

Where have all the MPRs gone? On the Optimal Selection of Multi-Point Relays[☆]

Leonardo Maccari^a, Mirko Maischberger^b, Renato Lo Cigno^a

^a*Department of Information Engineering and Computer Science, University of Trento, Italy*

^b*HERE Deutschland GmbH*

Abstract

Optimized Link State Routing (OLSR) is a widespread routing protocol in wireless mesh networks: static, mobile, ad-hoc, and even sensor networks. The selection of Multi-Point Relays (MPRs) that form a signaling backbone is at the heart of the protocol and it is a crucial process to reduce the signaling overhead. Since the protocol proposal and specification, the original heuristic for MPRs selection has been largely studied showing it has good local properties; however, this does not give insight about the properties of the global set of MPRs. Here lays the contribution of this paper: First we define the problem of the minimization of the global MPR set (the union of all the MPR sets) as a centralized integer linear programming problem, which is NP-hard. We are able to solve it for networks of practical size, up to 150 nodes. Second, we define a bound that we call the “distributed optimum,” which we show to be a lower bound for distributed MPR selection algorithms, still requiring considerable power to be computed. Finally, we set-up an experimental performance evaluation methodology and we show that a heuristic that we recently proposed performs very close to the distributed optimum, and always outperforms the original heuristic.

Keywords:

Optimized Link State Routing, Multi-Point Relay, Wireless Networks Management, Signalling Traffic

[☆]This work was financed partially by the European Commission, H2020-ICT-2015 Programme, Grant Number 688768 “netCommons” (Network Infrastructure as Commons).

Email addresses: leonardo.maccari@unitn.it (Leonardo Maccari),
mirko.maischberger@here.com (Mirko Maischberger),
renato.locigno@unitn.it (Renato Lo Cigno)

1. Introduction

Wireless mesh and ad-hoc networks are the subject of a very large body of literature that studied their characteristics and performances. They are an established reality for mainly two reasons: i) the continuous improvement of the 802.11 standard, and ii) the steep decrease in the price of professional wireless devices. Today it is possible to realize a full duplex, point-to-point, stable communication link based on 802.11ac with more than 100 Mbit/s for less than 150\$; and this on distances of several kilometers. The consequence is the possibility to realize city-wide mesh networks made of hundreds of nodes. Some of them have been deeply analysed in the recent literature [1, 2, 3], highlighting advantages, results achieved, but also critical points to be solved [4, 5]. The trend to build large wireless metropolitan (and beyond) mesh networks continues, helped by the evolution of 802.11. For instance, the support of beamforming in 802.11ac standard opens new possibilities to achieve both the flexibility of omnidirectional antennas and the performance of directive ones [6].

In parallel with the evolution of citizens-based, bottom-up wireless networks, the mesh paradigm is entering also in enterprise access networks, where so far the typical design consists in using a structured, wired Distribution System to connect all the Access Points (APs) to a centralized network controller. With gigabit access networks the wired infrastructure is strained and overloaded, and the so-called “tromboning” effect must be avoided¹.

Newer network products re-distribute the intelligence in the APs, which can decide to create mesh networks to keep the local traffic out of the wired infrastructure. As a result, some mainstream vendors of wireless products have recently started to produce purely wireless mesh devices and implementing cooperative control that transform an access network into a distributed wireless-wired heterogeneous network². Aerohive whitepaper [7] explains the fundamental concepts of this new trend.

As mesh networks grow larger and carry more traffic, it becomes of the utmost importance to improve the scalability of distributed routing protocols that build

¹The term tromboning is used to broadly identify unnecessary use of backbone or backhaul to carry traffic that would be naturally limited within the access network.

²See for instance the products by Ubiquiti <https://unifi-mesh.ubnt.com/> and Aerohive <http://www.aerohive.com/solutions/technology/cooperative-control-wlan-architecture.html> two of the largest world vendors of wireless enterprise products.

and maintain the mesh topology. One key theme that has been largely studied is the “topology reduction,” that is, the choice of a subset of nodes of the network that behaves as a virtual backbone. Several techniques have been proposed, such as clustering strategies, connected dominating sets, and MPRs [8]. We concentrate on MPRs, since they are at the base of OLSR [9], which is widely used in real mesh networks and has been largely studied in literature [10, 11, 12].

MPRs are defined as follows: given a node in a network graph, the set of its MPRs is a subset of its 1-hop neighbors (nodes with direct radio connection) that guarantees to reach all its 2-hop neighbors. Each node independently selects its MPR set, and only nodes that have been selected by at least one of their neighbors generate and forward topology-control (TC) messages. TCs are flooded to the whole network and represent the largest part of the control traffic. Thus reducing the total number of MPRs is a key factor to improve the scalability of the network. The selection of the smallest MPR set for a node is known to be an NP-complete problem, OLSR adopts a heuristic algorithm that produces an MPR set whose size is within a logarithmic bound from the local minimum [13]. Unfortunately, the minimization of each MPR set does not guarantee the minimization of the global MPR set (the union of all the MPR sets) that is the real goal to be pursued.

Focusing on the local selection of the MPR set and on its local properties is thus not the right way to tackle the problem. Only a few papers like [12, 14, 15] have considered the problem of the minimization of the total number of MPRs in the network. The lack of literature is due to two reasons: The first one is that the computation of MPRs is done locally by each node, so there is no easy way to perform a global coordination; the second one is that even a centralized approach would not scale for any meaningful network size, since minimizing the global MPR set is obviously NP-complete as selecting the local MPR set at each node is NP-complete. The works above compare their proposals with the outcome of the standard MPR selection or with some other heuristic; in doing this they show improved performance, but cannot discuss how far their results are from the global optimum, nor the residual improvement margin of any of these techniques.

In light of this discussion, the optimal selection of a global MPR set requires to know the real minimum set size and then to find an implementable distributed heuristic that gets reasonably close to the real optimum. The contribution of this paper lays exactly in filling these missing pieces.

First, we formalize the global MPR set minimization problem as an ILP (Integer Linear Programming) problem that can be solved efficiently for networks of small to medium size on off-the-shelf hardware. This is a key contribution that makes it possible to compare any technique with the true global minimum. To

achieve this goal we split the problem in two steps: *i)* Enumerate all the local solutions; and *ii)* Choose one solution per node in order to minimize their union. This global minimum is not of practical use as it requires a centralized coordination and high computation power. It represents a lower bound that can be taken as a reference, but no distributed protocol running on embedded hardware can implement it.

A second contribution is the derivation of an approximate algorithm that reduces the number of solutions generated by the first step so that it can be possibly handled by a mesh node, and transmitted to the rest of the network nodes. If the network is stable enough (for instance, a very dense mesh network used for indoor coverage) this strategy may be used without a centralized coordination, thus, we call this the ‘*distributed minimum*’, as it is a more reasonable bound to compare real protocols with, rather than the true NP-complete optimum.

The third contribution of this paper is a simple albeit essential modification of the standard OLSR heuristic. We proposed it in a preliminary work [16], it is fully distributed and does not need increased resources compared to the standard OLSR heuristic. We compare our proposal directly with those values, showing that it performs very close to the distributed minimum and also to the global minimum in some topologies. As a reference, we include in the comparison also the results obtained with the standard OLSR heuristic, which is still relevant as it is included in both the first and the second version of the OLSR RFC.

2. Related and Background Work

OLSR is one of the most widely used routing protocol, deployed in running mesh networks made of hundreds of nodes [1, 17, 18] and currently at its second generation [19]. The presence of MPR and their selection is one of the distinguishing features of OLSR.

The performance of MPR-based flooding in terms of number and distribution of MPRs has been initially investigated using a theoretical approach [20, 21]. This led to the definition of the RFC heuristic and to the description of its properties, showing that it performs reasonably well in reducing the size of each single MPR set [13].

Further improvement of the MPR choice has proven difficult. A wide body of literature focused on the study of heuristics that exploits the knowledge of more information to reduce the MPR sets, or to achieve some other goal. For instance, [22] shows that if a node knows the physical position of its neighbors it can compute the minimal MPR set in polynomial time. In [23] four variations of the same

heuristic are proposed that try to maximize MPR properties such as in-degree or overlapping of neighbors sets. In [24] the concept of MPR is extended in order to take into account energy-preserving strategies for battery powered networks. Another set of works address the problem of (locally) choosing the best MPR set taking into account link quality metrics [25]. Note that link-quality is essential for routing, but reducing broadcast is equally important, in fact, the last revision of the OLSR standard introduces two sets of MPRs (one for topology dissemination, and the second for traffic routing). While in Section 3 we review the relationship between routing and MPRs, the general problem of MPR minimization remains open. A comprehensive, albeit not very recent, survey on these strategies, including QoS-based ones, is presented in [26].

More recently the problem has been tackled again using other approaches. A new heuristic that increases the robustness of message diffusion, based on geometrical consideration, has been presented in [27]. Trust-based MPR choices have been analysed [28] to enhance the security of the system. In [29] the authors propose an heuristic that better takes into consideration the *willingness* of nodes to be chosen as MPRs. The same work also formalizes the problem of local MPR selection as a mixed-integer problem which can be solved with enough computation power to compare the proposed heuristics with the optimal result. In this paper we took a similar approach, but we do not limit our analysis to the local problem, we tackle the problem of global minimization of MPRs in the whole network.

Another set of works addressed the problem of minimizing the global MPR set (and not only each MPR set). In [11] and [15] the authors introduced a centralized algorithm to find the smallest global MPR set. The same authors in [14] propose a QoS based MPR selection scheme that introduces the notion of inefficient MPR (an MPR with few selectors). In [12] a cooperative MPR selection algorithm is presented in which nodes are split in master and slave roles and the MPR choice is performed accordingly. In [30] a theoretical analysis is presented that complements our work and a centralized scheme is also proposed that improves the original heuristic of about 10%.

Other works, non strictly OLSR-related, rely on MPRs. The authors of [31] introduce a basic pruning algorithm to produce a Connected Dominating Set (CDS) out of a non optimized MPR set. The CDS is a set of nodes that achieves the same broadcasting function of MPRs, but is created with a different logic and uses different rules to re-distribute the packets. The same direction is followed by [32] and [33], which improve the initial idea optimizing the CDS formation. Further improvements are proposed in [34]. Interestingly, these papers move from the same problem we described (overlapping MPR sets), but instead of minimiz-

ing the global MPR set while building it, they first compute an over-sized global MPR set as a base for a CDS, then introduce a mechanism to reduce the CDS a posteriori. The interest in MPRs is not limited to the context of mesh networks; as an example, MPR have been proposed to broadcast messages in distributed peer-to-peer systems [35] also reducing the risk of collusion attacks [36].

