

A Restricted-Weakly Connected Dominating Set for Role Assignment in a Multichannel MAC for Wireless Mesh Network

Benoit Darties, Fabrice Theoleyre, Andrzej Duda

▶ **To cite this version:**

Benoit Darties, Fabrice Theoleyre, Andrzej Duda. A Restricted-Weakly Connected Dominating Set for Role Assignment in a Multichannel MAC for Wireless Mesh Network. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), 2009, Morocco. pp.167-172. hal-00618338

HAL Id: hal-00618338

https://u-bourgogne.hal.science/hal-00618338

Submitted on 1 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A restricted-Weakly Connected Dominating Set for Role Assignment in a Multichannel MAC for Wireless Mesh Network

Benoît Darties, Fabrice Theoleyre, and Andrzej Duda
Grenoble Informatics Laboratory (LIG), CNRS, Grenoble INP
681 rue de la Passerelle, BP 72, 38402 Saint Martin d'Heres Cedex, France
Email: {firstname.lastname}@imag.fr

***Abstract*—We propose an efficient way of constructing the wireless mesh structure associated with *Molecular MAC*, a multichannel access method designed for efficient packet forwarding. We base our role assignment on a restricted Weakly Connected Dominating Set structure. After presenting a formal definition of the role assignment problem, we prove its NP-completeness. Then, we propose a centralized 2-approximation algorithm that maximizes the sum of radio link capacities in the molecular structure. Finally, we extend this protocol so that it can operate in a distributed way still providing the same guarantee. This distributed protocol is self-stabilizing thus robust to topology changes. Our simulation results show that the 2-approximation distributed protocol provides an improvement in throughput with respect to other protocols.**

***Index Terms*—wireless mesh networks; Molecular MAC; WCDS; 2-approximation;**

I. Introduction

We consider *wireless mesh networks* that use IEEE 802.11 wireless LANs for interconnecting mesh routers. When mesh routers use the legacy IEEE 802.11 networks with a single interface, the performance of packet forwarding quickly degrades with the number of hops due to channel contention and spatial problems such as hidden, exposed, masked, and blocked nodes [1]. We can observe that the capacity of a wireless mesh network strongly depends on the ability of nearby mesh routers to communicate in parallel, which is only possible if neighbor routers that may interfere use different channels.

One way of improving performance is to use multiple non-overlapping channels so that stations can transmit in parallel and without collisions. When a mesh router has several radio interfaces, it can tune them to different channels and use them simultaneously. The main problem in this approach is thus to assign static channels to interfaces in a way that maximizes network capacity, either with a centralized [2] or a distributed algorithm [3].

A more dynamic approach consists of using a MAC layer allowing each node to switch to a different channel on demand. In this way, the MAC layer can distribute load more uniformly over channels thus achieving better throughput. Moreover, the approach may benefit to stations with single or multiple interfaces. An example of such a method is MMAC (Multi-Channel MAC) [4] that proposed periodic Rendezvous Points during which nodes reserve channels to use later for data transmission. Molecular MAC [5] defined a *molecular architecture* based on allocating roles and channels to mesh nodes according to the analogy with *atoms* containing a *nucleus* and *electrons*. This approach requires a specific MAC mechanism to support multichannel multiplexing for efficient packet forwarding and results in significant performance improvement over MMAC.

In this paper, we address the problem of constructing the structure associated with Molecular MAC: it requires allocating different roles (*nucleus* or *electron*) to mesh nodes. Although such a structure is related to the problem of clustering, Molecular MAC imposes some specific constraints: we must guarantee global connectivity along with the requirement of alternating the roles of nodes—neighboring nodes that directly communicate must not have the same role so they do not experience the *deafness* problem when a node tries to send a frame on one channel while the intended receiver is listening or sending on another one. In addition to that, we need to maximize network capacity.

The contributions of this paper are threefold:

- we formally define the problem of role assignment and show its NP-completeness in a general case,
- we propose a centralized near-optimal algorithm for finding the required structure for Molecular MAC that guarantees the number of links closed to the optimal by the factor of 2 (such a guarantee is called *2-approximation*); the algorithm has running time $O(m)$, where m is the number of links;
- we also propose a distributed version of this 2-approximation that cope with node and/or link failures.

II. Molecular MAC Overview & Motivation

The present paper aims at defining protocols for constructing a suitable network structure required for Molecular MAC. Molecular MAC proposes to extend IEEE 802.11 to efficiently use multichannel nodes in multi-hop networks. It achieves efficient dynamic multichannel multiplexing along with solving the deafness problem.

