Efficient Broadcasting in Self-Organizing Multi-Hop Wireless Networks.

Nathalie Mitton, Eric Fleury

Efficient Broadcasting in Self-Organizing Multi-hop Wireless Networks

Nathalie Mitton and Eric Fleury

INRIA/ARES - INSA de Lyon - 69621 VILLEURBANNE Cedex
tel: 33-(0)472-436-415
`firstname.lastname@insa-lyon.fr`

Abstract. Multi-hop wireless networks (such as ad-hoc or sensor networks) consist of sets of mobile nodes without the support of a pre-existing fixed infrastructure. For the purpose of scalability, ad-hoc and sensor networks may both need to be organized into clusters and require some protocols to perform common global communication patterns and particularly for broadcasting. In a broadcasting task, a source node needs to send the same message to all the nodes in the network. Some desirable properties of a scalable broadcasting are energy and bandwidth efficiency, *i.e.*, message retransmissions should be minimized. In this paper, we propose to take advantage of the characteristics of a previous clustered structure to extend it to an efficient and scalable broadcasting structure. In this way, we build only one structure for both operations (organizing and broadcasting) by applying a distributed clustering algorithm. Our broadcasting improve the number of retransmissions as compared to existing solutions.
keywords: multi-hop wireless networks, self-organization, broadcasting.

1 Introduction

Multi-hop wireless networks (such as ad-hoc or sensor networks) consist of sets of mobile wireless nodes without the support of a pre-existing fixed infrastructure. They offer unique benefits for certain environments and applications as they can be quickly deployed. Each node acts as a router and may arbitrary appear or vanish. Protocols must adapt to frequent changes of the network topology. Ad-hoc networks and sensor networks are instances of multi-hop wireless networks. In ad-hoc networks, nodes are independent and may move at any time at different speeds. In sensor networks, nodes are more static and collect data they have to forward to specific nodes. For the purpose of scalability, ad-hoc and sensor networks may both need to be organized into clusters and require some protocols to perform common global communication patterns as for broadcasting. An organization is needed to allow the scalability in terms of number of nodes or/and node density without generating too much traffic (for routing, for instance) neither too much information to store. A common solution is to adapt a hierarchical organization by grouping nodes into clusters and bind them to a leader. Such an organization may allow the application of different routing schemes in and between clusters. In a broadcasting task, a source node needs to send the same message to all the nodes in the network. Such a functionality is needed, for example, when some queries about the measures (in sensor networks) or a node location (in ad hoc networks) need

to be disseminated over the whole network or within a cluster. Broadcasting in a cluster may also be useful for synchronizing nodes. The desirable properties of a scalable broadcasting are reachability, energy and bandwidth efficiency.

In this paper, we propose to take advantage of the characteristics of our previous wireless network clustered structure [9] to extend it to an efficient and scalable broadcasting structure. In this way, we build only one structure for both operations (organizing and broadcasting) by applying a distributed clustering algorithm. The resulting broadcasting, analyzed and compared to some other existing protocols, saves more retransmissions. The remainder of this paper is organized as follows. Section 2 presents some previous broadcasting solutions. Section 3 summarizes our previous work and highlights some characteristics of our cluster organization which might be useful for a broadcasting task. Section 4 presents the way we extend our cluster structure into a broadcasting structure and details our broadcasting scheme. Section 5 compares several broadcasting schemes by simulation and presents the results. Finally, we conclude in Section 6 by discussing possible future areas of investigation.

2 Broadcasting in multi-hop wireless networks

The desirable properties of a scalable broadcasting are reachability, energy and bandwidth efficiency. Indeed, as in a wireless environment, a node wastes energy when transmitting as well as receiving a packet, the number of retransmissions and receptions should be minimized. In this paper, we only consider reliability at the network layer, *i.e.*, a broadcasting scheme is said reliable if every node connected to the source receives a broadcast packet in a collision free environment. In this section, we focus on the solutions proposed in the literature for network layer broadcasting schemes which are based on dominating set and use omni-directional antennas.

The easiest way to broadcast a message over a network is the blind flooding, *i.e.*, each node re-emits the message upon first reception of it. Obviously, this causes many collisions and wastes bandwidth and energy. Therefore, this broadcasting technique can not be envisaged over large scale or very dense networks. This gave birth to more intelligent broadcasting protocols which try to minimize the number of retransmissions by selecting a subset of nodes allowed to forward a message. This subset is called a dominating set. To obtain a reliable broadcasting scheme, each node in the network should be either in the dominating set (and is called an internal node) or neighboring at least one node in the dominating set. The main challenge is to find a connected dominating set which minimizes the number of these transmitters as well as the number of copies of a same message received by a node. I. Stojemovic and J. Wu [15] classify the broadcasting schemes according to the kind of dominating set they use: cluster-based, source-dependent dominating set and source-independent dominating set schemes. All of them provide a reliable broadcasting task with a relevant number of retransmissions saved, compared to the blind flooding.

In oldest solutions, cluster-based, [3,5,8], the idea is that every node which has the lowest Id or the highest degree (Linked Cluster Architecture-LCA protocol) in its 1-neighborhood becomes a cluster-head. If a non-cluster-head node can hear more than one cluster-head among its neighbors, it becomes a gateway. The dominating set is thus

composed of both the cluster-heads and the gateways. From it, some optimizations have been proposed to localize the maintenance process and avoid the chain reaction which can occur in case of node mobility [4] or to limit the number of gateways [17].

