. *An Introduction to Queueing Systems*. Kluwer/Plenum Publishers, 2002.

[4] M. Chatterjee, S. K. Das, and D. Turgut. WCA: A weight based distributed clustering algorithm for mobile *ad hoc* networks. *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5(2):193–204, April 2002.

[5] C. Chaudet, D. Dhoutaut, and I. Guérin-Lassous. Experiments of some performance issues with IEEE 802.11b in *ad hoc* networks. In *WONS*, 2005.

[6] G. Chen, F. Garcia, J. Solano, and I. Stojmenovic. Connectivity-based k-hop clustering in wireless networks. In *HICSS*, January 2002.

[7] B.N. Clark, C.J. Colbourn, and D.S. Johnson. Unit Disk Graphs. *Discrete Math.*, 86(1-3), 1990.

[8] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol, October 2003. RFC 3626.

[9] A. Ephremides, J.E. Wieselthier, and D.J. Baker. A design concept for reliable mobile radio networks with frequency hoping signaling. In *IEEE 75*, 1987.

[10] M. Gerla and J. T-C. Tsai. Multicluster, mobile, multimedia radio network. *ACM/Baltzer Journal of Wireless Networks*, 1(3), July 1995.

[11] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda. Performance anomaly of 802.11b. In *INFOCOM*, 2003.

[12] N. Mitton, A. Busson, and E. Fleury. Self-organization in large scale ad hoc networks. In *MedHocNet*, June 2004.

[13] N. Mitton, E. Fleury, I. Guérin-Lassous, and S. Tixeuil. Self-stabilization in self-organized multihop wireless networks. In *WWAN*, June 2005.

[14] J. Moy. OSPF - Open Shortest Path First, 1994. RFC 1583.

[15] N. Nikaein, H. Labiod, and C. Bonnet. DDR-distributed dynamic routing algorithm for mobile *ad hoc* networks. In *MobiHoc*, November 2000.

[16] Tomoyuki Ohta, Shinji Inoue, and Yoshiaki Kakuda. An adaptive multihop clustering scheme for highly mobile *ad hoc* networks. In *ISADS*, April 2003.

[17] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-demand Distance Vector Routing, July 2003. RFC 3561.

[18] T. Razafindralambo and I. Guérin-Lassous. Increasing fairness and efficiency using the MadMac protocol in ad hoc networks. In *IFIP Networking*, 2006.

[19] D. Stoyan, S. Kendall, and J. Mecke. *Stochastic geometry and its applications, second edition*. John Wiley & Sons, 1995.

[20] A. C. Viana, M. Dias de Armorim, S. Fdida, and J. Ferreira de Rezende. Self-organization in spontaneous networks: the approach of DHT-based routing protocols. *Ad Hoc Networks Journal*, 2005.