In a previous work [16] we have introduced a distributed strategy to minimize the global MPR. We called this strategy Selector Set Tie Breaker (SSTB) since it changes the original heuristic in the tie-break criteria. SSTB is revisited in Section 3 in light of this paper contributions, improving the insight on why tie-breaks are so important to minimize the global MPR set. SSTB does not interfere with the standard OLSR protocol, it is completely back-compatible, can even be deployed on just a subset of the nodes in a network, which makes it suitable for gradual migrations of existent networks. SSTB is fully distributed and does not need more information than OLSR already distributes.

After we proposed SSTB, the authors of [8] discussed two MPR selection improvement approaches, one of which is basically SSTB, albeit they seem to be unaware of our work.

3. OLSR Primer

We recall only the concepts related to OLSR and MPR selection which are instrumental to follow the paper, since it is a subject very well treated in the literature. Table 1 collects the most relevant definitions used in the paper, some of them will be introduced later on.

In OLSR each node periodically sends an HELLO message, used by the neighbors to sense its presence. HELLO messages from node i contain the IP addresses of the 1-hop neighbors known at the time, so that at steady state each node will have a full knowledge of its 2-hop neighborhood.

Given this, the MPR set $M(i)$ of a node i is an arbitrary subset of its symmetric 1-hop neighborhood $N_1(i)$ that satisfies the following condition: Every node in the 2-hop neighborhood $N_2(i)$ of i must have a symmetric link³ with at least a node in $M(i)$. More formally:

$$\{i \cup N_2(i)\} \subseteq \bigcup_{j \in M(i)} N_1(j) \quad (1)$$

³A link is said symmetric if i receives HELLO form j and j receives HELLO from i , it is not required that the link quality is the same. OLSR filters out links that are not symmetric.

Symbol	Meaning
N	size of the network (number of nodes)
$N_1(i)$ ($N_2(i)$)	1 (2) hop neighbor set of node i
$N(i)$	$N_1(i) \cup N_2(i)$
$M(i)$	one possible MPR set for node i
$M^*(i)$	one minimal MPR set for node i (local minimum)
M_g	Global MPR set: the union of $M(i)$ for all i
$\widehat{M}(i)$	MPR set for node i that minimizes M_g
M_g^*	Minimal global MPR set, (union of $\widehat{M}(i)$ for all i)
S_g (S_g^*)	Size of M_g (M_g^*)
$W(i)$ ($W'(i)$)	the collection of all $M(i)$ ($M^*(i)$) for node i
r_i	the size of $W(i)$
$M^1(i)$	Forced MPRs for node i
$N_1^u(i)$	Set of useful neighbors of i for optimal MPR sel.
$s(i)$	Selectors set size for i (0 if i is not an MPR)

Table 1: Notation and symbols definition

If $j \in M(i)$, then j “covers” some of the 2-hop neighbors of i . Clearly $M(i) = N_1(i)$ is always a solution that corresponds to a basic link-state protocol without any optimization. When j chooses i as MPR, it notifies i of its choice, we say that j selects i and that j is one of the *MPR selectors* of i .

Every MPR behaves as follows:

- Periodically generates a TC message containing the list of its selectors;
- Re-broadcast (flood) the TC messages it receives from its selectors.

Each node in the network must receive a copy of each generated TC. TCs contain an approximation of the local topology around an MPR sufficient to allow the construction of shortest path routing tables between any couple of nodes. Each MPR generates a TC once every given time interval. This TC is re-broadcast once by all the MPRs, thus, as a first approximation the overall signaling grows quadratically with the number of MPR. Without MPRs, the same scales quadratically with N . Note that the constant generation of signalling messages is one of the critical scalability problems of link-state routing protocols. If topology reduction does not compensate the growth of the network the only alternative is to

use the Fisheye strategy, which is known to create temporary loops [37], or to reduce the frequency of the generation of messages. This last solution is pretty drastic, as it reduces the reactivity of the network. It has been shown that the time needed to repair a broken route grows linearly with the TC timer, and thus, large timers worsen the impact of temporary outages and reconfigurations. Even in a non-mobile mesh network, in fact, local changes to the routing topologies are frequent; these can be due to shadowing, interference, or congestion of links, and need to be sensed and propagated to the rest of the network in a timely manner. While HELLO messages are used to locally detect these changes, TCs are needed to notify the rest of the network in order for all the routing tables to be recalculated. In [38, 39, 40] we introduced Pop-Routing as a technique (fully compatible with MPRs) that addresses this issue and we give a detailed analysis of the relationship between TCs and network convergence showing how this timer influences route convergence time.

To minimize the size of each $M(i)$ OLSR introduces the heuristic described in Algorithm 1, whose performance is guaranteed within a logarithmic bound from the optimal [13]. The reasoning of the heuristic is to introduce first the forced MPRs, i.e., those nodes that cover 2-hop neighbors that would otherwise remain unreachable, and then use a greedy algorithm starting from the node that covers the highest number of uncovered 2-hop neighbors (a property called *reachability*). If two nodes have the same reachability, a tie-break is needed. In the greedy algorithm the tiebreak function determines the direction in which the solution tree is navigated, so it has a relevant impact in the final choice. The first tiebreaking criterion (see Algorithm 2) is through the *willingness*, a configuration parameter that encourage or discourage neighbors to select a node as MPR. However, unless a network is planned in advance, the willingness is set to the default value for all the nodes, thus it is rarely useful.

The second tiebreak criterion prefers the neighbor u with the highest connectivity degree in the set $N_1(u) \setminus N_1(j)$, i.e., the one that has more neighbors not in common with j . If also this second criteria fails, then no further tiebreak rule is specified and the choice becomes implementation-dependent.

Let $M^*(i)$ be the optimal (minimal) $M(i)$ and $M^1(i)$ the MPR set of forced MPRs generated by the first step of the algorithm⁴. It is intuitive that $M^1(i) \subseteq$

⁴ $M^*(i)$ may not be unique, there can be more than one solution with minimal size. For simplicity and without loss of generality whenever it is not relevant we refer to the minimum as if there is only one.

```

Input:  $N_1(j), N_2(j)$ 
 $N'_2(j) = N_2(j);$ 
  // Step 1
for  $w \in N_2(j)$  do
  | if  $N_1(w) \cap N_1(j) = u$  then
  | |  $u \rightarrow M(j);$ 
  | |  $N'_2(j) = N'_2(j) \setminus w;$ 
  | end
end
// Step 2
while  $N'_2(j) \neq \emptyset$  do
  |  $N'_1(j) = N_1(j) \setminus M(j);$ 
  | Rank( $N'_1(j)$ )
  | if Tie then
  | | Tiebreak( $N'_1(j)$ )
  | end
  |  $u \rightarrow M(j);$ 
  |  $N'_2(j) = N'_2(j) \setminus [N_1(u) \cap N_2(j)];$ 
end
// Function Rank
Input:  $N'_1(j)$ 
for  $w \in N'_1(j)$  do
  |  $w.reach = ||N_1(w) \cap N'_2(j)||$ 
end
Sort( $N'_1(j), u.reach$ );

```

Algorithm 1: The OLSR MPR selection heuristic

$M^*(i)$, since the nodes in $M^1(i)$ are necessary for coverage. Let $S^*(i) = ||M^*(i) \setminus M^1(i)||$ be number of additional nodes in the optimal solution and $S(i) = ||M(i) \setminus M^1(i)||$ the number of additional nodes given by a run of the heuristic. In [13] it is shown that the following relation holds:

$$S \leq \log_2(\Delta)S^* \quad (2)$$

where Δ is the maximum number of nodes in $N_2(i)$ that a node in $N_1(i)$ can cover. This property is independent from the tie breaking mechanism⁵.

⁵The OLSR RFC suggests a further step to reduce a-posteriori the MPR set after it has been

```

// Function Tiebreak
Input:  $N_1'(j), TopReach$ 
TieSet =  $\emptyset$ ;
for  $w \in N_1'(j)$  do
  if  $w.reach = TopReach$  then
     $u \rightarrow TieSet$ ;
  end
end
Sort(TieSet,  $u.willingness$ );
if Tie then
  for  $w \in TieSet$  do
     $w.cdeg = ||N_1(w) \setminus N_1(j)||$ ;
    if  $w.cdeg < max\_willingnes$  then
       $TieSet = TieSet \setminus w$ 
    end
  end
  Sort(TieSet,  $u.cdeg$ );
end

```

Algorithm 2: The standard OLSR tie-breaking: *TopReach* is the highest value of reachability in the set of possible MPRs

3.1. Selector Set Tie Breaker (SSTB)

In the SSTB strategy we proposed in [16], the only difference from the original heuristic is the tie-break. Before applying the original tie-break we introduce another step, which simply prefers among the possible nodes the one that is already an MPR for another node, and among MPRs, the one with the highest number of selectors. More formally, when node i needs to break the tie between two potential MPRs k and m , i will choose k if $s(k) > s(m)$. If $s(k) = s(m)$ the tie is not broken, and the OLSR tie-break applies. Even if the tie is not explicitly broken by Algorithm 2, we assume that an implementation of the heuristic will produce the same outcome when executed twice with the same input (i.e., when the topology does not change from one run to the next). The number of current selectors is known because an MPR generates TC messages that carry exactly this information. This simple strategy polarizes the choice of the MPRs towards the

built. To the best of our knowledge this further step does not improve the theoretical performance of the heuristic, and in any case it does not address the problem of the global MPR set, so we do not consider it here.

most “popular” ones, and thus increases the overlapping between MPR sets of neighbor nodes. Note that SSTB is different from other tie-breaking strategies, which generally use some specific node property (such as the willingness in [29]) to break ties. A tie-breaking strategy uses an ordering $o(\cdot)$ between nodes (for instance based on the originator IP address) and gives a preference based on the ordering. Consider node i and j that have to break the tie between two potential MPRs (let’s say k and m). If $o(k) > o(m)$ they will both choose k and this locally polarises the choice. Now consider a slightly different case in which node i needs to break the tie between k and m while j needs to break the tie between m and another node n , for which $o(n) > o(m)$. Then a local tie-breaking strategy may force i and j to choose k and n respectively. Instead, with SSTB, when i chooses m then $s(m)$ is incremented, this increases the chances that $s(m) > s(n)$, and that also j chooses m instead of n . SSTB is not a local static tie-breaking criteria, it is a dynamic tie-breaking criteria with the property of self-reinforcement.