In solutions based on source-dependent dominating set [7,13], the sending nodes select adjacent nodes that should relay the message. The set of relays of a node u is chosen to be minimal and such that each 2-hop neighbor of node u has at least one neighbor among the relays of u. Methods differ in details on how a node determines its forwarding list. The most popular of them is the one based on the Multi-Point Relay (MPR) of OLSR [13]. In OLSR, the MPR are also used for propagating the routing information. This kind of structure has thus a double use too.

In solutions based on source-independent dominating set, the set of internal nodes is independent of the source node. This is the case of our proposal. Many solutions have been proposed. A simple and efficient algorithm, the NES (*Neighbors Elimination Based Scheme*) [14,16], introduces the notion of *intermediate* nodes. Node A is *intermediate* if at least two of its neighbors are not direct neighbors. Two selection rules are then introduced to reduce the number of transmitter nodes. From it, several solutions have thus been derived [2,14].

3 Previous work and main objectives

In this section, we summarize our previous clustering work on which our broadcasting scheme proposition relies. Only basis and features which are relevant for broadcasting are mentioned here. For more details or other characteristics of our clustering heuristic, please refer to [9,10,11]. 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. We note $\mathcal{C}(u)$ the cluster owning the node u and $\mathcal{H}(u)$ the cluster-head of this cluster. We note $\Gamma_1(u)$ the set of nodes with which u shares a bidirectional link. $\delta(u) = |\Gamma_1(u)|$ is the degree of u.

Our objectives for introducing our clustering algorithm were motivated by the fact that in a wireless environment, the less information exchanged or stored, the better. First, we want a cluster organization suitable for large scale multi-hop networks, *i.e.*, non-overlapping clusters not restricted to a given fixed radius/diameter but with a flexible radius (*The clustering schemes mentioned in Section 2 have a radius of* 1, *in [1,6] the radius is set a priori*) and able to adapt to the different topologies. Second, we want the nodes to be able to compute the heuristic from local information, only using their 2-neighborhood. *In [1], if the cluster radius is set to d, the nodes need to gather information up to d hops away before taking any decision.* Finally, we desire an organization robust and stable over node mobility, *i.e.*, that we do not need to re-compute for each single change in the topology. Therefore, we introduce a new metric called *density*. The notion of density characterizes the "relative" importance of a node in the network and within its neighborhood. The underlying idea is that this link density (noted $\rho(u)$) should smooth local changes down in $\Gamma_1(u)$ by considering the ratio between the number of links and the number of nodes in $\Gamma_1(u)$.

Definition 1 (density).
  The density of a node $u \in V$ is $\rho(u) = \frac{|\{e=(v,w) \in E \mid w \in \{u\} \cup \Gamma_1(u) \ and \ v \in \Gamma_1(u)\}|}{\delta(u)}$.

On a regular basis, each node locally computes its density value and regularly locally broadcasts it to its 1-neighbors (*e.g.*, using `Hello` packets). Each node is thus able to compare its density value to its 1-neighbors' and decides by itself whether it joins one of them (the one with the highest density value) or it wins and elects itself as cluster-head. In case of ties, the node with the lowest Id 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 parent in the clustering tree (noted $\mathcal{P}(u) = w$) and that node u is a child of node w (noted $u \in \mathcal{C}h(w)$). A node's parent can also have joined another node and so on. A cluster then extends itself until it reaches another cluster. The cluster-head is the node which has elected itself. If none of the nodes has joined a node u ($\mathcal{C}h(u) = \emptyset$), u becomes a leaf. Thus, in this way, as every node chooses itself a parent among its 1-neighbors, a cluster can also be seen as a directed tree which root is the cluster-head. When building clusters, we then also build a spanning forest composed of as many directed acyclic graphs (DAG) as clusters.

3.1 Some characteristics of this clustering algorithm

This algorithm stabilizes when every node knows its *correct* cluster-head value. It has been proved by theory and simulations to self-stabilize within an expected low, constant and bounded time [11]. It has also been proved that a cluster-head is aware of an information sent by any node of its cluster in a low, constant and bounded time. The number of clusters built by this heuristic has been showed analytically and by simulations to tend toward a low and constant asymptote when the number of nodes in the network increases. Moreover, compared to other clustering schemes as DDR [12] or Max-min d cluster [1], our cluster organization has revealed to be more stable over node mobility and arrivals and to offer better behaviors over non-uniform topologies (see [9]). Other interesting features for broadcasting obtained by simulations are gathered in Table 1. They are commented in Section 3.2.

	500 nodes	600 nodes	700 nodes	800 nodes	900 nodes	1000 nodes
# clusters/trees	11.76	11.51	11.45	11.32	11.02	10.80
Diameter $D(\mathcal{C})$	4.99	5.52	5.5	5.65	6.34	6.1
Cluster-head eccentricity	3.01	3.09	3.37	3.17	3.19	3.23
Tree depth	3.27	3.34	3.33	3.34	3.43	3.51
% leaves	73,48%	74,96%	76,14%	76,81%	77,71%	78,23%
Non-leaves'degree(in trees)	3.82	3.99	4.19	4.36	4.51	4.62
Voronoi: Euclidean distance	84.17%	84.52%	84.00%	83.97%	83.82%	83.70%
Voronoi: # of hops	85.43%	84.55%	84.15%	83.80%	83.75%	83.34%

Table 1. Some clusters and clustering trees characteristics.

3.2 Objectives

As explained earlier, our clustering heuristic leads at the same time to the formation of a spanning forest. We thus propose to use these *"clustering trees"* as a basis for the

broadcasting task. This broadcasting scheme is dominating set-based where the non-leaf nodes (internal nodes) belong to the dominating set. As mentioned in Table 1, a great proportion of nodes (about 75%) are actually leaves, therefore a broadcasting scheme based on this dominating set is expected to save many retransmissions. As the *clustering trees* form a spanning forest, the set of trees actually is a dominating set of the network but is not a connected dominating set as the trees are independent. So, to perform a reliable broadcasting task in the whole network, we need to connect these trees by electing some gateways. Our gateway selection is described in Section 4.

