Alternating MPR: a balanced broadcast algorithm for MANETs

Aymeric Agon-Rambosson, Jonathan Lejeune, Julien Sopena and Pierre Sens
Sorbonne Université, CNRS, LIP6, DELYS
F-75005 Paris, France
firstname.lastname@lip6.fr

***Abstract*—Mobile Ad-Hoc Networks (MANETs) assume no previous network infrastructure and wireless communication between mobile and heterogeneous nodes. An efficient broadcast protocol is therefore paramount. When some neighborhood information is available beforehand through discovery, building a virtual overlay like MultiPoint Relay (MPR) can help improve reliability and decrease cost in messages. However, MPR overlays tend to unfairly stress specific nodes who happen to be well-connected, causing their premature death. We propose the alternating MPR protocol that strives to build several disjoint relay sets for each node, allowing broadcast messages to use each of them in turn. Our simulation of the full network stack of systems of various densities shows that alternating MPR spreads energy costs more evenly across the system, without harming reliability and at little cost in number of messages, allowing battery-powered nodes to survive longer.**

***Index Terms*—MANET, Broadcast, MPR, Energy fairness**

I. Introduction

A Mobile Ad-Hoc Network (MANET) consists of potentially heterogeneous mobile nodes communicating wirelessly directly with each other. This network paradigm developed at the end of the 1990s can accurately model edge computing environments, meshnets, wireless sensor networks, as well as networking situations consistent with IoT applications.

Since this paradigm assumes no preexisting network infrastructure, the nodes have to act as relays for packets not intended for them. Wireless communications are subject to faults, particularly collisions: a node situated inside the intersection of the covered areas of two other nodes transmitting at the same time will receive neither message correctly. In the case of wireless communications, these collisions are impossible to detect [1], [2], impractical to avoid [1], [2], and costly to address. This problem is particularly acute for the broadcast operation, both primordial in MANETs (efficient routing algorithms like [3] rely on a broadcast primitive) and very likely to cause collisions, since transmissions are synchronized. The most straightforward broadcast algorithm, *flooding*, makes every node relay each message at its first reception. *Flooding* is very likely, in dense environments, to cause the "broadcast storm" problem [4] because of too many simultaneous retransmissions, thus increasing the cost in messages and decreasing reliability.

This paper is funded by the French ANR project SeMaFoR

979-8-3503-9730-7/22/$31.00 ©2022 IEEE

By restricting the burden of retransmissions to a subset of the nodes, as proposed by [5], we can save messages and avoid collisions while preserving the cover rate of the broadcast. This means however that the burden of retransmissions is unevenly shared, and that the best connected nodes deplete their battery much quicker than the rest.

Contributions. We address this problem by proposing a broadcast primitive that is at the same time *effective* (delivering the message to the most nodes possible), *efficient* (causing as little retransmissions as possible) and *energy-fair* (sharing the burden of retransmissions as evenly as possible). We propose the novel ***alternating MultiPoint Relay (MPR)*** algorithm, in which several sets of relays are built, and they are used each in turn when a message is to be sent or retransmitted. We target the *dense* and *static* areas of generic MANETs, and we will show why this restriction does not imply loss of generality.

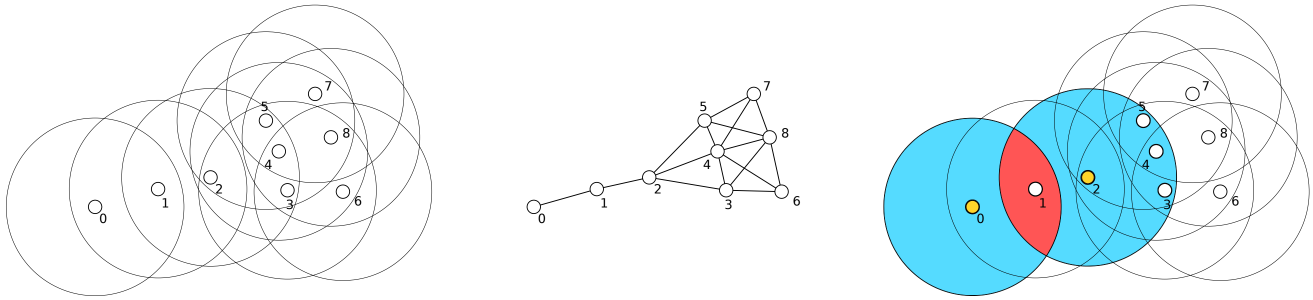
This paper is organized as follows. We present our hypotheses regarding the system and the model of communications (§II-A). Then we broadly recall the functioning of the 802.11 MAC protocol (§II-B) and the formal specification of the broadcast operation for MANETs (§II-C). We then briefly present *conditional flooding* and *overlay-based* algorithms from the literature (§III). We more specifically recall the functioning of static MPR (§IV-A), and particularly the relay selection process. We present our improvement, *alternating MPR* (§IV-B), by detailing a novel way of selecting relays. We evaluate the impact of graph topology on the effectiveness of our improvement (§V), and we present a quantitative evaluation of it on simulated graphs using a full network stack simulation in OMNeT++/INET [6] (§VI) and we conclude (§VII).

II. Model and background

We start by specifying the system and communication models. We then describe the MAC 802.11 protocol and the specification of the broadcast protocol for MANETs.