It is important to note that the choice of MPRs does not directly influence the routing decision. Unless forced to do so, OLSR does not route through MPRs only. MPRs are used to spread the knowledge of the network topology, including links to non-MPR nodes. Each node uses the information contained in TC messages, plus the information on the two-hop neighbors contained in HELLO messages to compute its routing tables with Dijkstra’s algorithm. Therefore, reducing the number of MPRs does not reduce the number of nodes that perform routing; non-MPR nodes can easily be part of the shortest paths and be used as next-hop. For this reason we do not test the consequences of MPR choice on routing performance, because in normal conditions there should be none, as already observed in [1]. If the links are extremely lossy then the number of MPRs influences the reliability of the broadcast function, as reducing the number of MPRs also reduces the redundancy of the shared information. While the general strategy must always be the minimization of the global MPR set, specific techniques can be introduced to improve redundancy for extreme cases. A deep analysis of the robustness implications of MPRs minimization are out of the scope of this paper, and it is a theme under active research in the MANET field [41, 42]. The interested reader can refer to an ongoing work from the authors on this subject [43] compatible with the technique we describe in this paper.

The new version of OLSR [19] introduces two different sets of MPRs, one for topology dissemination and another one as routing backbone. The techniques we devise in this paper apply only to the first one, leaving the second one, and hence routing, unchanged. Finally, we note that given a certain network topology, SSTB converges to a stable MPR set. Here we give a concise sketch of proof, which can

be easily extended to the general case in future works.

We already noted that if the network topology does not change, two runs of SSTB executed by node i will deterministically choose the same MPRs up to the first tie-break. At the tie-break if the selector sets of the candidate MPRs did not change from the previous run, SSTB will do the same choice. If this is true for all tie-breaks, then SSTB will deterministically choose the same $M(i)$. If instead, between one run and the next, the selector sets of the candidates change due to the choices of some other node, i may break the tie in a different way and $M(i)$ may change. A condition of instability happens when, $\forall k$, without any change to the network topology, if node i runs SSTB k times, a number Δ exists so that running SSTB $k + \Delta$ times produce an $M(i)$ different than the one at step k . For this to happen a feedback loop should be established between two nodes so that a tie-breaking decision by node i influences the decision taken by node j , which in turn influences the decision of node i and so on. Consider a node i and a node $j \in N(i)$ and two nodes $k, m \in N_1(i) \cap N_1(j)$. We show that a loop between i and j on the choice of k and m is impossible if nodes are not perfectly synchronized. Consider the initial condition in which $s(k) > s(m)$, then:

1. At time t_1 node i executes SSTB, and needs to break a tie between neighbors k and m . k is chosen and $s(k) - s(m) > 1$;
2. At time t_2 node j runs SSTB and needs to break a tie. We have four cases:
 - (a) The tie-break is between k and m . j will choose k and the situation does not change anymore,
 - (b) The tie-break is between $w \neq k$ and m . $s(m)$ can be incremented of one unit, the situation does not change anymore,
 - (c) The tie-break is between k and $w \neq m$. $s(k)$ can be decremented of one unit (if $k \in M(j)$) but the situation does not change anymore,
 - (d) The tie-break does not involve k or m , irrelevant.

Now consider the initial condition in which $s(k) = s(m)$, it is important to note that when node i runs SSTB, as a first step in the execution, it clears $M(i)$, and internally decreases $s(\cdot)$ for all its MPRs:

1. At time t_1 node i executes SSTB, and needs to break a tie between neighbors k and m . k is chosen and $s(k) - s(m) = 1$;

2. At time $t_2 > t_1$ node j executes SSTB and needs to break a tie. We have four choices:
 - (a) The tie is between k and m : then j must choose k and the situation does not change anymore,
 - (b) The tie is between k and $w \neq m$: if k is chosen the situation does not change anymore. Else, if w is chosen, there are two sub-cases:
 - i. At time $t < t_2$ $k \notin M(j)$: Then $s(k)$ is not decremented and the situation does not change anymore,
 - ii. At time $t < t_2$ $k \in M(j)$: Then at time $t > t_2$, $s(k) = s(m)$. When i re-runs SSTB, it will first clear $M(i)$, and contrarily to what done in t_1 it will choose m . Now i and j converged on m and the situation does not change anymore;
 - (c) The tie is between $w \neq k$ and m : if w is chosen, the situation does not change anymore. Else, two sub-cases may happen:
 - i. At time $t < t_2$ $m \in M(j)$: Then $s(m)$ is not incremented and the situation does not change anymore,
 - ii. At time $t < t_2$ $m \notin M(j)$: Then at time $t > t_2$, $s(k) = s(m)$. When i re-runs SSTB, it will first clear $M(i)$, and contrarily to what done in t_1 it will choose m . Now i and j converged on m and the situation does not change anymore;
 - (d) the tie does not involve k or m : irrelevant.

As we said, this is not a full demonstration, because a feedback may be created in a loop involving more than two nodes. The demonstration is similar but more elaborated and we leave it for future work.

If the choices are taken at the same time, then a loop may occur. Consider the following case in which at time t_0 , $s(k) = s(m) = 0$:

1. At time t_1 node i executes SSTB, and needs to break a tie between k and m . k is chosen and $s(k) = 1$; $s(m) = 0$;
2. At the same time t_1 node j executes SSTB, and makes a different choice, then $s(m) = s(k) = 1$;
3. At time t_2 node i receives a TC message from m , which says that $s(m) = 1$. Now i repeats SSTB: clears $M(i)$ and chooses m . At this point, two things may happen:

- (a) At time t_3 node i generates an HELLO message. This message is received by k , that now knows that i is not a selector anymore. No loop is created,
- (b) At time t_3 , before receiving an HELLO from i , node k generates a TC message that says that $s(k) = 1$. Now j will behave as node i at time t_2 , will prefer k instead of m and a loop is created.

Solving this issue it is pretty straightforward, we have to ensure that all nodes re-compute their MPRs right before sending an HELLO, so that their choice is notified to their MPRs without delay, and condition 3.b never verifies.

4. A different problem: Minimizing the total number of MPRs

Let M_g be the *global MPR set*, i.e., the union of all the MPR sets chosen by each node in the network, and let S_g be its size:

$$M_g = \bigcup_{i \in \{1 \dots N\}} M(i); \quad S_g = \|M_g\| \quad (3)$$

Let M_g^* be the choice of M_g with minimal size⁶, and let $S_g^* = \|M_g^*\|$. We call $\widehat{M}(i)$ the MPR set chosen by node i that satisfies

$$M_g^* = \bigcup_{i \in N} \widehat{M}(i) \quad (4)$$

As it often happens in combinatorial problems, the local minimization of each $M(i)$ does not always lead to the minimization of M_g . Consider the network of Figure 1 that represents a small portion of a larger network, and consider only numbered nodes for the sake of simplicity. For this portion of the network, the standard MPR choice of nodes 1, 6, 7 is

$$M^*(1) = \{2\}; \quad M^*(6) = \{3, 7\}; \quad M^*(7) = \{4, 6\}$$

and the contribution of nodes 1, 6, 7 to M_g is

$$M_g^{1,6,7} = M^*(1) \cup M^*(6) \cup M^*(7) = \{2, 3, 4, 6, 7\}$$

⁶Again, note that there can be more than one solution with minimal size. To improve readability and without loss of generality we do not introduce a set notation, but we refer the the general choice of one possible solution.

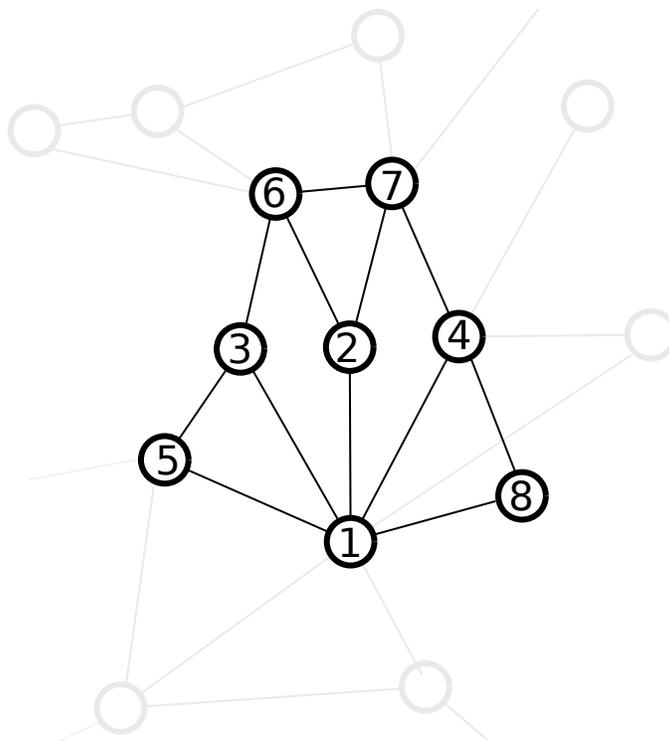


Figure 1: A small portion of a larger network

However, for node 1, the coverage given by node 2 could be replaced with the coverage given by node 3 and 4. Thus, the choice $M(1) = \{3, 4\}$, even if it is not a local optimum for node 1 would exclude node 2 from M_g . Since node 1, 6, 7 are all the neighbors of node 2, if none of them selects node 2 no other node can select it, and the M_g of the whole network is reduced of one unit, independently from the behaviour of the other nodes of G . We can thus say that

$$\widehat{M}(1) = \{3, 4\}; \quad \widehat{M}(6) = \{3, 7\}; \quad \widehat{M}(7) = \{4, 6\}$$

$$M_g^{1,6,7} = \{3, 4, 6, 7\}$$

and that $|\widehat{M}(1)| > |M^*(1)|$. This example is generalized by Proposition 1.

Proposition 1. *Given a graph G and an optimal solution M_g^* for the global MPR choice problem, such that*

$$S_g^* = \left\| \bigcup_{i \in \{1..N\}} \widehat{M}(i) \right\|$$

then

$$S_g^* \leq \left\| \bigcup_{i \in \{1..N\}} M^*(i) \right\| \quad (5)$$

Proposition 1 simply states that in general the locally optimal solutions $M^*(i)$ do not give rise to a globally optimal solution so we must look for $\widehat{M}(i)$ also among the MPR sets with non-minimal size, as the example showed. We need an algorithm to find a suitable set of $\widehat{M}(i)$ whose union yields an optimal M_g^* .

Let $W(i) = \{M_1(i), \dots, M_{r_i}(i)\}$ be the set made of all the solutions to the MPR problem for node i , where r_i is the size of $W(i)$. $W(i)$ contains the solutions of minimal size plus all the other non locally optimal solutions. Finding an M_g is an optimization problem that entails finding a suitable set of $\widehat{M}(i)$. This problem can be decomposed into two sub-problems in cascade:

1. For each node i enumerate all the elements of $W(i)$;
2. From each $W(i)$ pick one element $\widehat{M}(i)$ so that the union of all the chosen elements is minimized.