As already mentioned, each node only needs to know its 2-neighborhood to choose its parents, and to know whether it has been chosen as parent by one of its neighbors or it is a leaf. Thus, the forwarding decision of a non-gateway node is based on local state information. Only the gateway selection can be qualified of quasi-local (according to the classification of [17]) as only few nodes need information up to 4 hops away (tree depth). Thus, our broadcasting scheme does not induce a high costly maintenance. We propose to use this structure not only to perform a traditional broadcasting in the whole network but also for broadcasting in a cluster only. This kind of task might be interesting for clustered architectures when, for instance, a cluster-head needs to spread information only in its cluster like in sensor networks, for instance, where the base station may need to update devices or spread a query over them. The eccentricity of a node is the greater distance in number of hops between itself and any other node in its cluster. We can see in Table 1 that the tree depth is pretty low and close to the optimal we could expect which is the cluster-head eccentricity. This presents a good property for performing a broadcasting within our clusters. Indeed, none node is really far away from its cluster-head and can expect to receive quickly an information it would spread. Moreover, we computed the proportion of points closer to their cluster-head than any other one in Euclidean distance (Voronoi: Euclidean distance in Table 1) and in number of hops (Voronoi: # of hops in Table 1). Results show that a large part of nodes (more than 83%) lays in the Voronoi cell of their cluster-head whatever the process intensity. This characteristic is useful in terms of broadcasting efficiency as if the cluster-heads need to spread information over their own cluster, if most of the nodes are closer to the one which sends the information, we save bandwidth, energy and latency.

4 Our contribution

In this section, we first propose an algorithm for the gateway selection, then we detail the two kinds of broadcasting: within a cluster and in the whole network.

4.1 Gateway selection

A gateway between two neighboring clusters $\mathcal{C}(u)$ and $\mathcal{C}(v)$ actually is a pair of nodes $\langle x, y \rangle$ noted $\mathcal{G}ateway(\mathcal{C}(u), \mathcal{C}(v)) = \langle x, y \rangle$ such that $x \in \mathcal{C}(u)$, $y \in \mathcal{C}(v)$ and $x \in \Gamma_1(y)$. In such a pair, we will say that node x is the gateway node and that node y is the mirror-gateway node of the gateway. If $\mathcal{C}(u)$ and $\mathcal{C}(v)$ are two neighboring clusters, we note the gateway $\mathcal{G}ateway(\mathcal{C}(u), \mathcal{C}(v)) = \langle x = GW(\mathcal{C}(u), \mathcal{C}(v)), y = GWm(\mathcal{C}(u), \mathcal{C}(v)) \rangle$, where $GW(\mathcal{C}(u), \mathcal{C}(v))$ is the gateway

node and $GWm(\mathcal{C}(u), \mathcal{C}(v))$ is the mirror-gateway node. Note that $GW(\mathcal{C}(u), \mathcal{C}(v)) \in \mathcal{C}(u)$ and $GWm(\mathcal{C}(u), \mathcal{C}(v)) \in \mathcal{C}(v)$.

To select a gateway between two clusters, we thus need to define a pair of nodes. Our selection algorithm runs in two steps. The first step allows each frontier node to locally choose its "mirror(s)" in the neighboring cluster(s). We call a node u a frontier node if at least one of its neighbors does not belong to the same cluster than u. A frontier node and its mirror then form an eligible pair. The second step selects the most appropriate pair as the gateway. The algorithm tries to promote the selection of the nodes which already are internal nodes in order to minimize the size of the dominating set.

Mirror node selection. As seen in Section 3, as the density-based clustering algorithm uses the node Id as the last decision criterion, every node u might be aware in an expected bounded and low time, whether it exists among its neighbors a node v which does not belong to the same cluster than u. If so, node u is a frontier node and so is a possible gateway node for $\mathcal{Gateway}(\mathcal{C}(u), \mathcal{C}(v))$. Each frontier node u then selects its *mirror node* among its neighbors which do not belong to $\mathcal{C}(u)$. To do so, u first selects the non-leaf nodes, *i.e.*, the internal nodes/transmitters in every case and chooses among them the node with the highest density value. If every node $v \in \Gamma_1(u)$ such that $\mathcal{C}(u) \neq \mathcal{C}(v)$ is a leaf, u chooses the node with the lowest degree in order to limit the receptions induced by an emission of the mirror. In case of ties, the lowest Id decides. If u is a frontier node of the cluster $\mathcal{C}(v)$ ($\mathcal{C}(v) \neq \mathcal{C}(u)$), we note $m(u, \mathcal{C}(v))$ the mirror chosen by u in $\mathcal{C}(v)$. Note that if a node u is a frontier node for several clusters, it has to select a mirror for each cluster.