A. System and communication models

We consider a static set N of uniquely identifiable nodes $n_0, n_1, \ldots, n_{N-1}$ in a 2-dimensional square. We assume no availability of GPS information on the devices. Nodes communicate via omnidirectional wireless transceivers with fixed transmission T and reception R ranges that are identical and large enough to guarantee network connectivity. The *coverage*



(a) Coverage areas (b) Graph representation (c) Collisions

Fig. 1. Because of constant transmission range, the system can be represented as an undirected graph.

area of a node is therefore given by the circle centered on the node of radius T (Fig. 1a). When the node emits a frame, all other nodes within that circle receive the signal with a probability 1, while all nodes outside of the circle receive the signal with probability 0. Hence, the cost to transmit a message to one's immediate *neighborhood* (the nodes within transmission range) is independent of the size of that neighborhood. Because of physical limitations of radiotransmitters, nodes are not able to send and receive simultaneously. The system can therefore be modeled as a *geometric graph*: whenever a node u is part of the neighborhood of node v, then the opposite is also true, and this relationship can be represented by an undirected edge connecting u and v (Fig. 1b). Now, whenever a node sends a message, all *adjacent* nodes in the graph can be considered to have received it, with one notable exception: as mentioned earlier, a node located in the intersection of the coverage areas of two distinct nodes that transmit a frame simultaneously will receive neither signal correctly: this phenomenon is called *collision* (Fig. 1c). Collisions are impossible to detect by the sender [2], and must therefore be avoided by some Medium Access Control protocol.

B. IEEE 802.11 MAC protocol

The role of the MAC layer of the IEEE 802.11 protocol is to schedule the use of the wireless medium (in our case, the carrier radio wave) in order to avoid collisions. In this protocol, nodes that wish to transmit a frame must first listen to the carrier to check that no other node is already transmitting. More specifically, the node creates a random counter, and decrements it each time it senses that there has been no signal for a specific period of time called DIFS (*DCF InterFrame Spacing*). The frame is sent when the counter reaches 0. The protocol optionally allows the nodes that receive the frame correctly to immediately send an acknowledgement frame. This optional part of the protocol is obviously disabled when the frame is broadcast, as all neighbors would send the acknowledgement frame at the same time, effectively nullifying the chances of the sending node to receive any of these correctly.

C. MANET broadcast specification

The goal of a MANET broadcast protocol is to send a message from a *source* node to the biggest amount of reachable nodes in the network, despite mobility and collisions. Any node in the system can send a message, and a message may have to be split into multiple MAC frames. Because of the absence of previous infrastructure, nodes in the network may forward messages upon the first reception, and therefore act as *relays*. Following receptions of the same message (*duplicates*) are simply discarded (which supposes the maintaining in each node of a moving memory of already received messages).

The goal is to achieve a good *coverage* (i.e. proportion of nodes that have delivered the message) while minimizing the number of *relays* in the network (and therefore the *retransmission rate*), in order to decrease energy costs and avoid collisions. This problem can be assimilated to the computing of a minimal *connected dominating set* over the graph representing the system: the connected dominating set is such that if every node in the set retransmits, we are assured to have full coverage (in the absence of collisions). However, the nodes are assumed to have no previous information whatsoever on the network.

III. Broadcast protocols and related works

This section briefly reviews some MANET broadcast algorithms from the literature. The most straightforward broadcast algorithm, *flooding*, simply considers every node as a relay, and the broadcast algorithms proposed during the last twenty years have tried to improve upon flooding by finding various methods to reduce the number of relays [7], [8]. Two main categories have been proposed by [8], namely *conditional flooding* and *overlay-based*.

A. Conditional Flooding

Unconditional flooding is the simplest MANET broadcast protocol in which every node acts as a relay. Hence, every first reception of a message causes a retransmission, which means a large number of redundant duplicates, and many potentially overlapping transmissions in dense areas. The idea behind conditional flooding is to allow nodes not to retransmit when it can be reasonably inferred that their neighborhood has been sufficiently covered and that their own retransmission would not bring about any gain in coverage. Hence, algorithms add various conditions on the retransmission: for instance, in [9] the authors would condition retransmission on a probability either constant in the whole system, or dependent on the local node density (assumed to be obtainable via regular HELLO messages). In dense parts of the network, the retransmission probability would be low: the expected amount of retransmissions would still be high enough to guarantee coverage, but not too high as not to cause collisions. In sparse parts of the network, the retransmission probability would need to be high, to maintain a high enough level of expected retransmissions.

Another way to estimate local node density is to use a counter of the received duplicates of the same message in a specific time interval, as in [10] or [11]. These solutions have the advantage of not needing extra HELLO messages, but increase latency of retransmissions. There is also the possibility to estimate local node density using the distance from the sender, as in [12].

Despite the conditioning of the retransmission, these algorithms remain costly, and determining the parameters of the condition is difficult : the optimal value of the parameter (probability, counter value, local density threshold) heavily depends on the network topology. On the other hand, conditional flooding is good in regions with high mobility: the decision to retransmit is based solely on information obtained at the time of the dissemination, and is therefore quite accurate and up to date.

B. *Overlay-based algorithms*

The other main class of broadcast algorithms strives to precompute an *overlay* on the graph, i.e a *connected dominant subgraph* of relays : only the vertices contained in the subgraph would retransmit, effectively forming a persistent dissemination backbone. This subgraph would need to be connected and dominant to ensure total coverage of the graph, but would contain as few relays as possible. Computing such a subgraph requires some graph information, such as the set of vertices and the edges connecting them. However, because the nodes are assumed to have no information whatsoever on the graph, this overlay will have to be built in a decentralized manner, using only local information.