This work was partially supported by the European Commission project WIP under contract 2740, the French Ministry of Research project AIRNET under contract ANR-05-RNRT-012-01 and ARESA under contract ANR-05-RNRT-01703.

The idea of Molecular MAC is to partition a mesh network into spatially distinct *atoms* that are similar to traditional IEEE 802.11 infrastructure mode cells: a *nucleus* node in each cell always uses a channel different from interfering nuclei. Nodes between two nuclei, which we call *electrons*, dynamically switch channels to communicate with neighbor nuclei. There is no direct link between two electrons so to communicate, they use a nucleus like an access point in a 802.11 WLAN. Two nuclei do not directly communicate neither, because they use different channels. If they need to communicate, they can do it through neighboring electrons.

Molecular MAC uses a nucleus as a kind of a *virtual access point*: it is either a receiver or a transmitter of any frame in an atom. Packet forwarding relies on electrons that switch between channels of all neighboring atoms. This scheme assigns one channel to each nucleus and all electrons use it for communication with the nucleus (for both reception and transmission). By choosing different channels, two neighboring atoms that would have interfered, if they had used the same channel, can limit interference problems, which contributes to achieving higher capacity.

In Molecular MAC, a nucleus chooses its channel according to the load measured locally, announces it to all electrons, and stays tuned to it. An electron belonging to several atoms knows the channels of its nuclei and can switch to a given channel when it wants to communicate.

For efficient packet forwarding an electron explicitly requests a data frame from a nucleus by sending a special control frame, a `pull` that acts a little bit like a `Clear To Send` (CTS) reservation frame. Each nucleus piggybacks the list of pending destinations in each data frame so that the electrons know when sending a `pull` is relevant. An electron periodically scans each atom it belongs to for pending packets. Moreover, if some packets are pending, it randomly chooses to send or receive one. A nucleus maintains activity by sending an empty data frame to implement a quick notification. When a node wants to join the mesh, it performs multichannel neighborhood discovery to detect eventual neighboring nuclei and become an electron. Otherwise, it takes the role of a nucleus.

Figure 1 illustrates the structure of a molecular mesh. The identifiers of nuclei are numbers while they are letters for electrons. We can note that there is no link between two electrons nor between two nuclei. However, we need to keep the number of unused links small to allow for redundant paths in the network for better connectivity and fault-tolerance.

Note that in this paper, we limit its scope to mesh nodes with a single radio interface, however we can easily generalize the results to the case of multiple interfaces.

III. Related Work

The problem of constructing a molecular mesh relates to the problem of clustering for which many efficient distributed algorithms exist. We can elect clusterheads with a Maximum Independent Set [6], however in some cases, clusterheads can

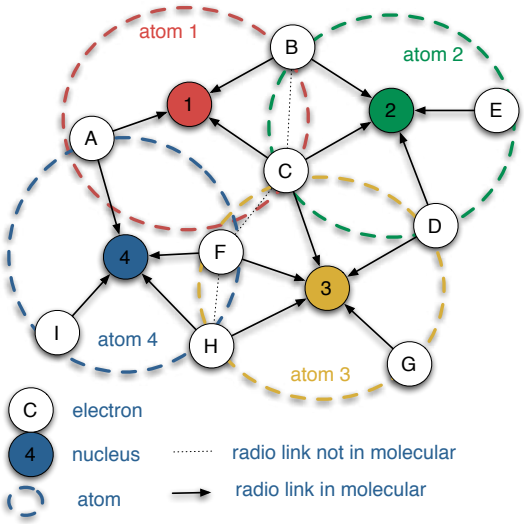


Fig. 1. Example of a molecular structure.

be exactly three hops apart, which means that under Molecular MAC the network becomes disconnected.

Many authors studied the Weakly Connected Dominating Sets (WCDS) and applied them to various network problems such as routing or pre-shared key distribution [7]. Construction of a WCDS consists of electing *dominators* so that each node is a neighbor of at least one dominator and the set of edges for which one endpoint is a dominator forms a connected set. The problem of finding a WCDS with minimum cardinality is NP-hard [8]. Chen et al. [9] extended the centralized algorithm of Guha et al. [10]: in each step, they choose the best node to color, the component forming the WCDS being larger for each round. Other authors proposed a divide-and-conquer approach to reduce the time complexity of the algorithm [11], [12]. The concept of WCDS appears as a suitable structure for electing electrons and nuclei, however, we cannot use it as is since two nuclei cannot directly communicate in Molecular MAC to avoid deafness. Thus, Molecular MAC leads to a structure similar to a WCDS in which we remove edges between dominators.