Algorithm 1 Mirror selection

For each frontier node u, *i.e.*, $\exists v \in \Gamma_1(u)$ s.t. $\mathcal{C}(v) \neq \mathcal{C}(u)$
    For each cluster $\mathcal{C}$ **for which** u **is a frontier node:** $\mathcal{C} \neq \mathcal{C}(u)$ **and** $\exists v \in \Gamma_1(u) \cap \mathcal{C}$**.**
        Select S the set of nodes such that $S = \mathcal{C} \cap \Gamma_1(u) \cap \{v \mid Ch(v) \neq \emptyset\}$.
        ▷ *u first selects the set of the non-leaf nodes as they are transmitters in every case.*
        if $(S \neq \emptyset)$ **then** Select S' the set of nodes such that $S' = \{v \mid v = max_{w \in S}\rho(w)\}$.
        ▷ *u collects internal nodes with the highest density in order to promote stability.*
        else ▷ *All the possibly mirrors of u are leaves.*
            $S = \{\mathcal{C} \cap \Gamma_1(u)\}$.
            Select S' the set of nodes such that $S' = \{v \mid v = min_{w \in S}\delta(w)\}$.
            ▷ *u collects the leaves with the lowest degree in order to minimize the receptions*
            induced by the addition of this node in the dominating set.
        end
        if $(S' = \{v\})$ **then** $m(u, \mathcal{C}) = v$.
        ▷ *There are no ties. S' contains only one node: the mirror of u.*
        else $m(u, \mathcal{C}) = v$ such that $Id(v) = min_{w \in S'}Id(w)$.
        ▷ *There are ties. u elects the node with the lowest Id.*
        end

Gateway selection. Once each frontier node has chosen its *mirror*, we have to choose the most appropriate pair as gateway. Once a gateway node u is elected as $GW(\mathcal{C}(u), \mathcal{C}(x))$, we have $\mathcal{Gateway}(\mathcal{C}(u), \mathcal{C}(v)) = \langle GW(\mathcal{C}(u), \mathcal{C}(x)), mirror(GW(\mathcal{C}(u), \mathcal{C}(x)))\rangle$. According to the taxonomy of [17],

this step is quasi-local unlike the first one which is local. The gateways are *directed* gateways in the meaning that two clusters are linked by two gateways $\mathcal{G}ateway(\mathcal{C}(u), \mathcal{C}(v))$ and $\mathcal{G}ateway(\mathcal{C}(v), \mathcal{C}(u))$ which may be different.

The gateway selection we propose is distributed *i.e.*, a selection is performed at every level in the tree and tries to limit useless receptions by favoring internal nodes. Frontier nodes send to their parent the following information: their Id, whether they are leaves and whether they have a leaf as mirror. Each parent selects the best candidate among its children and sends the same information up to its own parent and so on, up to reach the cluster-head. Thus, the selection is semi-distributed as every internal node eliminates some candidates. In this way, only small size packets are forwarded from the frontier nodes to the cluster-head. As mentioned in Table 1, the mean degree of all the internal nodes (cluster-head included) is small and constant whatever the number of nodes, which induce a small and bounded number of messages at each level which is also bounded by a low constant [11].

	500nodes	600nodes	700nodes	800nodes	900nodes	1000nodes
#clusters	11.93	11.64	11.36	11.30	11.14	10.72
#gw selected per cluster	5.86	6.02	6.16	6.20	6.22	6.26
#gw used per cluster	1.76	1.74	1.73	1.76	1.68	1.66

Table 2. Number of gateways selected and used per cluster for a global broadcasting initiated at a randomly-chosen source in function of the intensity process.

Let's express that v is in the subtree rooted in u (noted $v \in s\mathcal{T}(u)$) if u is the parent of node v or if the parent of node v is in the subtree rooted in u: $\{v \in s\mathcal{T}(u) \cap \Gamma_1(u)\} \Leftrightarrow \{v \in \mathcal{C}h(u)\}$ or $\{v \in s\mathcal{T}(u) \cap \bar{\Gamma}_1(u)\} \Leftrightarrow \{\mathcal{P}(v) \in s\mathcal{T}(u)\}$.

The best candidate choice is performed as follows. For each of the neighboring clusters $\mathcal{C}(x)$ of its subtree, an internal node u considers the set G of the candidate nodes (frontier nodes) ($G = \{v \in s\mathcal{T}(u) \mid \exists w \in \Gamma_1(v) \mid \mathcal{C}(w) \neq \mathcal{C}(u)\}$). Then, it selects among them the subset $G' \subset G$ of the internal nodes, still in order to limit the number of transmitter nodes. If G' is only composed of leaves, the selection is processed among all candidates of G. From there, u favors the nodes which mirror is a non-leaf node, then it uses either the density value (if remaining candidates are non-leaves) or degree (otherwise) to decide. At the end, in case of ties, the node with the lowest Id is elected as $GW(\mathcal{C}(u), \mathcal{C}(x))$. Note that, if $\mathcal{C}(u)$ and $\mathcal{C}(v)$ are two neighboring clusters, $\mathcal{G}ateway(\mathcal{C}(u), \mathcal{C}(v)) \neq \mathcal{G}ateway(\mathcal{C}(v), \mathcal{C}(u))$ in most cases.

Algorithm 2 Gateway selection

For each internal node u**:**
    For each cluster $\mathcal{C} \neq \mathcal{C}(u)$ **for which** $\exists v \in s\mathcal{T}(u)$ **which is a frontier node:**
        Gather the set G of candidate nodes: $G = \{v \in s\mathcal{T}(u) | \exists w \in \Gamma_1(v) \mid \mathcal{C}(w) = \mathcal{C}\}$.
        Select $G' \subset G$ the set of nodes v s.t. $G' = G \cap \{v | \mathcal{C}h(v) \neq \emptyset\}$.
        ▷ *u first selects the set of the non-leaf nodes as they are transmitters in every case.*
        if $(G' \neq \emptyset)$ **then**
        ▷ *There are non-leaf candidates. u will favor the ones with a non-leaf mirror and/or the highest density.*