One possible decentralized manner to build this subgraph is for every vertex to build it on the graph representing the connectivity information of its neighborhood, that is the set of vertices situated at a specific *geodesic distance* (i.e. length of a *shortest* path). For instance, a vertex can obtain this connectivity information over its 2-hop neighborhood by regularly sending and receiving HELLO messages containing his own identifier and the list of its neighbors. After some time, each vertex would be capable of maintaining a list of its 1-hop neighbors, and for each of those, a list of 2-hop neighbors connected to it.

Using this information, it becomes possible for each node to precompute a set of relays in its own 1-neighborhood, and when the time comes to send or retransmit a message, to designate explicitly these relay nodes in the header of the message. The different *overlay-based* algorithms differ according to the neighborhood information they use (1-, 2-, and more rarely 3-hop neighborhood information), the invariant constraint put on the relay set [1], and the algorithm used to compute the relay set.

Overlay-based algorithm will outperform conditional flooding in dense and stable MANETs: they have access to complete neighborhood information to build the smallest possible relay set, whereas conditional flooding algorithms can only rely on proxy information like local density. However, they will underperform in sparse or dynamic MANETs: in sparse MANETs where the set of relays is going to be nearly equal to the set of neighbors, the extra cost of HELLO messages will not be offset by the gains caused by fewer relays, and in dynamic MANETs the dynamicity will either require more frequent HELLO messages to keep the neighborhood information up to date.

In fact, as established by [15], it is very difficult to propose an efficient algorithm in both sparse/dynamic and dense/stable MANETs. This difficulty has led [16] to propose an interoperability protocol allowing one algorithm or the other to be executed according to the local characteristics of the network, and moving from one to the other in a coordinated manner. For this reason, an algorithm like MultiPoint Relay (MPR), proposed in [5], effectively assuming dense and static environments, can be efficient in a general MANET setting if it could always be associated with a conditional flooding algorithm for the sparser and more dynamic parts of the network. It uses a greedy algorithm to build, from the 1-hop neighbors of a specific node, a set of relays sufficient to cover all of the 2-hop neighbors of the node. The set is not guaranteed to be minimal (finding the minimal sufficient set would be a NP-hard problem).

In the MPR approach, the relay set is however deterministic: a specific 2-hop neighborhood will always produce the same set, which means that a sequence of messages transmitted by the source node will always be retransmitted by the same 1-hop neighbors. The energy burden of retransmissions is therefore unevenly shared, which causes the best-connected nodes to die sooner (since they tend to be selected in the relay set). The dilemma posed by MPR versus conditional flooding is the following: choosing MPR allows retransmission saving and improves reliability, but always stresses the few same nodes. Choosing conditional flooding shares the burden of retransmissions evenly, but at the cost of more retransmissions and potentially more collisions. We propose to keep the best of both worlds with alternating MPR: we achieve fewer retransmissions and better reliability than flooding, while sharing more evenly the energy costs of retransmitting than MPR.

[1] In [5] and [13], the relay set ensures complete coverage of the 2-hop neighborhood, while in [14], the authors are more concerned about routing and want to ensure that the relay sets form a spanning forest over the graph.

IV. ALGORITHM

We recall the functioning of relay selection in static MPR. Both static and alternating MPR assume that each node knows its 2-hop neighborhood, that is the list of its 1-hop neighbors, of its 2-hop neighbors and all the outgoing connections of every 1-hop neighbor. The goal is to ensure total coverage of the 2-hop neighborhood (every 2-hop neighbor is covered by at least one relay).

A. Static MPR

Algorithm 1: Relay selection, static MPR

Input:
N_1 : *set* (1-hop neighborhood)
$N_2[n_j \in N_1] \mapsto \{n_k, \dots\}$: *map of sets* (2-hop neighborhood)

1 *local variable* $U \longleftarrow \bigcup_{n_j \in N_1} N_2[n_j]$: *set* (nodes to cover);
2 *local variable* $R \longleftarrow \emptyset$: *set* (relays);
3 **forall** $n_j \in N_1$ **do**
4   **if** $\exists n_{iso} \in N_2[n_j] \mid \forall n_{k \neq j}, n_{iso} \notin N_2[n_k]$ **then**
5         $R \longleftarrow R \cup \{n_j\}$;
6         $U \longleftarrow U \setminus N_2[n_j]$;
7 **while** $U \neq \emptyset$ **do**
8   select $n_j \in N_1$ maximizing $|N_2[n_j] \cap U|$;
9     $R \longleftarrow R \cup \{n_j\}$;
10     $U \longleftarrow U \setminus N_2[n_j]$;
11 **return** R;

Algorithm 2: Relay selection, alternating MPR

Input:
N_1 : *set* (1-hop neighborhood)
$N_2[n_j \in N_1] \mapsto \{n_k, \dots\}$: *map of sets* (2-hop neighborhood)

1 *local variable* $U \longleftarrow \bigcup_{n_j \in N_1} N_2[n_j]$: *set* (nodes to cover);
2 *local variable* $R \longleftarrow []$: *list of sets* (relays);
3 *local variable* $F \longleftarrow \emptyset$: *set* (forbidden relays);
4 *local variable* $C \longleftarrow \emptyset$: *set* (current relays);
5 *local variable* $i \longleftarrow 0$: *scalar* (relay set index);
6 **repeat**
7   **forall** $n_j \in N_1$ **do**
8   **if** $\exists n_{iso} \in N_2[n_j] \mid \forall n_{k \neq j}, n_{iso} \notin N_2[n_k]$ **then**
9             $C \longleftarrow C \cup \{n_j\}$;
10             $U \longleftarrow U \setminus N_2[n_j]$;
11   **while** $U \neq \emptyset$ **do**
12   select $n_j \in N_1 \cap \overline{F}$ maximizing $|N_2[n_j] \cap U|$;
13   **if** $N_1 \cap \overline{F} = \emptyset$ **then**
14   **return** R
15         $C \longleftarrow C \cup \{n_j\}$;
16         $F \longleftarrow F \cup \{n_j\}$;
17         $U \longleftarrow U \setminus N_2[n_j]$;
18     $R[i] \longleftarrow C$;
19     $C \longleftarrow \emptyset$;
20     $U \longleftarrow \bigcup_{n_j \in N_1} N_2[n_j]$;
21     $i \longleftarrow i + 1$;
22 **until** *true*;