Sections 4.1 and 4.2 analyze the two sub-problems separately. Before that, we demonstrate how minimizing M_g reduces TC overhead, and that this reduction is to be added to the reduction obtained by minimizing each $M(i)$. We define the TC overhead (O) as the total number of TC packets that all wireless radios receive per second. As interference is measured at the receivers, this is a better metric than the number of generated packets. Given a certain choice M_g with a corresponding S_g , we want to show that another choice M'_g with a corresponding $S'_g < S_g$ will reduce the overhead. Proposition 2 gives an approximate upper and a lower bound for the reduction of the overhead, under two realistic constraints:

$$\|s(k)\| > 1 \quad \forall k \quad (6)$$

$$\sum_{k \in M_g} \|s(k)\| \gg S_g \quad (7)$$

Equation (6) allows us to assume that every MPR will forward the TC it receives at least once (MPRs do not forward TCs to the selector they received it from). Consider that if $k \in M^1(i)$ then there exist another node j so that $k \in M^1(j)$ and k has more than one selector. If $k \in M(i)$ but $k \notin M^1(i)$ this is not granted but highly probable. We can craft pathological cases in which an MPR

has less than 2 selectors (as the 4-node diamond network we described [16]), but we consider this condition very unlikely in real networks. The second condition generalizes the first, saying that the average number of selectors per MPR is much larger than one.

These assumption are based on a previous work in which we showed that the distribution of the selector set sizes in realistic networks has a very small number of MPRs with selector set size equal to 1 and is long tailed, so the average is dominated by the MPRs with the highest number of selectors [44]. In general, they hold when the graph density is high, which is the case in which it makes more sense to use MPRs.

Proposition 2. *Given a connected network graph G , and two choices for MPRs M_g and M'_g corresponding to a total size S_g , $S'_g < S_g$ and a choice of MPRs for every node j given by $M(j)$ and $M'(j)$, and given that the following conditions are true:*

$$\|s(k)\| > 1 \forall k, \quad (8)$$

$$\sum_{k \in M_g} \|s(k)\| \gg S_g \quad (9)$$

if:

$$\sum_{j=1 \dots N} \|M(j)\| = \sum_{j=1 \dots N} \|M'(j)\|, \quad (10)$$

then for the generated overheads O and O' holds that:

$$\frac{O'}{O} \approx \left(\frac{S'_g}{S_g} \right)^2 \quad (\text{upper bound}) \quad (11)$$

$$\frac{O'}{O} \approx \frac{S'_g}{S_g} \quad (\text{lower bound}) \quad (12)$$

Proof. Let's assume a TC is generated by each MPR every time unit, then a network with S_g MPRs generates S_g TCs per time unit. By definition, each MPR i receives every generated TC at least once and re-forwards all of them at least once (Eq. (8)), and at most $\|s(i)\| - 1$ times. We consider two cases that represent the extremes of the possible gamma of configurations, all the other configurations fall in this range.

In the first case, each node is equipped with a single wireless radio, which receives and re-broadcasts the TC. In this case the overhead is given by:

$$O = S_g \sum_{k|k \in M_g} \|N_1(k)\| = S_g^2 d \quad (13)$$

where d is the average degree of the MPRs. Equivalently $O' = (S'_g)^2 d'$, and thus:

$$\frac{O'}{O} = \left(\frac{S'_g}{S_g}\right)^2 \frac{d'}{d} \approx \left(\frac{S'_g}{S_g}\right)^2 \quad (14)$$

the approximation comes Eq. (9) as $\|N_1(k)\| \geq \|s(k)\|$ and thus the average degree of the MPRs is not strongly influenced by their number. This is actually a known and intuitive results which demonstrates Eq. (11).

In the other case, an MPR uses a different wireless radio for each of its selectors. The total network overhead is given by:

$$O = S_g \sum_{k|k \in M_g} (\|s(k)\| - 1) \simeq S_g \sum_{k|k \in M_g} \|s(k)\| \quad (15)$$

where we used Eq. (8). Note that two different choices of M_g can yield relevant differences in values of $\|s(k)\|$. For instance, in the example of Fig. 1 reducing S_g of one unit increases $\|M(2)\|$ of one unit, and thus increases the sum, and this effect may replicate in all the MPRs.

It must be noted that for any choice of M_g Eq. (16) holds:

$$\sum_{k|k \in M_g} \|s(k)\| = \sum_{i=1 \dots N} \|M(i)\| \quad (16)$$

meaning that the sum of the number of selectors per MPR equals the total number of selections per node. Thus, if we plug Eq. (16) into Eq. (15) and Eq. (8) holds, also Eq. (12) is demonstrated.

For any situation in between (MPR i has a number of radios from one to $\|s(i)\|$) the overhead reduction is a linear combination of these boundaries. \square

The best case of Proposition 2 is very common, as small mesh networks, ad-hoc networks and sensor networks generally fall in that category. For large-scale mesh networks, in which every node can have more than one radio, Proposition 2 shows that as long as the goal of minimizing S_g does not impact the goal of minimizing the sum of $\|M(j)\|$, reducing S_g has an impact at least linear on the overhead. In these networks, the use of multiple radios per node enables to have nodes

with a high number of neighbors (nodes with tens of neighbors were described in the literature [1, 2]) and justifies the approximation of Eq. (15). Note that this is exactly what happens when we compare SSTB to the original heuristic, as modifying only the tie-break we do not change its theoretical properties and thus we do not increase $\sum_{j=1\dots N} \|M(j)\|$. In fact, SSTB would have not changed the initial assignment in Fig. 1.

Summing up, reducing S_g without increasing $\sum_{j=1\dots N} \|M(j)\|$ (as with SSTB), produces a reduction of overhead that is at least linear and at most quadratic with S_g under assumptions we consider realistic.

4.1. Generating all the $W(i)$

Algorithm 3 reports a straightforward implementation that enumerates all the $W(i)$ noting that r_i is upper-bounded by the number of all possible MPR sets, which in turn is upper bounded by the number of all the possible combinations of neighbors of i of any size:

$$r_i \leq \sum_{j=0}^{\|N_1(i)\|} \binom{\|N_1(i)\|}{j} \quad (17)$$

thus enumerating all the possible elements of $W(i)$ is NP.

To reduce r_i we introduce a heuristic that limits the number of iterations in the outer loop of Algorithm 3. The heuristic is based on the observation that $w \in M^1(i) \rightarrow w \in M_g^*$. So any neighbor j of w can choose w without increasing S_g and this in many cases can greatly reduce the number of iterations required to compute $W(i)$. The baseline of the heuristic is to reduce the number of possible MPR sets $W(i)$, or in other words find a smaller upper bound to $r(i)$.

Heuristic r_i

1. For every node i compute $M^1(i)$ and set $M^1 = \bigcup_{i=0}^N M^1(i)$, this step is straightforward and computationally light;
2. For every node i set $M^1(i) = N_1(i) \cap M^1$, in practice we set the initial kernel of MPRs not only to the locally mandatory MPRs, but to all neighbors that are also mandatory MPR for any node;
3. For every node i define the set of useful neighbors $N_1^u(i) = N_1(i) \setminus M^1(i)$, i.e., the set of neighbors that are not mandatory MPR;

```

// G = The whole network graph
Input: G
N1(i), N2(i) = GetNeighbors(i, G)
W(i) = ∅;
for j ∈ 1...||N1(i)|| do
  // combinations(j, M) enumerates all the possible
  // combinations of size i from a set M
  C = combinations(j, N1(i));
  for c ∈ C do
    // IsMPRSet(c, i, G) checks if the set c is a
    // valid MPR set for i
    if IsMPRSet(c, i, G) then
      | W(i).add(c);
    end
  end
end
return W(i)

```

Algorithm 3: Straightforward enumeration of all the elements of W(i)

4. Purge from $N_1^u(i)$ all nodes that have zero reachability because the nodes in $M^1(i)$ already cover all their neighbors, these nodes will not be selected as MPR in any case.

This heuristic reduces the upper bound of r_i to:

$$r_i \leq \sum_{j=0}^{\|N^u(i)\|} \binom{\|N^u(i)\|}{j} \leq \sum_{j=0}^{\|N_1(i)\|} \binom{\|N_1(i)\|}{j} \quad (18)$$

with an improvement given by the relationship $\|N^u(i)\| \leq \|N_1(i) \setminus M^1\|$.

Algorithm 4 reports the enumeration of all $W(i)$ modified with Heuristic r_i . Algorithm 4 stills find the optimal solution, and the problem remains NP since there is no theoretical guarantee that $\|N^u(i)\| \leq \|N_1(i) \setminus M^1\|$, as for certain graphs it can be $M^1 = \emptyset$ and thus $\|N^u(i)\| = \|N_1(i)\|$. In Section 6 we present a numeric estimation on the difference between $\|N^u(i)\|$ and $\|N_1(i)\|$ in the graphs we analyzed and we show that Algorithm 4 allows the enumeration all the solutions for network graphs of practical size (up to 150 nodes) generated with a

```

// G = The whole network graph
Input:  $G$ 
 $N^u(i) = \emptyset$ ;  $M^1 = \emptyset$ ;  $W(i) = \emptyset$ 
// Compute the set of mandatory MPRs. It is done
// once per graph and not repeated for every node  $i$ .
for  $k \in \{1 \dots N\}$  do
|  $N_1(k), N_2(k) = \text{GetNeighbors}(k, G)$ 
|   for  $j \in N_2(k)$  do
|   | if  $N_1(j) \cap N_1(k) = u$  then
|   | |  $u \rightarrow M^1$ 
|   | end
|   end
| end
end
 $M^1(i) = N_1(i) \cap M^1$ 
// Compute the reachability of nodes in  $M^1(i)$ 
 $R = \emptyset$ 
for  $j \in M^1(i)$  do
|  $R = R \cup (N_1(j) \cap N_2(i))$ 
end
// Add to  $N^u(i)$  the nodes that have reachability  $> 0$ 
for  $j \in N_1(i)$  do
|  $reach = \|(N_1(j) \cap N_2(i)) \setminus R\|$ 
|   if  $reach \neq 0$  then
|   |  $j \rightarrow N^u(i)$ 
|   end
| end
end
// Finally compute  $W(i)$ 
for  $j \in 1 \dots \|N^u(i)\|$  do
|  $C = \text{combinations}(j, N^u(i))$ ;
|   for  $c \in C$  do
|   | if  $\text{IsMPRSet}(c, i, G)$  then
|   | |  $W(i).add(c)$ ;
|   | end
|   end
| end
end
return  $W(i)$ 

```

Algorithm 4: Enumeration of all the elements of $W(i)$ exploiting Heuristic r_i

realistic topology generator. Instead, the brute force Algorithm 3 fails (i.e., does not finish within a predefined time boundary) for much smaller networks.