Select $G" \subset G'$ the set of nodes s.t. $G" = G' \cap \{v|\mathcal{C}h(m(v,\mathcal{C}) \neq \emptyset\}$.
if $(G" \neq \emptyset)$ **then** Select $Finalist \subset G"$ the set of nodes s.t. $Finalist = \{v|\rho(v) = max_{w \in G"}\rho(w)\}$.
▷ *Internal Node↔Internal Node Gateway.*
else Select $Finalist \subset G"$ the set of nodes s.t. $Finalist = \{v|\rho(v) = max_{w \in G'}\rho(w)\}$.
▷ *Internal Node↔Leaf Gateway.*
end
else
▷ *All candidates are leaves. u will favor the ones with a non-leaf mirror and/or the smallest degree.*
Select $G" \subset G$ the set of nodes s.t. $G" = G \cap \{v|\mathcal{C}h(m(v,\mathcal{C}) \neq \emptyset\}$.
if $(G" \neq \emptyset)$ **then** Select $Finalist \subset G"$ the set of nodes s.t. $Finalist = \{v|\delta(v) = min_{w \in G"}\delta(w)\}$.
▷ *Leaf↔Internal Node Gateway.*
else Select $Finalist \subset G"$ the set of nodes s.t. $Finalist = \{v|\delta(v) = min_{w \in G'}\delta(w)\}$.
▷ *Leaf↔Leaf Gateway.*
end
end
if $(Finalist = \{v\})$ **then** $Winner = v$.
else $Winner = v|Id(v) = min_{w \in Finalist}Id(w)$.
▷ *There are ties. u elects the node with the lowest Id.*
end
if $u = \mathcal{H}(u)$ **then** $Winner$ becomes the final gateway node: $\mathcal{G}ateway(\mathcal{C}(u),\mathcal{C}) = \langle Winner, m(Winner,\mathcal{C}) \rangle$ **else** Send the $Winner$ identity to $nF(u)$.
end

4.2 Broadcasting heuristic

We need to be able to perform three kinds of broadcasting tasks: local (broadcast to 1-neighbors), clusters and global (broadcast to all the nodes in the network). Therefore, we need the use of a broadcasting address in the packet[1].

When the broadcasting task is performed within a cluster $\mathcal{C}(u)$, the message is forwarded upon first reception by all the non-leaf nodes of $\mathcal{C}(u)$. When the broadcasting task is performed in the whole network, all the non-leaf nodes in the network forward the message as well as the gateway and mirror-gateway nodes under some conditions. As our gateways are directed, a gateway node $GW(\mathcal{C}(u),\mathcal{C}(w))$ forwards the message only if it is coming from its own cluster $\mathcal{C}(u)$ and a mirror-gateway node $GWm(\mathcal{C}(u),\mathcal{C}(w))$ forwards it only if it is coming from the cluster $\mathcal{C}(u)$ for which it is a mirror. But a mirror-gateway node $GWm(\mathcal{C}(u),\mathcal{C}(w))$ forwards a message coming from $\mathcal{C}(u)$ whatever the transmitter node in $\mathcal{C}(w)$ which is not necessarily the gateway node $GW(\mathcal{C}(u),\mathcal{C}(w))$ for which it is a mirror.

Algorithm 3 Broadcasting algorithm

For all node u, upon reception of a broadcast packet P coming from node $v \in \Gamma_1(u)$
▷ *Node v is the previous hop and not necessary the broadcast source.*

---

[1] Such a broadcasting address is also used in IPv6.

if (u receives P for the first time)
    if *global broadcasting*
        if ($\mathcal{C}h(u) \neq \emptyset$) **then** Forward $\triangleright u$ *is an internal node.*
        else
            if ($\mathcal{C}(u) = \mathcal{C}(v)$ and $u = GW(\mathcal{C}(u), \mathcal{C}(w)) \forall w \in V$) **then** Forward
            $\triangleright u$ *is a gateway node and P is coming from its cluster.*
            end
            if ($\mathcal{C}(u) \neq \mathcal{C}(v)$ and $u = GWm(\mathcal{C}(v), \mathcal{C}(u))$) **then** Forward
            $\triangleright P$ *is coming from the cluster for which u is a mirror-gateway node.*
            end
        end
    else $\triangleright$ *It is question of a broadcasting within a cluster.*
    $\triangleright P$ *is forwarded only by internal nodes of the considered cluster.*
        if (($\mathcal{C}(v) = \mathcal{C}(u)$) and ($\mathcal{C}h(u) \neq \emptyset$)) **then** Forward **end**
    end

5 Simulation results

We performed simulations in order to qualify the gateways elected and to compare the broadcasting tasks performed with our clustering and gateway selection schemes. This section details the results. All the simulations follow the same model. We use a simulator we developed which assume an ideal MAC layer. Nodes are randomly deployed using a Poisson point process in a 1×1 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 0.1 in all tests. In each case, each statistic is the average over 1000 simulations. When several algorithms are compared, they are compared for each iteration over the same node distribution.

5.1 Gateways: election and utilization

If we consider two neighboring clusters $\mathcal{C}(u)$ and $\mathcal{C}(\text{v})$, we may have four different types of gateways between them:

- Leaf↔Leaf gateways: $GW(\mathcal{C}(u), \mathcal{C}(v))$ and $GWm(\mathcal{C}(u), \mathcal{C}(v))$ are both leaves. This kind of gateway is the more costly as it adds two transmitter nodes and thus induces more receptions.
- Leaf↔Internal Node gateways: $GW(\mathcal{C}(u), \mathcal{C}(v))$ is a leaf and $GWm(\mathcal{C}(u), \mathcal{C}(v))$ an internal node. This kind of gateway adds only one transmitter node. It's the less popular, as shown later by simulations.
- Internal Node↔Leaf gateways: $GW(\mathcal{C}(u), \mathcal{C}(v))$ is an internal node and $GWm(\mathcal{C}(u), \mathcal{C}(v))$ a leaf. This kind of gateway adds only one transmitter node.
- Internal Node↔Internal Node gateways: $GW(\mathcal{C}(u), \mathcal{C}(v))$ and $GWm(\mathcal{C}(u), \mathcal{C}(v))$ are both internal nodes. This kind of gateway is the one we try to favor since it does not add extra-cost at all as it does not add any transmitter neither induces any additional receptions. But, as we will see, they unfortunately are the less popular ones.