In our previous work, we have first focused on functional MAC aspects of the molecular approach [5]. In particular, we have evaluated the performance of the proposed Molecular MAC and shown its efficiency for multi-hop packet forwarding. Then, we have considered the problem of constructing a suitable mesh structure and proposed a centralized optimal algorithm through MILP (Mixed Integer Linear Programming) formulation and two heuristic protocols [13]. In this paper, we extend the last work by proposing a protocol that achieves 2-approximation: in any case, it guarantees at least 50% performance of the optimal centralized algorithm, the performance objective being the sum of link capacities in the constructed molecular mesh network.

IV. Problem Definition

We consider the problem of constructing a mesh network suitable for Molecular MAC. First, we introduce our notations:

- we model the network as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ of n vertices and m edges, where vertices $\mathcal{V}$ are nodes and edges $\mathcal{E}$ are communication links between nodes.
- $N(u)$ denotes the neighbors of u (excluding u).
- A weight function $w : \mathcal{E} \mapsto \mathcal{R}$ assigns the capacity of a link to each edge in the graph.

In Molecular MAC, a link may exist between two nodes if and only if its endpoints have different roles (nucleus or electron). Maintaining network connectivity implies the existence of at least one path between each pair of nodes that alternates electrons and nuclei, i.e. the union of edges $(nucleus, electron)$ must form a bipartite connected graph.

We aim at maximizing the sum of link capacities while ensuring network connectivity. Thus, we should assign the roles so that the resulting graph maximizes the sum of radio link capacities between nuclei and electrons. However, simultaneously assigning roles and channels in a distributed way and attaining this objective is difficult. Thus, a good trade-off between performance and complexity would be to split the problem into two parts solved sequentially:

1) first define node roles regardless of channel assignment that would minimize interference;
2) then allocate channels to nuclei in a way that minimizes interference. This allocation represents the familiar coloring problem widely studied in the literature [14].

The first part corresponds to our main objective in this paper.

Formally speaking, finding the optimal role assignment that maximizes the sum of edge capacities regardless of interference corresponds to finding the following subgraph that we call a Maximum Weighted Edge Connected Bipartite Subgraph (Max-WECBS) defined as follows:
Input: a connected graph $G = (V, E)$ and a weight function $w : E \Rightarrow \mathcal{R}$
Output: a bipartite connected graph $G_B = (X, Y, E')$, where $X \cup Y = V$, $E' \subseteq E$ and $E' = \{(x, y)\}/x \in X \ and \ y \in Y$ (set X contains electrons while set Y contains nuclei).
Objective: maximize $\sum_{e \in E'} w(e)$.

The considered problem also relates to that of Maximum Weighted Cut (Max-WC), the weighted version of the Maximum Cut [15] that partitions the vertices of a graph into two subsets by maximizing the number of edges with one endpoint in each subset. If we only consider instances with strictly positive weights, then Max-WC and Max-WECBS lead to the same objective: indeed one can show that the edges of any maximum weighted cut necessarily form a bipartite connected graph. However, they differ with respect to the fact that Max-WC does not have the connectivity constraint whereas it appears in Max-WECBS. Thus, approximation protocols for Max-WC do not necessarily lead to valid solutions for Max-WECBS. Note also that fully local 2-approximation protocols exist for Max-WC. On the contrary, we have mentioned in the previous work [13] that there is no fully localized protocol for constructing a Max-WECBS. If we accept negative or zero weights, both problems are clearly different since the optimal solution for Max-WC does not necessarily results in a connected component.

We conclude this section with the following theorem:

Theorem 1: Max-WECBS is NP-complete.

Proof: The proof follows from a reduction from the Maximum Cut problem. Recall that Maximum Cut is NP-complete and hard to approximate [15] (the weighted version is as hard to approximate as the original problem [16]).

Let a connected graph $G = (V, E)$ be an instance of Max-Cut and let $E' \subseteq E$ be the edges belonging to the optimal solution. Then $G' = (V, E')$ is bipartite. Moreover, G' is necessary connected, otherwise E' will not be maximum. Thus, G' is also a solution of Max-WECBS on the instance given by G and by the weight function that assigns the same strictly positive weight to all edges. Reciprocal is also straightforward. ■

Taking into account this fact we propose an algorithm that approximates the optimal solution for constructing a Max-WECBS.

V. 2-APPROXIMATION FOR MAX-WECBS

This section describes a centralized algorithm that provides 2-approximation for Max-WECBS with the computational time of $O(m)$, where m is the number of edges.