The relay set is built in the following manner (Algorithm 1): first, the *necessary* relays are added to the relay set (lines 3 to 6). A relay is necessary when it is the only one providing cover to a nonempty set of 2-hop neighbors: if it wasn't included, the relay set couldn't possibly be complete. All 2-hop neighbors covered by these necessary relays are pruned from the set of 2-hop neighbors left to cover (line 6).

Then, a *sufficient* relay set is calculated (lines 7 to 10): the 1-hop neighbor that covers the most yet uncovered 2-hop neighbors (line 8) is added to the relay set (line 9), and the 2-hop neighbors thus covered are pruned (line 10). The algorithm stops when there isn't any 2-hop neighbor left to cover. We know the algorithm eventually stops, because every 2-hop neighbor is covered by at least one 1-hop neighbor. In the worst case scenario, all the 1-hop neighbors are included in the relay set.

The relay set returned by this algorithm is deterministic: given a specific node and its 2-hop neighborhood, the same relay set will be chosen over and over. Therefore, given a sequence of sent or retransmitted messages, all the messages will be retransmitted by the same relays (if we assume a static 2-hop neighborhood), which will deplete their battery, while the remainder of the 1-hop neighbors will be idle.

B. Alternating MPR

In order to address this problem, we propose in *Alternating MPR* (Algorithm 2) building several relay sets (lines 2 and 5). These relay sets will be disjoint except for the necessary relays, which will have to be in every relay set. In order to do this, several local variables are introduced, namely a "forbidden" relay set (line 3) and a temporary variable to store the relay set being built (line 4). The algorithm is an infinite loop (lines 6 to 22) of relay set building, which ends as soon as a relay set cannot be built (line 14). The building of the current relay set begins with the inclusion of every necessary relay (lines 7 to 10) and continues with the inclusion of a sufficient set (lines 11 to 17). This time, the relays added according to the greedy criterion (line 12) are also added to the forbidden set (line 16), which guarantees that they won't be used in a future relay set. Once the current relay set is sufficient, it is added to the set of relay sets (line 18), the current relay set (line 19) and the set of 2-hop neighbors to cover (line 20) are reset, before starting up again. The algorithm returns at least the relay set computed by static MPR, and eventually terminates: the amount of relay sets that we can build is necessarily finite. By design, the same non-necessary relay will not be present in more than one relay set.

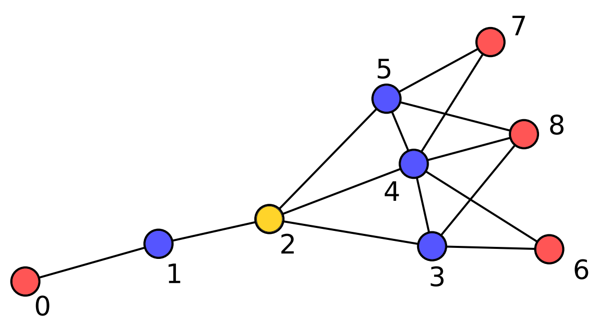


Fig. 2. Algorithm walkthrough

In order to better understand the progress of the algorithm, we will run it on Fig. 2, which represents the knowledge node 2 has about its 2-hop neighborhood: $\{1, 3, 4, 5\}$ is its 1-hop neighborhood, 0 is a 2-hop neighbor via $\{1\}$, 7 is a 2-hop neighbor via $\{5, 4\}$, 6 via $\{3, 4\}$, and 8 via $\{3, 4, 5\}$. Node 2 starts the algorithm with no relay sets, an empty forbidden relay set and a set of nodes to cover initialized to its 2-hop neighborhood. It attempts to build its first relay set by identifying the isolated 2-hop neighbors, in our case 0. It therefore identifies 1 as a necessary relay, and includes it in its current relay set. Next it computes a sufficient set: nodes 6, 7 and 8 need to be covered. The 1-hop neighbor with the best adjacency in this set is 4. It is included in both the current relay set and the forbidden relay set. The set of nodes left to cover is empty, so $\{1, 4\}$ is a sufficient relay set. Static MPR would terminate right here. Alternating MPR attempts to build as many relay sets as possible, node 2 will therefore try to build another one. The set of nodes to cover is still $\{0, 6, 7, 8\}$, and the set of forbidden relays is now $\{4\}$. Node 1 is still included in the relay set because it is necessary, and we move on to the construction of the sufficient set: 5 and 3 are ex aequo in adjacency in the set $\{7, 8, 6\}$. Node 3 is then chosen and added to both the current relay set and the forbidden set, and 6 and 8 are now covered. The only node capable of covering 7 is 5, and will therefore be included in the relay set and the forbidden set. $\{1, 3, 5\}$ constitutes a second sufficient relay set. Any further attempt to build a third relay set would necessarily fail, as the forbidden set is now $\{3, 4, 5\}$ and no other node would be capable of covering either 6, 7 or 8. The algorithm therefore terminates with the sets $\{1, 4\}$ and $\{1, 3, 5\}$ as illustrated on Fig. 3.