4.2. Finding M_g^*

Algorithm 4 enumerates all the useful MPR sets $W(i)$, so it solves the first sub-problem of the optimization. The second sub-problem is choosing one suitable $\widehat{M}(i)$ per node. The problem can be described as follows: given a family of sets $W(i); i = 1 \dots N$ select one and one only element $\widehat{M}(i)$ from each set $W(i)$ so that their union is minimal. This is a problem known in the literature and called *Maximum Subset Intersection* [45], and it was proven to be NP-hard since it can be reduced to a Max-Clique problem.

In our case it is possible to formalize it as an ILP problem as follows. Given the set of available solutions $W(i) = \{M_1(i), \dots, M_{r_i}(i)\}$ for every node i let:

- δ_{ik} be a binary variable equal to 1 when node i chooses the k -th set from $W(i)$, that is when $M(i) = M_k(i)$;
- γ_j be binary variable equal to 1 when node j is selected as MPR by at least one node, so that, given a solution $M_g, S_g = \sum_{j=0}^n \gamma_j$;

so that the Maximum Subset Intersection problem can be programmed as:

$$\text{minimize } \sum_{j=1}^N \gamma_j \quad (19)$$

$$\text{subject to } \sum_{k=1}^{r_i} \delta_{ik} = 1 \quad \forall i \in \{1, \dots, N\} \quad (20)$$

$$\gamma_j = \delta_{ik} \quad \forall j, i, k : j \in M_k(i) \quad (21)$$

The objective function Eq. (19) states that we want to minimize the number of nodes used as MPRs. Constraint Eq. (20) imposes that exactly one solution k is chosen for each node i . The constraint is expressed as one equation per node, each equation with r_i variables. Constraint Eq. (21) ties the value of δ and γ , imposing that γ_j must be one if the j -th node is an MPR for node i in the k -th solution. For each node i this constraint is made of $r_i * \sum_{k=1}^{r_i} ||M_k(i)||$ equations.

With this formulation, state-of-art ILP solvers can find the optimal solution for the input generated by networks up to 150 nodes, which is a reasonable size to study the behavior of mesh/ad-hoc wireless networks.

```

// Compute  $W'(i)$ 
for  $j \in 1 \dots ||N^u(i)||$  do
   $C = \text{combinations}(j, N^u(i));$ 
   $f = \text{false}$ 
  for  $c \in C$  do
    if  $\text{IsMPRSet}(c, i, G)$  then
       $W(i).add(c)$ 
      if not  $f$  then
         $f = \text{true}$ 
      end
    end
  end
  if  $f$  then
    break
  end
end
return  $W(i)$ 

```

Algorithm 5: Modified heuristic to compute the subset $W'(i)$

4.3. A lower bound on distributed heuristic algorithms

The ILP and the algorithms in Sections 4.1 and 4.2 allow the computation of the global optimal solution M_g^* given the topology graph. With OLSR each node in the network knows the full network topology, so it can theoretically compute $W(i) \forall i$, solve the ILP problem and select the best $\widehat{M}(i)$ for itself. In practice this is not possible. The main obstacle is the size of $W(i)$ which grows as the number of all the subsets of $N_1(i)$ of any size. In our powerful servers we were able to solve the problem for the mentioned network size, but in a real distributed network devices do not have the processing power and the storage space to compute and save $W(i)$ for every node. Therefore, M_g^* is to be considered as a theoretical lower bound, and it is very useful to understand the gain margin available, but it is not meaningful to compare distributed, on-line algorithms usable in real systems against it.

For this reason we introduce a cooperative heuristic approach that is the best that can be done under the following assumptions:

- Let $W'(i) \subset W(i)$ be the subset of $W(i)$ made of only the solutions of minimal size. Every node can heuristically compute $W'(i)$ replacing the last loop of Algorithm 4 with the code in Algorithm 5.

- The network is stable enough so that every node has enough time to compute $W'(i)$ and to share it with all the other nodes in the network.
- Since $\|W'(i)\| \ll \|W(i)\|$ the number of input values to the Maximum Subset Intersection ILP problem defined by Eqs. (19) to (21) is small enough to be solved locally by each node.

The generated solution M_d^* , that we call the *distributed minimum*, is the performance bound for distributed approaches realistically implementable on networks with acceptable (albeit demanding) computational resources and can be used as a fair comparison for other distributed heuristics.

5. Evaluation Goals and Methodology

This paper has multiple evaluation goals. First of all, analyzing the global optimum in realistic networks, which has never been done to the best of our knowledge, and compare it with the solution given by the OLSR standard. Second, we compare the novel heuristic we have proposed (SSTB) with the global optimum, and also with the distributed optimum introduced in Section 4.3. The evaluation process is an integral, valuable part of our contribution. Many works analyze networks with random nodes placement, unit-disk communication model, no shadowing, and node density that can be infinitely increased. These works present interesting asymptotic results, but to appreciate the value and the applicability of a heuristic, it must be evaluated in realistic scenarios using either real experiments or detailed simulations that include all the protocol and scenario features, thus the evaluation setup becomes complex and requires some explanation.

Two separate instruments have been integrated to obtain a suitable evaluation methodology: the Omnet++ network simulator⁷ and the Gurobi ILP solver⁸. These two instruments are glued using the Python NetworkX⁹ graph manipulation library. The whole code-base is published as open source code¹⁰.

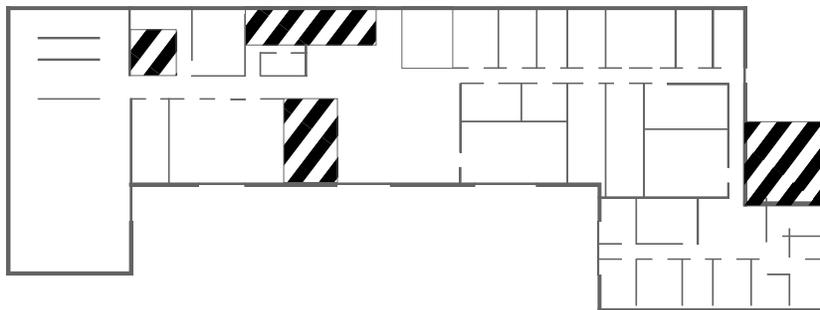


Figure 2: The map of the first floor of our department

5.1. The Omnet++ Simulator

Omnet++ is used to generate realistic topologies and to run OLSR for two different scenarios: i) outdoor with obstacles, and ii) indoor (we will refer to them as *OO* and *IN*). Simulated nodes are equipped with 802.11 wireless radios using the Omnet++ channel model (based on the corrected NIST BER tables [46]) together with a realistic ray-tracing fading model [47] that takes into consideration the presence of obstacles. In the *IN* scenario, we replicate the map of our department (roughly 100×40 m) with three distinct wall thickness respecting the building plan as depicted in Fig. 2. The *OO* scenario instead reconstructs the campus of our University in a 600×600 m area, as depicted in Fig. 3, where the obstacles are only the main buildings. In both cases the area is split in squares of 8×8 m (*OO*) and 4×2 m (*IN*), that can include points of interest for the users (bar, library, classroom, etc.). Nodes are grouped in clusters, each cluster is assigned a point of interest in one of the square areas. Each point of interest is placed in the middle of the area, and nodes are distributed around it with a uniformly random chosen radius (lower than half of the longer edge of the block) and angle (which generates a higher density close to the center). For each scenario a batch of simulation runs with different random seeds is performed with two configurations: in the first one we increase the number of nodes (from 50 to 150 in steps of 20 nodes for *OO*,

⁷<http://omnetpp.org>

⁸<http://www.gurobi.com>

⁹<http://networkx.github.io>

¹⁰See <https://ans.disi.unitn.it/redmine/projects/mpr-papers> for the source code, generated topologies, python scripts to parse the results, and basic instructions to replicate the simulations are available and updated as the work progresses.

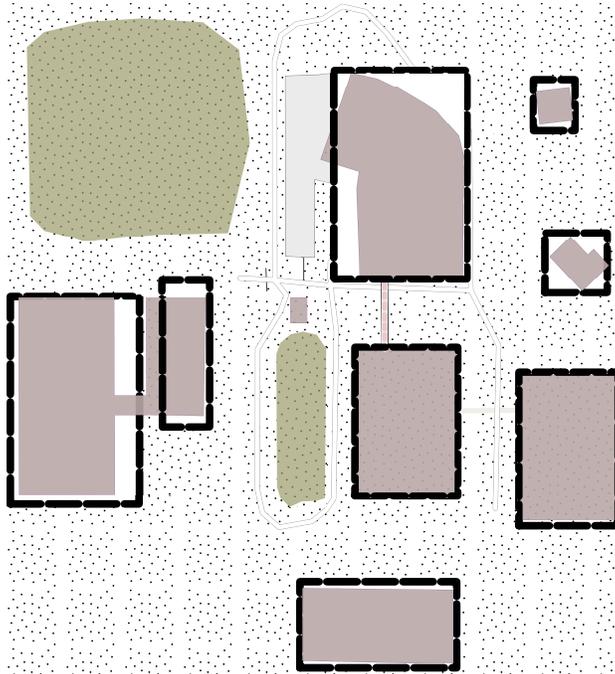


Figure 3: An approximated map of Trento University Campus. Thick dashed lines contour the relevant obstacles and how they are reproduced in the simulator

and from 50 to 110 for IN) and we keep the number of nodes per cluster constant (roughly around 10 per cluster). In the second we keep constant the number of nodes (100) and we increase the number of clusters from 3 to 9, so that the average number of nodes per cluster reduces.

Clustered nodes distributions are typical of crowded public places [48] and represent more realistic topologies compared to randomly placed nodes. These scenarios model a number of different situations and generate heterogeneous topologies that give a deep insight in the performance of the different algorithms.

The whole idea of MPR nodes is useful when the network is dense enough so that the generation of signaling can be reduced selecting only a subset of the nodes as MPRs. Nevertheless as base-line comparison, we add a third scenario: a squared area without obstacles and random placement of nodes. We call it OR and it is used as a sanity check to avoid the risk that the observed performance is the result of the scenarios rather than the algorithms themselves.