Table 2 shows the mean number of gateways a cluster has to elect and maintain and the mean number of gateways used when a global broadcasting task is performed. As we can note, the number of gateways to elect is reasonable and remains almost constant while process intensity increases. This shows a scalability feature of this heuristic. Nevertheless, this was predictable since, as we saw in Section 3, the number of clusters is constant from a certain amount of nodes, so is the mean number of neighboring clusters for a cluster and so the number of gateways to elect. Moreover, we also saw that the cluster topology was close to a Voronoi tessellation. Yet in a Voronoi tessellation, a cell has 6 neighbors in average. This is actually the mean number of gateways a cluster has to elect. Figure 1(a) gives the proportion of each kind of gateways selected. We can note that the two less elected kinds of gateways are the Leaf↔Internal Node and Internal Node↔Internal Node gateways. This is due to the fact that, by construction, most of frontier nodes are leaves. This also explains the great proportion of other kinds since, as soon as there is an internal node as a frontier node, it is elected (and thus Internal Node↔Leaf gateways are preferred to Leaf↔Internal Node ones). The more sparse the network, the less chance to find internal nodes on borders. So, the proportion of Internal node↔Leaf gateways increases with the intensity process while the proportion of Leaf↔Leaf gateways decreases.

When a global broadcasting task is performed, all gateways are not necessary used since 2 neighboring clusters $\mathcal{C}(u)$ and $\mathcal{C}(v)$ are connected via 2 gateways $GW(\mathcal{C}(u), \mathcal{C}(v))$ and $GW(\mathcal{C}(v), \mathcal{C}(u))$ and in most cases, only one of them will be used. As Table 2 shows, the number of gateways used per cluster is quite constant and remains pretty low, always comprised between 1 and 2. This means that generally, either the broadcast enters a cluster and dies in it (in this case, it uses only one gateway), either it crosses it and thus uses two gateways (one to enter the cluster, one to leave it).

Figure 1(b) shows the proportion of each kind of gateways used when a global broadcasting operation is initiated. As we can see, most of them are the ones which add only one transmitter node. This is true even for low intensities of node when the number of Leaf↔Leaf gateways elected was the highest. This shows that our algorithm can adapt and favor internal nodes naturally. As the mean number of gateways used is low and that for each gateway, we add only one transmitter node, the induced cost is low as well.

5.2 Broadcasting performances

In order to evaluate our algorithm, we chose to compare it to representative broadcasting schemes seen in Section 2: blind flooding, LCA [8] (cluster-based schemes), Multi-Point Relay [13] (source-dependent dominating set) and Neighbors Elimination-Based [16] (source-independent dominating set). The significant characteristics we note are the proportion of nodes which need to re-emit the message (size of the dominating set), the mean number of copies of the broadcast message that a node receives (useless receptions) and the latency (time needed for the last node to receive the broadcast packet initiated at the source). As the main goal is to limit energy consumption and bandwidth occupation in order to maximize the network lifetime, all these values have to be as low as possible.
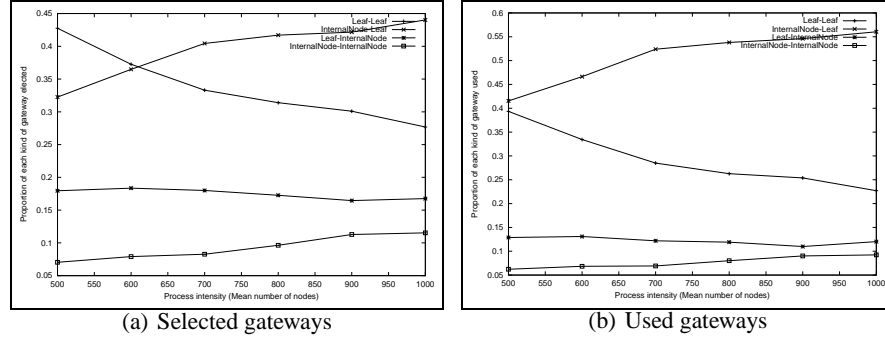
(a) Selected gateways

(b) Used gateways

Fig. 1. Proportion of each kind of selected gateways and used gateways per cluster as a function of the process intensity.(+: Leaf↔Leaf; ×: Internal Node↔Leaf; ∗: Leaf↔Internal Node; □: Internal Node↔Internal Node;)

Broadcasting over the whole network: global broadcasting. The broadcasting task is initiated at a randomly-chosen source over the whole network.
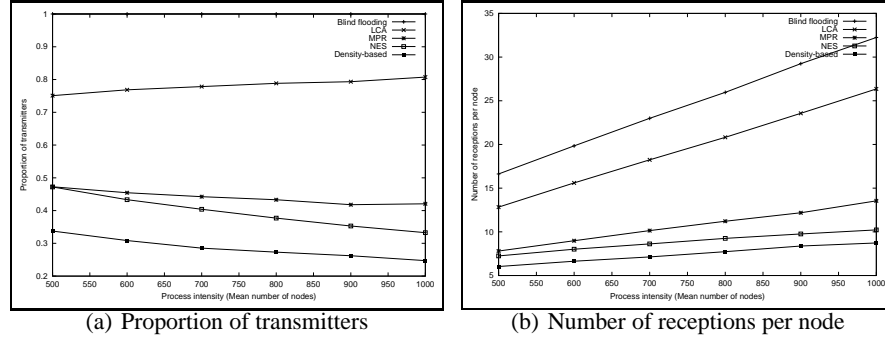