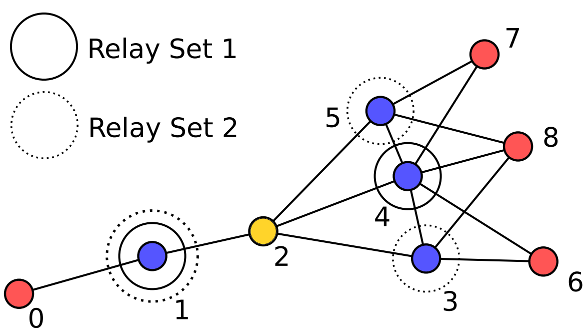


Fig. 3. Final algorithm state

V. Impact of graph topology

The number of relay sets alternating MPR is able to build, and their respective size, depends on the structural properties of the 2-hop neighborhood of the node. Nonetheless, because the "best" relays will be included in the first sets, we can expect the size of relay sets to increase the more we build. An algorithm using each of these sets in turn (round-robin) would probably generate more messages than one using only the first set (which would be the original static version of MPR). On the other hand, we can legitimately expect the repartition of retransmissions to be more balanced in the alternating version.

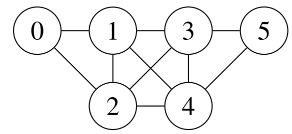


Fig. 4. Double corridor topology

Let us illustrate this using the specific graph configuration shown by Fig. 4. In the case of static MPR, a broadcast originating from node 0 would systematically be forwarded by the nodes 1 and 3 (if we assume a deterministic method of breaking possible ties at line 8 of Algorithm 1). In the case of alternating MPR with a round-robin between relay sets, every even broadcast would be forwarded by the same nodes as static MPR, but every odd one by the nodes 2 and 4. Thus, over a series of broadcasts originating from 0, the energy cost of the retransmission would be spread evenly across all nodes 1, 2, 3 and 4, rather than only on the nodes 1 and 3.

However, in our previous example from Fig. 2, for each broadcast originating from node 2, we would need a retransmission from node 1 to achieve complete coverage, because node 1 is *necessary*: it is the only 1-hop neighbor of 2 that is also a 1-hop neighbor of 0. Then, **any algorithm** that guarantees complete coverage of its 2-hop neighborhood (in particular, any version of MPR) creates among its 1-hop neighbors a class of *necessary* relays it will always have to select.

In a similar manner, if we assume Fig. 2 to be an entire graph (and not just the 2-hop neighborhood of node 2) and a series of broadcast originating from node 2, then the nodes 0, 6, 7, 8 never retransmit: they have no neighbors to cover that haven't already been covered by previous retransmissions. This comes from their position as *border nodes*: their entire 1-hop neighborhood is a subset of the 0- and 1-hop neighborhood of the node they received the message from. As a consequence of that, any retransmission from their part is necessarily redundant: all of their neighbors have already been reached by the time they retransmit. Generally speaking, **any algorithm** that minimizes retransmissions by forbidding strictly redundant ones (in particular, any version of MPR) creates among its 1-hop neighbors a class of *border* nodes that it will never select.

The uneven repartition of retransmissions across the graph is partly determined by the number of *necessary* and *border* nodes, and these depend on the structure of the graph. Hence, in order to measure the impact of our algorithm on both extra retransmission cost and its fairer repartition, we have to run both algorithms on a large sample of randomly generated graphs.

VI. Evaluation

A. Metrics

Performance of a MANET broadcast algorithm is traditionally evaluated in the literature according to the following metrics [7]:

- the reachability, that is the ratio of nodes that have received the message by the total number of nodes.
- the average rate of retransmissions caused by the algorithm.

To evaluate the repartition of retransmissions, we will also measure the variance of the rate of retransmissions:

$$\mathbb{V} = \frac{\sum_{i \in V} (x_i - \bar{x})^2}{n} \quad (1)$$

x_i being the rate of retransmissions of node i, V the vertex set of the graph, n the cardinal of V, and $\bar{x}$ the average rate of retransmissions across the graph. We see that this nonnegative quantity will be higher the more the repartition of retransmissions across the graph is unfair, and that a variance of 0 means perfect equality in the rate of retransmissions: in this case, the rate for each node is identical to the average rate.

The variance can be decomposed as such:

$$\mathbb{V} = \frac{\sum_{i \in B} (x_i - \bar{x})^2 + \sum_{i \in \overline{B}} (x_i - \bar{x})^2}{n} \quad (2)$$

With B the set of necessary and border nodes. The first term is the part of the variance that will be constant regardless of the algorithm, provided the algorithm garantees perfect coverage of the 2-hop neighborhood and forbids strictly redundant retransmissions (by forbidding the selection of border nodes). Therefore, the quantity:

$$\mathbb{V}_{min} = \frac{\sum_{i \in B} (x_i - \bar{x})^2}{n} \quad (3)$$

is the *theoretical minimum variance* for any MPR algorithm: any attempt to lower the variance below that threshold comes at the expense of either coverage (by not including strictly necessary relays) or retransmission rate (by selecting as relays nodes that add no coverage). $\mathbb{V}_{min}$ depends on the graph topology and the source of the broadcast. It can be computed by marking beforehand the vertices of the graph which are either necessary relays or border nodes, assuming a specific broadcast source node, as illustrated in Fig. 5.