For each scenario, we execute several runs with different random seeds to

compute confidence intervals on results. Each run is considered only if the main connected component of the graph includes more than 90% of the nodes: cases when the network is partitioned are normal and realistic, but they make result interpretation impossible because of border effects and the fact that two small networks are topologically different from a large one. Each of the points in the graphs included in the remaining sections was computed with at least 15 runs.

5.2. *The ILP Solver*

In each run, Omnet++ produces three outputs: i) the network topology G generated in a realistic scenario, ii) the number of MPRs generated by the standard OLSR strategy during the simulation, and iii) the number of MPRs generated by the SSTB strategy.

At the end of each simulation run, the network topology G is given in input to the Python implementation of Algorithm 4 and 5, which generate the MPS (Mathematical Programming System) description of the ILP problem based on both the sets $W(i)$ and $W'(i)$ for every i . The MPS format is compatible with many LP solvers. We have run both the GLPK open source solver and the GUROBI commercial software¹¹, and we have chosen the latter since it gives faster convergence to the optimal solution. GUROBI is able to solve ILP problems using a set of heuristics and strategies that allow the solution of problems with thousands of constraints as the one we consider. GUROBI has been run in a server with 16 cores, 2.4 GHz CPU, and 64 GB RAM. We do not enter into the details of the procedure used by the solver because it is completely out of the scope of this paper. In some cases, though, we interrupted the process after a maximum of 6 minutes from its start. Considering both the global and distributed optimum computation this happened about 7% of the cases and in particular it never happened for the distributed optimum. In some pathological cases in fact, the network is so dense that one single node can have up to 67 neighbors and the computation of the whole $W(i)$ is not feasible.

We compare SSTB with the standard OLSR and with the two bounds we defined in this paper, while we do not consider other proposed techniques. There are three main reasons for this choice. The first is that, since for the first time we are able to compute a global (and distributed) minimum, we want to know how close SSTB compares with these bounds, and not to other heuristics. The second is that the original greedy OLSR heuristic is still the de-facto standard, as it was in-

¹¹<http://www.gurobi.com/resources/getting-started/mip-basics>

cluded in the last revision of the OLSR protocol [19]. The third is that most other proposals, like the many mentioned in Section 2, improve the OLSR heuristic by including additional features, like link quality or Willingness, which is something we don't require.

In conclusion, in Section 6 we show four absolute performance metrics and two relative ones. The absolute metrics are (ordered by increasing expected performance):

- S_g as obtained in Omnet++ by the standard OLSR algorithm. We refer to this value as M-OLSR.
- S_g as obtained in Omnet++ by the SSTB strategy. We refer to this value as M-SSTB.
- S_d^* computed a-posteriori by GUROBI using as input the partial solution set W' . We refer to this value as M-LP-Dist.
- S_g^* computed a-posteriori by GUROBI using as input the full solution set W . We refer to this value as M-LP-Cent.

Figure 4 reproduces the performance evaluation framework and the interactions between each component. The two relative metrics we present to simplify the evaluation of SSTB are:

- The relative gain of SSTB over OLSR:

$$\text{GAIN}_{\text{SSTB}} = 1 - \frac{\text{M-SSTB}}{\text{M-OLSR}}$$

- The relative distance of SSTB from the M-LP-Dist:

$$\text{DIST}_{\text{SSTB}} = 1 - \frac{\text{M-LP-Dist}}{\text{M-SSTB}}$$

6. Numerical Results and Discussion

First of all let's discuss the gain introduced by the heuristic approximation that computes $W'(i)$ instead of the entire set $W(i)$. Figure 5 reports the ECDF of $\|N^u(i)\|$ and $\|N_1(i)\|$ computed for each node, for each graph of the scenarios with the highest number of hosts (*OO* and *OR* scenario with 150 hosts, *IN* with

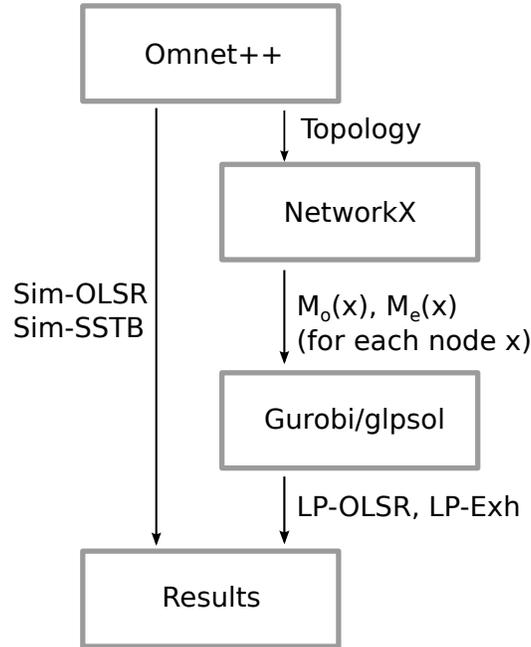


Figure 4: Methodology and framework for the evaluation

110 hosts). The figure shows that the heuristic strongly decreases the average value, and that in particular, for more than one third of the cases (two thirds for the *OO* and *OR* scenarios), the heuristic reduces the enumeration of all the possible solutions to only one solution since $\|N^u(i)\| \leq 1$. Even if the upper limit is not theoretically changed, it is clear that in realistic cases the gain is huge and the heuristic allows the computation of the optimum.

The results of the simulation with *OO* scenario are reported in Figure 6, Figure 7, and Figure 8, while for the *IN* scenario results are in Figure 9, Figure 10, and Figure 11. The whiskers in the plots are the confidence intervals computed at 95% confidence level.

A first evidence shown in all the graphs is that the OLSR heuristic, even if it is theoretically close to the local minimum, produces global set M_g far from optimal. The same graphs also show that SSTB always improves significantly over OLSR, and that SSTB dramatically decreases S_g when the network topology is dense. Figures 8 and 11, that report the relative gain, show that the gain of SSTB over OLSR is always much larger than the distance between SSTB and the distributed optimum, hinting that further gains over SSTB will not come easily. Recall that the total number of TC grows quadratically with number of MPRs thus, reducing

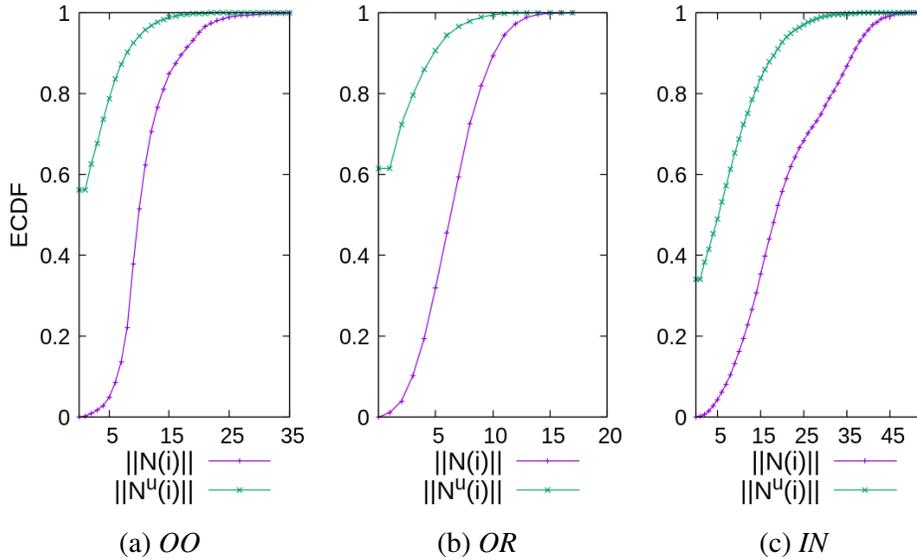


Figure 5: Size of $N(i)$ and $N^u(i)$ in the most dense network scenarios.

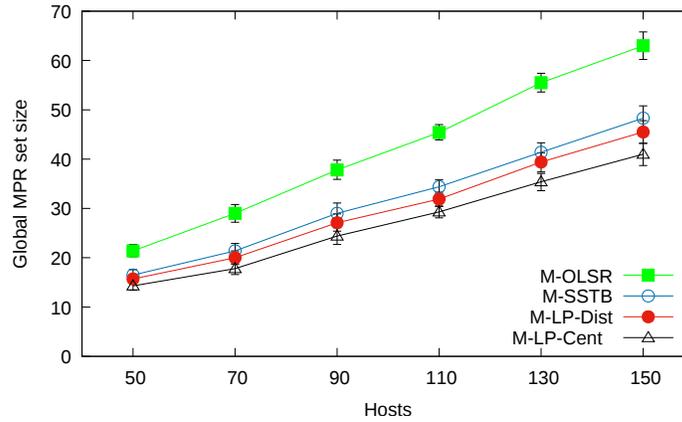


Figure 6: Absolute metrics for S_g in the *OO* scenario increasing the number of hosts, from 50 to 150.

the total number of MPRs by 30% means reducing the signaling effort to 0.5 of its original level, allowing to build significantly larger networks.

Figure 12 reports the results for the random placement scenario (*OR*) increasing the number of hosts. Both the absolute metrics and the relative gain of SSTB

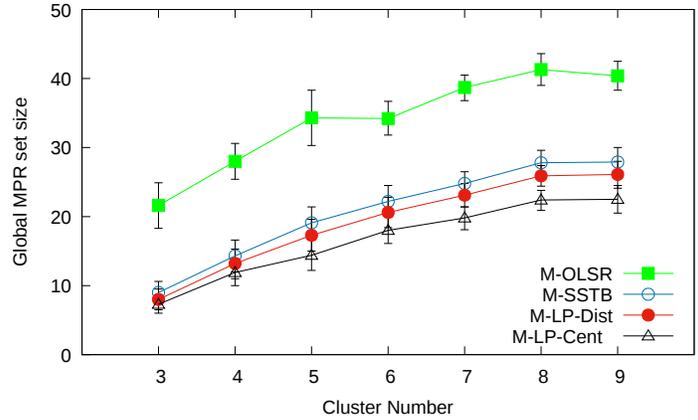
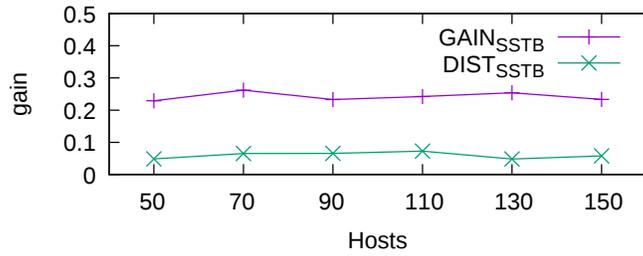
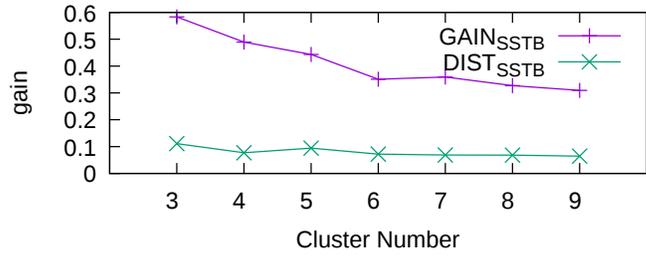


Figure 7: Absolute metrics for S_g in the OO scenario increasing the number of clusters with fixed number of hosts.