(a) Proportion of transmitters

(b) Number of receptions per node

Fig. 2. Proportion of transmitters *(a)* and Mean number of receptions per node *(b)w.r.t* the different algorithms and the mean number of nodes.(+: Blind Flooding; ×: LCA; ∗: MPR; □: NES; ■ Density-based;)

As Figure 2 plots, when a global broadcasting task is initiated, our algorithm induces less re-transmissions (Figure 2(a)) and less receptions(Figure 2(b)) than other algorithms. Thus, it spends less energy and resources.

Since in the MPR selection, the relays are selected in order to reach the 2-neighborhood after two hops, the k-neighborhood of the source is reached within k hops. Under the assumption of an ideal MAC layer, MPR gives the optimal results in terms of latency (Number of hops). We thus compare our heuristic to the MPR one to measure how far we are from the optimal solution. We consider a time unit as a transmission step (*i.e.*, 1 hop). Table 3 presents the results. Yet, we can note that, even if our algorithm is not optimal regarding the latency, results are very close to it. Figure 3 represents the propagation in time for a broadcast packet initiated at the centered source at time 0. Cluster-heads appear in blue and the source in green. The color of other nodes depends on the time they receive the broadcast packet. The darker the color, the shorter the time.
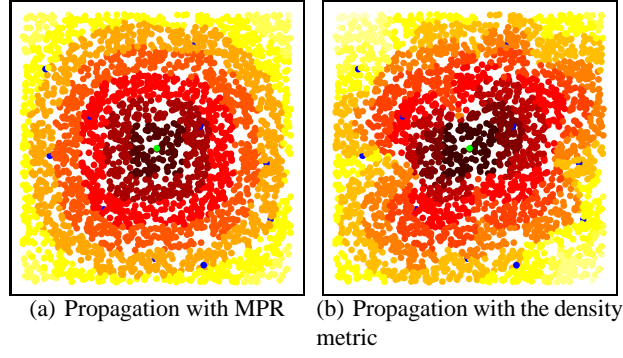
(a) Propagation with MPR   (b) Propagation with the density metric

Fig. 3. Propagation time of a general broadcasting initiated at a centered source (a) using MPR (b) using density-based clustering trees.

	500 nodes		700 nodes		800 nodes		900 nodes		1000 nodes	
	MEAN	MAX	MEAN	MAX	MEAN	MAX	MEAN	MAX	MEAN	MAX
MPR	5.13	8.97	4.88	8.40	4.88	8.40	4.81	8.23	4.78	8.07
Density	6.31	11.05	6.22	10.78	6.24	10.95	6.15	10.66	6.19	10.74

Table 3. Mean and Max time for receiving the message. *"MAX" values represent the time needed for the last node to receive the packet. "MEAN" values represent the mean time a node needed for a node to receive the broadcast packet.*

Broadcasting within clusters: cluster broadcasting. We now suppose that the broadcasting task is performed in each cluster, initiated at the cluster-heads. We thus have as many broadcastings as clusters.

We can see on Figure 4, that our broadcasting algorithm still obtains the best results regarding the number of receptions (Figure 4(a)) and the transmitter ratio (Figure 4(b)). Table 4 and Figure 5 present the results regarding the latency. Once again, we can observe that, even if our algorithm is not optimal, results are very close as, in average, a node in our algorithm needs only 0.5 steps more than the optimal value to receive the packet. This also shows that, even if the routes in the trees from the cluster-heads to other nodes are not always the shortest ones, they are very close to them.

	500 nodes		700 nodes		800 nodes		900 nodes		1000 nodes	
	MEAN	MAX	MEAN	MAX	MEAN	MAX	MEAN	MAX	MEAN	MAX
MPR	1.76	4.71	1.78	4.85	1.81	4.83	1.81	4.80	1.82	5.00
Density	1.80	5.08	1.83	5.38	1.87	5.29	1.87	5.50	1.88	5.30

Table 4. Mean and Max time for receiving a cluster broadcast message. *"MAX" values represent the time needed for every node to receive the packet at least once, "MEAN" values the mean time a node has to wait till the first reception of the packet.*

In Table 5, we give the proportion of nodes which receive the first copy of the packet by their parent or by one of their children. This feature shows whether the message which is sent by the cluster-head follows the branches of the trees. Indeed, a node u always receives the message by its parent (as all non-leaf nodes forward the message) but it could have received it before from another way as paths are not always optimal. In this case, the shorter route between u and its cluster-head is not found by following

(a) Proportion of transmitters   (b) Number of receptions
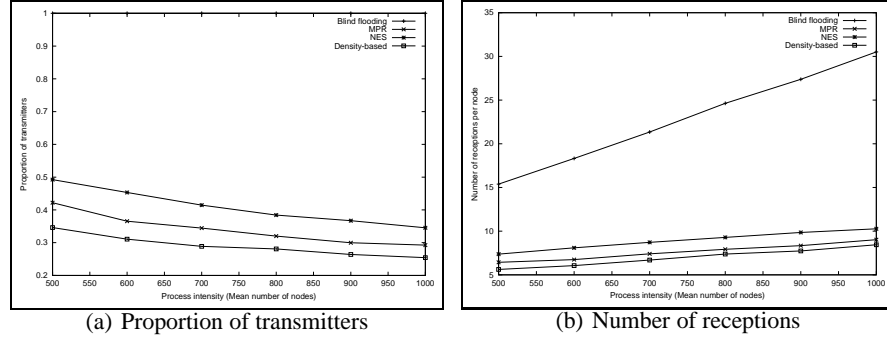
Fig. 4. Mean number of receptions per node (a) and Proportion of transmitter nodes (b) for a cluster broadcasting scheme *w.r.t.* the process intensity and the metric used.(+: Blind Flooding; ×: MPR; ∗: NES; □: Density-based;)