B. Graph generation

Since the expected gain of our algorithm depends on the structural properties of the graph, the way to generate the graphs matter. We propose using the procedure of random geometric graph [17]: we place N nodes at random in a 500 x 500 meters square, according to a uniform distribution. We assume a range of radiotransmitters R of 100 meters. Two nodes are connected if, and only if, their Euclidean distance is less than 100 meters. Such a network is not necessarily connected, we therefore prune the generated networks that aren't. Since the value of N will have a direct influence on the average degree of the graph [17], and therefore the performance of MPR algorithms, we propose to create 5 samples of 100 random geometric graphs with N being respectively 50, 100, 150, 200 and 250. In our simulated samples, this corresponds to an average node degree of respectively 5.16, 10.46, 15.69, 20.93 and 26.16.

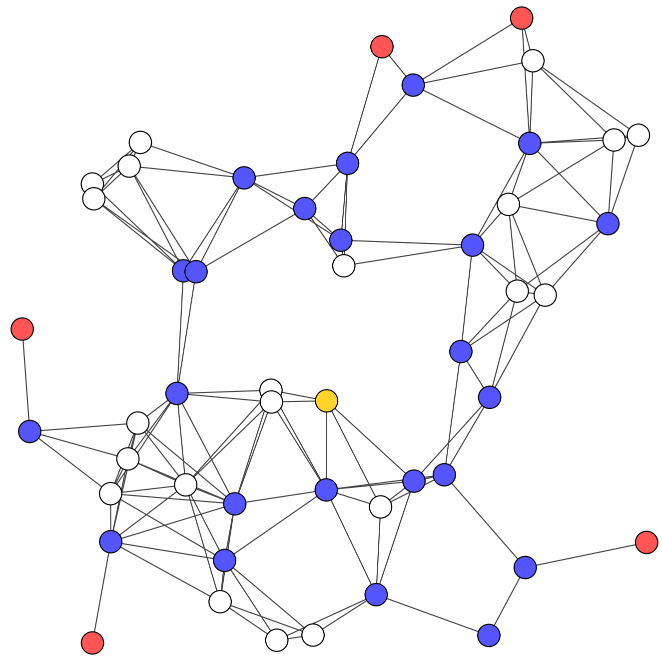


Fig. 5. Assuming a broadcast starting from the yellow node, we can mark the necessary nodes (blue) and the border nodes (red).

C. Discrete event simulator

For a realistic simulation of MANET, we need an accurate modelisation of the WiFi radiotransmitters and of the IEEE 802.11 MAC layer, so that the phenomenon of collisions is correctly represented. The discrete event simulator OMNeT++ [6], and the associated INET framework, provides such a model that we will use. We have implemented static MPR, alternating MPR, as well as plain flooding in order to provide a baseline of the number of retransmissions. A specific node will be chosen at random to initiate a broadcast. The messages will have an applicative payload of fixed size, and will be sent sequentially, such that a message will have to be completely propagated before the start of the broadcast of the next one.

Because we are interested in the repartition of energy costs, we decided to implement a simple energy model that associates a specific amount of power drawn to each state of the antenna. In line with [15], we based our values for drawn power on the Broadcom BCM4329 (or 4325) chip [18]. The energy consumption for each state is the following: idle (0 W), receiving (0.3 W) and transmitting (1.2 W). These values are used in our own state-based energy model.

D. Results

We first check with Fig. 6 that our algorithm does not degrade reachability: it is almost at 100% for every degree setting:

Regarding the proportion of retransmitted messages on Fig. 7: as expected, flooding retransmits all messages, while

TABLE I
SIMULATION PARAMETERS

Area	500 x 500 m
Number of nodes	50, 100, 150, 200, 250
Transmission range	100 m
Transmitter bitrate	10 Mbps
Applicative payload	56 B
Amount of messages	20
Antenna power draw	0W (I), 0.3W (R), 1.2W (T)

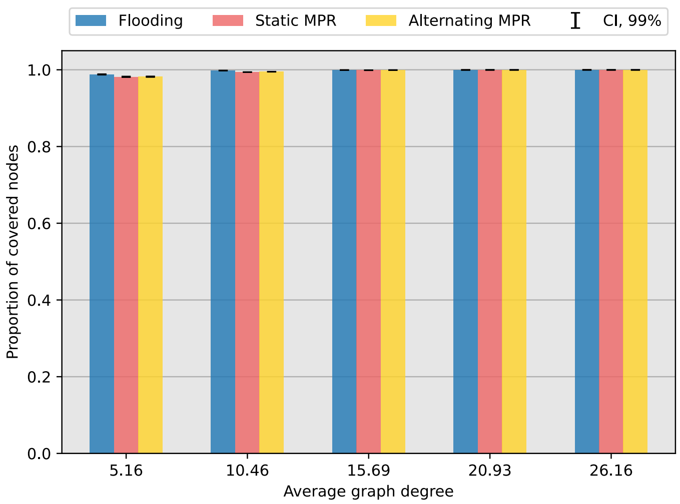


Fig. 6. Reachability

static and alternating MPR allow to save more than half the retransmissions. Alternating MPR causes slightly more retransmissions (between 2% and 5% more), which stems from the fact that the relay sets after the first one contain more relays.