(a) Increasing number of hosts.



(b) Increasing number of clusters.

Figure 8: Relative metrics for SSTB in the OO scenario.

are plotted in this figure, while we do not report the plots increasing the number of clusters as in this scenario there is no clustering of nodes. This scenario is

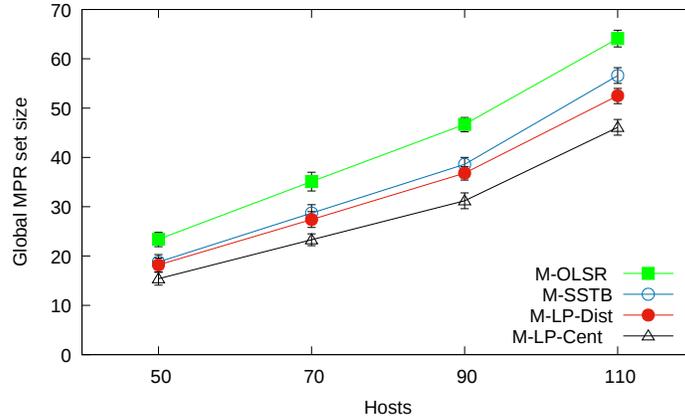


Figure 9: Absolute metrics for S_g in the IN scenario increasing the number of hosts, from 50 to 110.

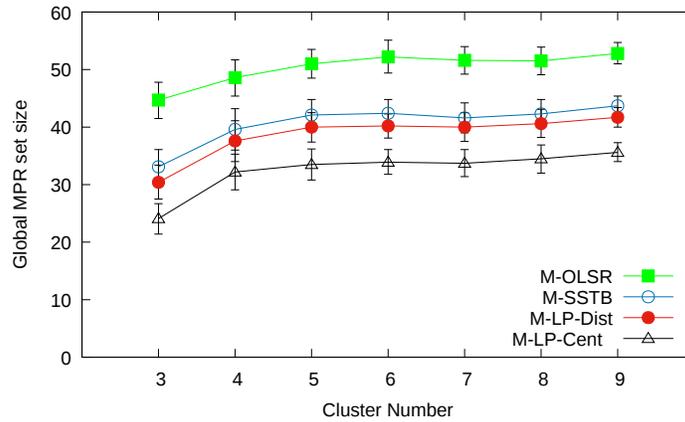
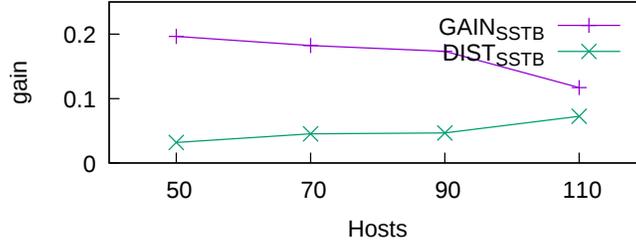
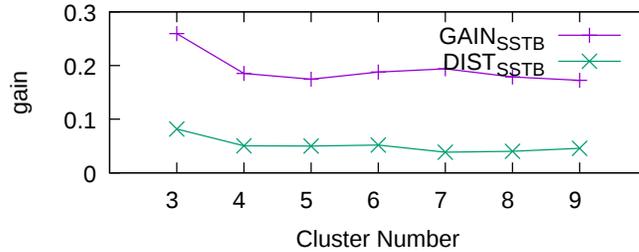


Figure 10: Absolute metrics for S_g in the IN scenario increasing the number of clusters with fixed number of hosts.

less dense compared to the others and it has the least correlation among nodes positions. In this scenario we know that S_g is larger, and we also expect less potential gain, which is confirmed by the absolute metrics in Figure 12a. Even in this scenario, however, SSTB is able to achieve more than 10% of gain compared to standard OLSR heuristic, but most of all its results are very close to the distributed optimum S_g^* .



(a) Increasing number of hosts.



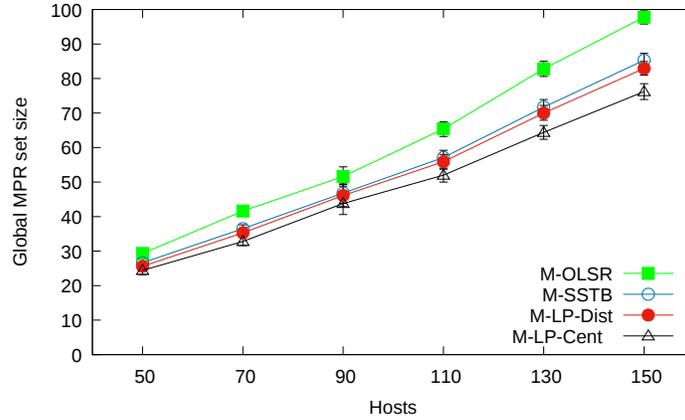
(b) Increasing number of clusters.

Figure 11: Relative metrics for SSTB in the *IN* scenario.

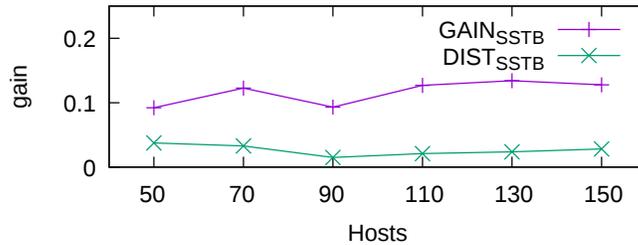
6.1. Results with mobility

So far, we focused on static networks, where it is easy to compute the optimal solutions using NetworkX and GUROBI. Nevertheless, as already discussed in [16], SSTB works also in mobile networks, as it only changes the tie-breaking mechanism in MPR selection. Unfortunately, in mobile networks not only the optimum global MPR sets are difficult to compute, but even the definition of the optimum is difficult to define as the topology graph $G(t)$ is time dependent. One can analyze consecutive snapshots of the topology, but the optimization procedure is hardly sustainable not only because of the computational complexity, but also because the on-line routing protocol works sort of incrementally, while the ILP formulation is inherently static. For these conceptual difficulties in defining an optimal selection with mobile networks we limit the analysis to OLSR and SSTB.

We consider a network made of 100 hosts that follow the realistic mobility model presented in [49]. This model splits the users into clusters and assigns every cluster to an area block, hosts are free to roam everywhere but are attracted towards the areas that contain more hosts of the same cluster. This mobility model derives its properties from social science results and the statistical properties of the traces generated are close to the ones measured in real experiments, for more



(a) Absolute metrics for S_g in the *OR* scenario; increasing the number of hosts.



(b) Relative metrics for SSTB in the *OR* scenario, increasing number of hosts.

Figure 12: Results for the *OR* scenario.

details see [49]. In our implementation we used the same scenario as the one in Fig. 3 and we introduced obstacle avoidance, so the hosts are able to move to their next waypoint avoiding convex obstacles. We varied the number of clusters from 3 to 9, as we did for the stationary configurations, each node moves with a random speed ranging from 0.2 to 1.5 m/s.

The results are reported in Fig. 13, for each run we measure the average S_g sampled over the 500 s of the simulation (excluding the first 30s which are a transitory phase, for a total of 20 samples). Each point in the graph is the average over 32 runs.

Compared to the corresponding static scenario (see Fig. 7) the number of MPRs is higher for both curves. This is due to the fact that with mobility the hosts tend to *explore* areas that are beyond the limit of their block, so that they spread on a larger area. The gain introduced by SSTB, anyway, is perfectly comparable

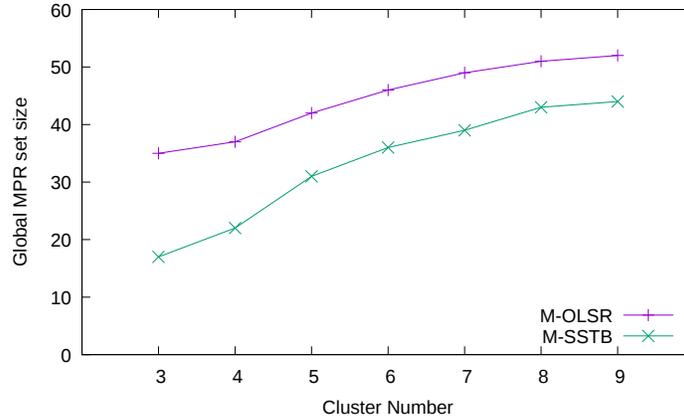


Figure 13: Results in a mobile scenario increasing the number of clusters

with the static case and confirms that SSTB can greatly reduce S_g , especially in dense networks.

7. Conclusions

Multi-Point Relays have been extensively studied in the past as a topology control technique with the goal of reducing the overhead in routing protocols. They are at the base of the OLSR routing protocol and they are even more important now that in-production mesh networks made of hundreds of nodes are emerging: The efficient use of MPRs is a key factor for efficiency and scalability. So far, several MPR selection heuristics were proposed and compared with the original one, but no evidence was produced to assess how they perform compared to the global optimum, because the global optimum was never studied.

This paper strengthens this line of research with three main contributions. First the ILP formalization for the minimization of the global MPR set. This shows, as expected, that this problem cannot be solved using local minimization strategies and we introduce an approach that makes it tractable with off-the-shelf hardware on networks of reasonable size, up to 150 nodes. The corresponding global optimum can be used as a theoretical lower bound, but it is impractical in real networks: it uses a centralized algorithm and it requires large computation power to be calculated. Second, we introduce another bound, which we call the *distributed minimum*. This technique can be implemented in a distributed network if the hardware of mesh nodes is powerful enough and the network stable enough. For this

reason we consider it a “fairer” term of comparison for distributed algorithms. Third, with an extensive experimental campaign we show that SSTB achieves a performance that is close to the distributed minimum, and in some cases to the global minimum, while always outperforming the original heuristic.

There is still some room for improvement to get closer to the global optimum. In cases in which the network topology is reasonably stable and the computation can be offloaded to some external server, for instance, a cloud-based approach can be used in order to reduce the gap to the global optimum.