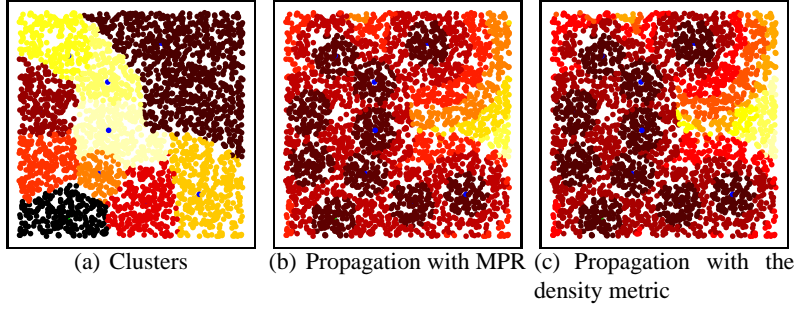(a) Clusters   (b) Propagation with MPR   (c) Propagation with the density metric

Fig. 5. Propagation time of a cluster broadcast packet over the topology plotted in *(a)* using MPR *(b)* or density-based trees *(c)*.

the route in the tree. We can thus see that routes are the shortest ones in number of hops for more than 70% of the cases. We can also observe that, as the message progresses down the tree, none of the nodes receives it the first by one of their children.

	500nodes	600nodes	700nodes	800nodes	900nodes	1000nodes
% by parent	78.74%	76.81%	74.57%	73.21%	71.31%	70.13%
% by a child	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 5. Proportion of nodes which receive the first copy of the packet by their parent or by one of their children.

6 Conclusion and perspectives

We have proposed a broadcasting scheme over a clustered topology for multi-hop wireless networks. We can thus obtain a double-use structure with low cost, as the maintenance is local and quasi-local. The cost is bounded by the tree depth which is a low constant. Moreover, two kinds of broadcasting task may be performed over it since

we can perform global broadcastings in the whole network as well as broadcastings confined inside a cluster. Our proposed algorithm offers better results than some existing broadcasting schemes. More precisely, it saves more retransmissions since the number of internal nodes selected is lower (for both global and cluster-confined broadcastings). Moreover, the number of duplicated packets received is also lower. Note that reducing both emissions and receptions is an important factor when designing energy aware broadcasting protocols. Future works will be dedicated to investigate robustness of clustering-based broadcasting protocol in presence of node and link failures. Preliminary results tend to show that the confined broadcasting is more robust that the global one and thus we investigate more deeply the impact of the choice and number of gateways between clusters.

References

1. A. Amis, R. Prakash, T. Vuong, and D. Huynh. Max-Min D-cluster formation in wireless ad hoc networks. In *Proceedings of the IEEE INFOCOM*, Tel-Aviv, Isral, March 2000. IEEE.
2. J. Cartigny, F. Ingelrest, and D. Simplot-Ryl. RNG relay subset flooding protocol in mobile ad-hoc networks. *IJFCS*, pages 253–265, 2003.
3. G. Chen, F. Garcia, J. Solano, and I. Stojmenovic. Connectivity-based k-hop clustering in wireless networks. In *HICSS'02*, Hawaii, USA, January 2002.
4. C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. In *ICCS/ISPACS'96*, Singapore, November 1996.
5. A. Ephremides, J. Wieselthier, and D. Baker. A design concept for reliable mobile radio networks with frequency hoping signaling. In *IEEE 75*, pages 56–73, 1987.
6. Y. Fernandess and D. Malkhi. k-clustering in wireless ad hoc networks. In *POMC*, Toulouse, France, 2002.
7. H. Lim and C. Kim. Multicast tree construction and flooding in wireless ad hoc networks. In *ACM MSWiM Workshop at MobiCom 2000*, Boston, MA, USA, August 2000.
8. C. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE JSAC*, 15(7):1265–1275, 1997.
9. N. Mitton, A. Busson, and E. Fleury. Self-organization in large scale ad hoc networks. In *MED-HOC-NET 04*, Bodrum, Turkey, June 2004.
10. N. Mitton, A. Busson, and E. Fleury. Broadcast in self-organizing wireless multi-hop network. Research report RR-5487, INRIA, February 2005.
11. N. Mitton, E. Fleury, I. Guérin-Lassous, and S. Tixeuil. Self-stabilization in self-organized multihop wireless networks. In *WWAN'05*, Columbus, Ohio, USA, June 2005.
12. N. Nikaein, H. Labiod, and C. Bonnet. DDR-distributed dynamic routing algorithm for mobile ad hoc networks. In *MobiHoc*, Boston, MA, USA, November, 20th 2000.
13. A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *HICSS'02*, Hawaii, USA, January 2002.
14. I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination-based broadcasting algortithms in wireless networks. *IEEE TPDS*, 13(1), January 2002.
15. I. Stojmenovic and J. Wu. Broadcasting and activity scheduling in ad hoc networks. *IEEE Mobile Ad Hoc Networking*, pages 205–229, 2004.
16. J. Wu and H. Li. A dominating set based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, pages 13–36, 2001.
17. J. Wu and W. Lou. Forward node set based broadcast in clustered mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 3(2):141–154, 2003.