Given input graph $G = (V, E)$ and weight function w, we construct partition $\{X, Y\}$ of V such that bipartite subgraph $G' = (X, Y, E' = \{\{u, v\} | u \in X, v \in Y\})$ is connected (set X contains electrons while set Y contains nuclei). We initially place nodes in set $R = V$ while both X and Y are empty. The algorithm starts with choosing a random node from R and placing it in X. During the iterative part, we successively take nodes from R and place them in one of the sets X or Y according to the criteria described below. To maintain the connectivity of the resulting bipartite subgraph, the protocol only chooses nodes from R incident to at least one node from $X \cup Y$.

To decide about the set (X or Y) for placement of node u, the algorithm computes two values: $w_X(u)$ and $w_Y(u)$ corresponding to the sum of weights of the edges between u and nodes belonging to X for $w_X(u)$, respectively to Y for $w_Y(u)$. The algorithm decides to place node u in X if $w_X(u) < w_Y(u)$ and in Y otherwise. Thus, at each iteration, the sum of weights of the new edges included in the solution is greater or equal to the sum of weights of forbidden edges (i.e. edges between the newly included node and nodes in the same set).

Obviously, the resulting bipartite graph will be connected. Indeed, let assume by induction that the component of nodes already in $X \cup Y$ is connected. The algorithm chooses a neighbor of a node already either in X or in Y (i.e. $w_Y(u) > 0$ or $w_X(u) > 0$). If node u is placed in X (respectively in Y), at least one neighbor of the component is in Y (respectively in X). This edge will be kept in the bipartite graph and the new component remains connected.

At the end of the execution, the sum of weights of edges added to the solution is at least the half of the sum of

weights of all the edges of the graph, which gives us a 2-approximation. Algorithm 1 presents a formal specification.

Algorithm 1: 2-approximation for Max-WECBS

input : Connected graph $G = (V, E)$, weight function $w : E \Rightarrow \mathcal{R}$
output: Partition $\{X, Y\}$ of V inducing a connected bipartite subgraph of G

```
1  begin
2      Initialization of the algorithm;
3      for u ∈ X do
4          w_X(u) = 0;
5          w_Y(u) = 0;
6      randomly choose u ∈ V;
7      R ⟵ V − {u};
8      X ⟵ {u};
9      Y ⟵ ∅;
10     for v ∈ N(u) do
11         w_X(u) = w({u, v});
12     iterative part;
13     while R ≠ ∅ do
14         choose u ∈ R such that N(u) ∩ {X ∪ Y} ≠ ∅;
15         R ⟵ R − {u};
16         if w_X(u) ≥ w_Y(u) then
17             Y = Y ∪ {u};
18             for v ∈ {R ∩ N(u)} do
19                 w_Y(v) = w_Y(v) + w({u, v});
20         else
21             X = X ∪ {u};
22             for v ∈ {R ∩ N(u)} do
23                 w_X(v) = w_X(v) + w({u, v});
24 end
```

Let us analyze the computational time of Algorithm 1: it executes the initialization part (lines 2-11) in $O(n)$ time complexity. The iterative part (lines 12-23) runs exactly n times and each part can be executed in $O(\delta(u))$ time complexity, where u is the current selected node and $\delta(u)$ is its degree in G. Thus, the protocol achieves the running time of $O(\sum_{u \in V} \delta(u))$, i.e $O(m)$, where m is the number of edges.

VI. Protocol version of the 2-approximation

While Algorithm 1 is efficient and finds a 2-approximation, it is centralized so we cannot directly apply it to multi-hop networks. Moreover, it does not take into account topology changes that would require an entire re-execution. In this section we present a distributed version of the algorithm—a protocol achieving the 2-approximation in a mesh network: nodes periodically broadcast `hello` packets and determine their own role from `hello` packets transmitted by adjacent nodes. Since in a realistic radio environment packets may be lost due to bad channel conditions, fading, or collisions, and the network topology may vary in time, we propose a self-stabilizing version so that the protocol converges to a valid solution in spite of lost packets or added as well as removed nodes.

A. Description

Centralized Algorithm 1 requires selecting a node at random at the execution startup (line 6). In the distributed version, the node with the highest id becomes the startup node called also a "leader". Then, we construct an order in the network by considering the instant at which a node decides on its role. This instant is a variable local to each node so that nodes do not require global synchronization. A node decides on its role in function of the role decided by its neighbors.

1) Election of the leader: As multi-hop networks usually use `hellos` for neighbor discovery, we add the fields `leader-id` and `seq-num` required by our protocol to existing `hellos`. A node includes in the `leader-id` field the highest id received so far or its id, if it is greater than the received values (in this case the node considers itself as a leader). The leader increments `seq-num` sent in each `hello` packet so that nodes can detect its departure: if `seq-num` does not increase for e.g. 3 `hello` periods, nodes consider `leader-id` as invalid.