In mobile networks instead, more research is needed to understand what is the best trade-off between the minimization of the global MPR set and the stability of the routing protocol and the reliability of the broadcast function based on MPRs.

Bibliography

- [1] L. Maccari, R. Lo Cigno, A week in the life of three large wireless community networks, *Ad Hoc Networks* 24, Part B (2015) 175–190.
- [2] L. Cerda-Alabern, On the topology characterization of Guifi.net, in: 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 2012.
- [3] B. Braem, C. Blondia, C. Barz, H. Rogge, F. Freitag, L. Navarro, J. Bonicoli, S. Papathanasiou, P. Escrich, R. Baig Vias, A. L. Kaplan, A. Neumann, I. Vilata i Balaguer, B. Tatum, M. Matson, A Case for Research with and on Community Networks, *SIGCOMM Computer Communications Review* 43 (3) (2013) 68–73.
- [4] R. Lo Cigno, L. Maccari, Urban Wireless Community Networks: Challenges and Solutions for Smart City Communications, in: *ACM International Workshop on Wireless and Mobile Technologies for Smart Cities (WiMobCity '14)*, part of *MobiHoc 2014*, Philadelphia, PA, US, 2014, pp. 49–54.
- [5] L. Baldesi, L. Maccari, R. Lo Cigno, Improving P2P streaming in Wireless Community Networks, *Computer Networks* 93, Part 2 (2015) 389–403.
- [6] A. U. Chaudhry, R. H. Hafez, J. W. Chinneck, Realistic interference-free channel assignment for dynamic wireless mesh networks using beamforming, *Ad Hoc Networks* 51 (2016) 21 – 35.

- [7] Aerohive Inc., Controllers, cloud, & cooperative control. the 5 generations of wlan architecture, White Paper (2015).
URL <http://www.aerohive.com/solutions/technology/cooperative-control-wlan-architecture.html>
- [8] A. Boushaba, A. Benabbou, R. Benabbou, A. Zahi, M. Oumsis, Multi-point relay selection strategies to reduce topology control traffic for OLSR protocol in MANETs, *Journal of Network and Computer Applications* 53 (2015) 91–102.
- [9] T. Clausen, P. Jaquet, Optimized Link State Routing Protocol (OLSR), RFC 3626 (IETF, Oct. 2003).
- [10] C. Adjih, E. Baccelli, T. Clausen, P. Jacquet, G. Rodolakis, Fish eye OLSR scaling properties, *IEEE Journal of Communication and Networks (JCN)* 6 (4) (2004) 343–351.
- [11] K. Yamada, T. Itokawa, T. Kitasuka, M. Aritsugi, Redundant TC message senders in OLSR, *IEICE Transactions on Information and Systems* 93 (12) (2010) 3269–3272.
- [12] K. Yamada, T. Itokawa, T. Kitasuka, M. Aritsugi, Cooperative MPR selection to reduce topology control packets in OLSR, in: *IEEE Region 10 Conference (TENCON)*, Fukuoka, Japan, 2010.
- [13] A. Qayyum, L. Viennot, A. Laouiti, Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks, Tech. Rep. RR-3898, INRIA (2000).
- [14] T. Koga, S. Tagashira, T. Kitasuka, T. Nakanishi, A. Fukuda, Highly efficient multipoint relay selections in link state QoS routing protocol for multi-hop wireless networks, in: *10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Kos, Greece, 2009.
- [15] T. Kitasuka, S. Tagashira, Density of Multipoint Relays in Dense Wireless Multi-hop Networks, in: *2nd International Conference on Networking and Computing (ICNC)*, Osaka, Japan, 2011.
- [16] L. Maccari, R. Lo Cigno, How to Reduce and Stabilize MPR sets in OLSR networks, in: *8th IEEE International Conference on Wireless and Mobile*

Computing, Networking and Communications (WiMob), Barcelona, Spain, 2012, pp. 381–388.

- [17] C. Barz, H. Rogge, Improved community network node design using a dlep based radio-to-router interface, in: 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, 2012.
- [18] L. Maccari, An analysis of the Ninux wireless community network, in: International Workshop on Community Networks and Bottom-up-Broadband (CNBuB), Lyon, France, 2013.
- [19] T. Clausen, C. Dearlove, P. Jacquet, U. Herberg, The Optimized Link State Routing Protocol Version 2, RFC 7181 (Apr. 2014).
- [20] P. Jacquet, A. Laouti, P. Minet, L. Viennot, Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad Hoc Wireless Network Models, in: 2nd IFIP-TC6 Networking Conference, Pisa, Italy, 2002.
- [21] A. Busson, N. Mitton, E. Fleury, Analysis of the multi-point relay selection in OLSR and implications, in: 4th Annual Mediterranean Ad Hoc Networking Workshop, le de Porquerolles, France, 2005.
- [22] M. Baysan, K. Sarac, R. Chandrasekaran, S. Bereg, A polynomial time solution to minimum forwarding set problem in wireless networks under unit disk coverage model, *IEEE Transactions on Parallel and Distributed Systems* 20 (7) (2009) 913–924.
- [23] B. Mans, N. Shrestha, Performance evaluation of approximation algorithms for multipoint relay selection, in: 3rd Annual Mediterranean Ad Hoc Networking Workshop, Bodrum, Turkey, 2004.
- [24] J. Lipman, P. Boustead, J. Chicharo, J. Judge, Resource aware information dissemination in ad hoc networks, in: 11th IEEE International Conference on Networks (ICON), Sydney, Australia, 2003.
- [25] P. Sondi, D. Gantsou, S. Lecomte, Design guidelines for quality of service support in optimized link state routing-based mobile ad hoc networks, *Ad Hoc Networks* 11 (1) (2013) 298 – 323.

- [26] O. Liang, Y. A. Sekercioglu, N. Mani, A survey of multipoint relay based broadcast schemes in wireless ad hoc networks, *IEEE Communications Surveys & Tutorials* 8 (4) (2006) 30–46.
- [27] J. H. Ahn, T.-J. Lee, Multipoint relay selection for robust broadcast in ad hoc networks, *Ad Hoc Networks* 17 (2014) 82 – 97.
- [28] A. B. Waluyo, D. Taniar, W. Rahayu, A. Aikebaier, M. Takizawa, B. Srinivasan, Trustworthy-based efficient data broadcast model for p2p interaction in resource-constrained wireless environments, *Journal of Computer and System Sciences* 78 (6) (2012) 1716 – 1736.
- [29] D. Gantsou, P. Sondi, S. Hanafi, Revisiting multipoint relay selection in the optimized link state routing protocol, *International Journal of Communication Networks and Distributed Systems* 2 (1) (2009) 4–15.
- [30] T. Kitasuka, S. Tagashira, Shared MPR sets for moderately dense wireless multihop networks, *International Journal of Distributed Sensor Networks*, (2015) 11 (6).
- [31] C. Adjih, P. Jacquet, L. Viennot, Computing connected dominated sets with multipoint relays, *Ad Hoc and Sensor Wireless Networks* 1 (1-2) (2005) 27–39.
- [32] J. Wu, An enhanced approach to determine a small forward node set based on multipoint relays, in: *58th IEEE Vehicular Technology Conference, Orlando, USA, 2003*.
- [33] J. Wu, W. Lou, F. Dai, Extended multipoint relays to determine connected dominating sets in MANETs, *IEEE Transactions on Computers* 55 (3) (2006) 334–347.
- [34] Y. Fu, X. Wang, W. Shi, S. Li, Connectivity based greedy algorithm with multipoint relaying for mobile ad hoc networks, in: *4th IEEE International Conference on Mobile Ad-hoc and Sensor Networks, 2008*.
- [35] Z. Liao, S. Liu, S. Xi, An Improved Multipoint Relaying Scheme for Message Propagation in Distributed Peer-to-Peer System, *International Journal of Distributed Sensor Networks* 10 (1).

- [36] G. Ciccarelli, R. Lo Cigno, Collusion in Peer-to-Peer Systems, *Computer Networks* 55 (15) (2011) 3517–3532.
- [37] Y. Faheem, J. L. Rougier, Loop avoidance for Fish-Eye OLSR in sparse wireless mesh networks, in: *IEEE International Conference on Wireless On-Demand Network Systems and Services (WONS)*, Snowbird, USA, 2009.
- [38] L. Maccari, R. Lo Cigno, Pop-Routing: Centrality-based Tuning of Control Messages for Faster Route Convergence, in: *IEEE International Conference on Computer Communications (INFOCOM)*, San Francisco (USA), 2016.
- [39] L. Maccari, L. Ghiro, A. Guerrieri, A. Montresor, R. Lo Cigno, On the Distributed Computation of Load Centrality and Its Application to DV Routing, in: *IEEE International Conference on Computer Communications (INFOCOM)*, Honolulu (USA), 2018.
- [40] L. Maccari, Q. Nguyen, R. Lo Cigno, On the Computation of Centrality Metrics for Network Security in Mesh Networks, in: *IEEE Global Communications Conference (Globecom)*, Washington DC, USA, 2016.
- [41] U. Sterner, U. Uppman, On the robustness of OLSR in a mobile tactical scenario in rural terrain, in: *International Conference on Military Communications and Information Systems (ICMCIS)*, Oulu, Finland, 2017, pp. 1–8.
- [42] J. Nilsson, U. Sterner, Performance of MPR-based broadcast methods in mobile ad hoc networks for rural terrains, in: *International Conference on Military Communications and Information Systems (ICMCIS)*, Cracow, Poland, 2015.
- [43] L. Maccari, R. Lo Cigno, Technical report TR-DISI-005, tuning the robustness of routing information diffusion with multi-point relays (2017).
URL <http://hdl.handle.net/11572/172141>
- [44] L. Maccari, R. Lo Cigno, Betweenness estimation in olsr-based multi-hop networks for distributed filtering, *Elsevier Journal of Computer and System Sciences special issue on Wireless Networks Intrusion* 80 (3) (2014) 670–685.
- [45] R. Clifford, A. Popa, Maximum subset intersection, *Information Processing Letters* 111 (7) (2011) 323–325.

- [46] G. Pei, T. R. Henderson, Validation of OFDM error rate model in ns-3 (2010).
URL <http://www.nsnam.org/~pei/80211ofdm.pdf>
- [47] C. Sommer, D. Eckhoff, R. German, F. Dressler, A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments, in: 8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS), Bardonecchia, Italy, 2011.
- [48] F. Qiu, X. Hu, Modeling group structures in pedestrian crowd simulation, *Simulation Modelling Practice and Theory* 18 (2) (2010) 190 – 205.
- [49] M. Musolesi, C. Mascolo, A community based mobility model for ad hoc network research, in: 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, Florence, Italy, 2006.