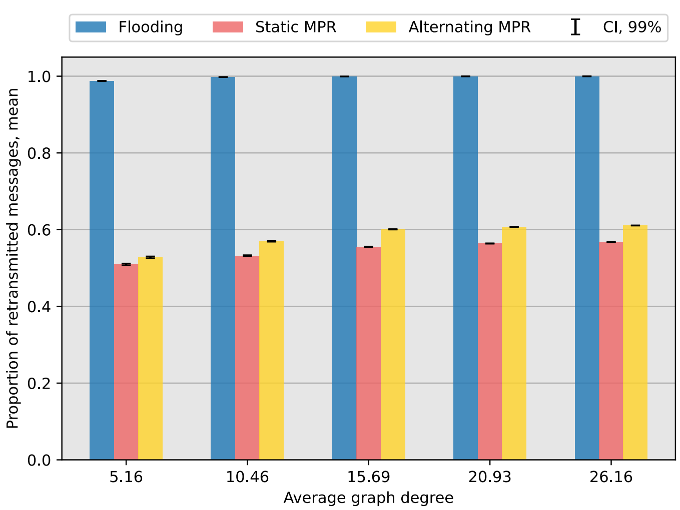


Fig. 7. Retransmissions (mean)

However, this slight extra cost in retransmissions allows us to reach a sensibly fairer repartition of them, as shown by Fig. 8. Flooding offers the fairest repartition, since every node retransmits once per application message when there are no collisions, which explains the near zero variance. As expected, the use of alternating sets of relays offers a fairer repartition than the use of only one. Indeed, alternating MPR realizes between 40% (highest average degree of 26.16) and 55% (lowest average degree of 5.16) of the maximal possible variance decrease.

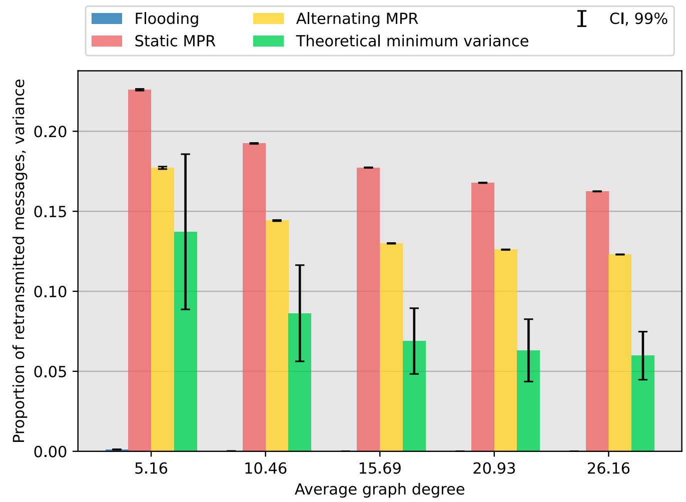


Fig. 8. Retransmissions (variance)

These findings are confirmed when we compute in the energy cost of transmissions in Fig. 9: we see alternating MPR realizing between 38% (for 20.93 and 26.16) and 55% (for 5.16) of the maximal possible variance decrease.

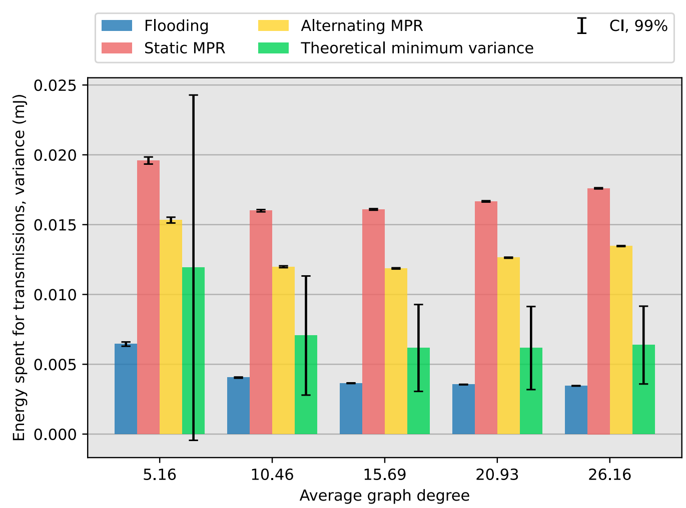
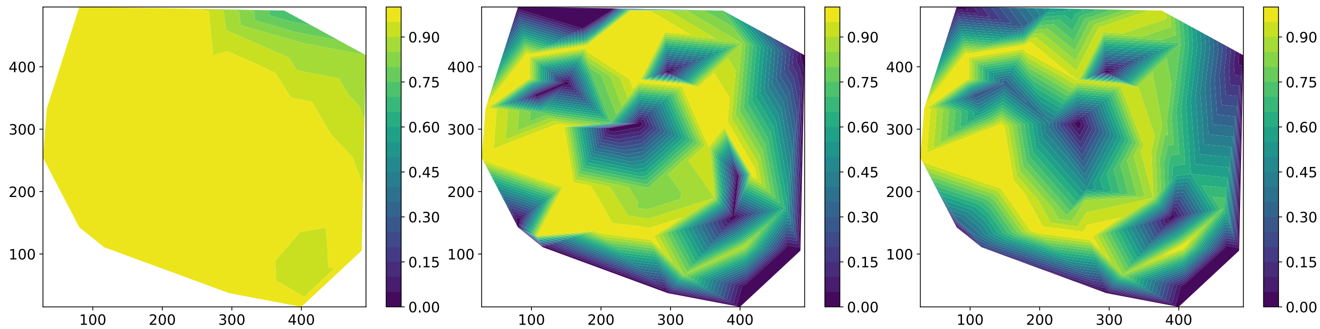


Fig. 9. Energy spending (variance)

Fig. 10 allows us to better understand the effect of alternating MPR on the retransmission rate across the graph: in the case of flooding, the rate is almost everywhere identical to 1, since every node who receives retransmits (collisions have occurred where the rate is not exactly 1). For static MPR, the retransmission rate is 1 in some areas, and very low in others, particularly in areas including border nodes. We note that alternating MPR manages to reduce the size of the yellow



(a) Flooding (b) Static MPR (c) Alternating MPR

Fig. 10. Contour of the rate of retransmissions

area. When comparing with Fig. 5 [2], we see that most yellow areas alternating MPR failed to reduce are those that contain the necessary nodes.

In conclusion, our algorithm strives to be a good compromise between a very fair but inefficient algorithm (flooding) and a very efficient but very unfair one (static MPR).

VII. Conclusion and future works

We have proposed an improvement of the MPR broadcast algorithm that allows alleviating its bad repartition of retransmissions, at a little cost in the total of occurred retransmissions. Our improvement consists in building several relay sets, whose intersections include only strictly necessary relays, and using them in turn. Our experiments on a large sample of random connected geometric graphs show that our mechanism significantly decreases the variance while only slightly increasing the total number of retransmissions, and keeping the reachability intact.

As we have seen, the connected dominating set (CDS) resulting from the MPR algorithm is specific to a source. Several improvements to MPR have been proposed in order to produce a connected dominating set that would be independent of the source [13], [19], [20], where regardless of the source of the broadcast, all nodes of the CDS, and only those, would retransmit. The problem of energy unfairness will therefore be even more acute than in a source-dependent CDS like MPR, where unfairness is partly masked by the fact that the different CDS are not identical in general. We would therefore like to provide an algorithm that would build several source-independent CDS rather than a single one, and construct these CDS as small and disjoint from each other as possible.

[2]It is the same graph and broadcast source in both cases.

References

[1] A. S. Tanenbaum and D. Wetherall, *Computer networks, 5th Edition*. Pearson, 2011.

[2] IEEE 802.11 Working Group, "Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2016*, 2016.

[3] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *IEEE Intern. Multitopic Conference*, 2001.

[4] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2-3, 2002.

[5] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *IEEE annual Hawaii Intl. Conference on System Sciences*, 2002.

[6] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, 2008.

[7] D. Gutiérrez-Reina, S. L. Toral-Marín, P. Johnson, and F. J. Barrero-García, "A survey on probabilistic broadcast schemes for wireless ad hoc networks," *Ad Hoc Networks*, vol. 25, 2015.

[8] P. Ruiz and P. Bouvry, "Survey on broadcast algorithms for mobile ad hoc networks," *ACM Computing Surveys*, vol. 48, no. 1, 2015.

[9] Z. J. Haas, J. Y. Halpern, and E. L. Li, "Gossip-based ad hoc routing," in *IEEE Conference on Computer Communications*, 2002.

[10] A. Mohammed, M. Ould-Khaoua, L. M. Mackenzie, C. Perkins, and J.-D. Abdulai, "Probabilistic counter-based route discovery for mobile ad hoc networks," in *International Conference on Wireless Communications and Mobile Computing*, 2009.

[11] H. Ling, D. Mossé, and T. Znati, "Coverage-based probabilistic forwarding in ad hoc routing," in *International Conference On Computer Communications and Networks*, 2005, pp. 13–18.

[12] J. Cartigny and D. Simplot, "Border node retransmission based probabilistic broadcast protocols in ad-hoc networks," in *IEEE annual Hawaii Intl. Conference on System Sciences*, 2003.

[13] C. Adjih, P. Jacquet, and L. Viennot, "Computing connected dominated sets with multipoint relays," *Ad Hoc & Sensor Wireless Networks*, vol. 1, no. 1-2, pp. 27–39, 2005.

[14] A. Casteigts, S. Chaumette, F. Guinand, and Y. Pigné, "Distributed maintenance of anytime available spanning trees in dynamic networks," in *Ad-hoc, Mobile, and Wireless Network Conference*, 2013.

[15] R. Carvajal-Gómez, I. Y. Gonzalez-Herrera, Y.-D. Bromberg, L. Réveillère, and E. Rivière, "Density and mobility-driven evaluation of broadcast algorithms for manets," in *IEEE International Conference on Distributed Computing Systems*, 2017.

[16] R. Carvajal-Gómez, Y.-D. Bromberg, Y. Elkhatib, L. Réveillère, and E. Rivière, "Emergent overlays for adaptive manet broadcast," in *Symposium on Reliable Distributed Systems*, 2019.

[17] M. D. Penrose, *Random Geometric Graphs*. Oxford University Press, 2003.

[18] BROADCOM. (2016). BCM4325 datasheet, [Online]. Available: https : / / www . infineon . com / dgdl / Infineon - BCM4329_Single - Chip_IEEE_802 . 11_a_b_g_n_MAC_Baseband_Radio_with_Integrated_Bluetooth_2 . 1_ + _EDR_and_FM_Transceiver_(generic)_Datasheet - AdditionalTechnicalInformation - v07_00 - EN . pdf ? fileId = 8ac78c8c7d0d8da4017d0ee21a5a6872.

[19] J. Wu and W. Lou, "Extended multipoint relays to determine connected dominating sets in manets," in *IEEE Conference on Sensor and Ad Hoc Communications and Networks*, 2004, pp. 621–630.

[20] J. Wu, "An enhanced approach to determine a small forward node set based on multipoint relays," in *IEEE Vehicular Technology Conference*, vol. 4, 2003, pp. 2774–2777.