2) Creation of an order: To guarantee the 2-approximation, nodes have to make consistent decisions, even if some `hellos` are lost. Consequently, we create an order through the `decision-date` variable that represents the instant at which a node decides on its role. It is only a local variable so nodes do not need to synchronize.

3) Decision on a role: When a node decides on its role, it takes into account the role of all neighbors that have already taken a role, i.e. their `decision-date` is lower. It chooses the role that maximizes the capacity of links with neighbors with a different role.

4) Protocol operation: The leader node chooses the role of a nucleus, sets its `decision-date` to 0, and broadcasts its decision in a `hello` packet. Each node maintains a neighborhood table called `originator-table` fed by received `hellos`. When a node receives a `hello` with a higher `leader-id`, it proceeds as follows:

1) it activates a timer set to a random value $t_{backoff}$ ($t_{backoff} \in (0..\Delta_{\texttt{decision-date}}]$, where $\Delta_{\texttt{decision-date}}$ is a configurable parameter),
2) it sets its own `decision-date` to the `decision--date` of the sender $+t_{backoff}$,
3) after $t_{backoff}$, the node decides on its role: it must maximize the sum of the weights of all radio links to a neighbor with a different role and a lower `decision--date`,
4) it broadcasts its new role and `decision-date` in a `hello`,
5) the node stores the list of all neighbors with an earlier `decision-date` than itself at the instant of its decision in `originator-table`. It is important to note that only these nodes infuence its decision.

B. Dealing with inconsistent decisions

Nodes may make inconsistent decisions because of packet losses, if a node does not know that a neighbor has already chosen a role. However, the `decision-date` variable helps to detect such situations: a node detects that a neighbor is not in `originator-table`, but it has an earlier `decision--date` than itself. If `decision-date` values are equal, the

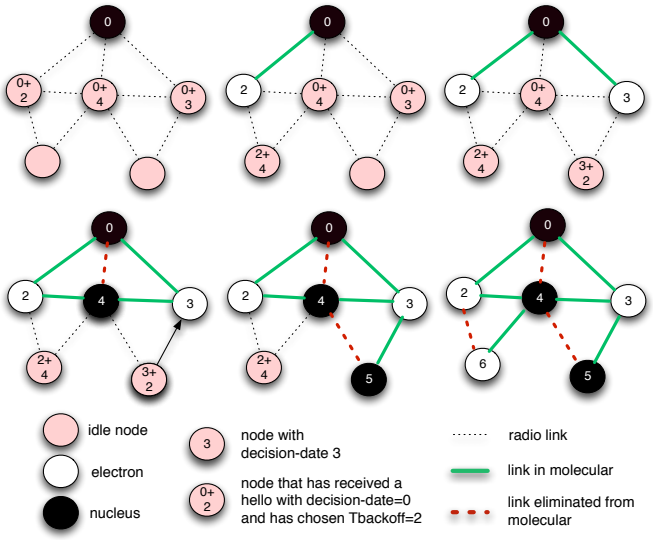


Fig. 2. Execution of the protocol on a simple topology.

node will compare the ids to break the tie. This means that the list of nodes that influenced its decision changed and the node needs to reconsider its role by following the rules below:

1) it adds the conflicting neighbor to `originator-table`,
2) it reconsiders its role by taking into account all the nodes in the updated `originator-table`,
3) if its role has changed, it immediately sends a `hello`.

Since such situations may lead to cascading changes, a node has to react to role changes. In particular, if a neighbor of a given node present in the `originator-table` has changed its decision, the node must reconsider its own decision in the same way by applying the rules above. Note that the rules result in a self-stabilizing behavior of the distributed protocol.

C. Illustration

Let us illustrate the execution of the protocol on a simple topology presented in Figure 2. For the sake of simplicity, we assume that all radio links have the same weight and we name each node by its `decision-date`. We only show how nodes decide on their roles so we assume that the node at the top is the leader.

First, the leader becomes a nucleus at the `decision-date` of 0 and its neighbors increment their `decision-date` by random $t_{backoff}$. The node that has chosen the backoff of 2 decides first: it sets its `decision-date` to 2, chooses the inverse role of the leader since it has only one neighbor, and notifies its neighbors with a `hello`. The neighbor below 2 sets its `decision-date` to 2 and adds a random backoff of 4. Node 3 makes a similar decision to node 2. In the next step node 4 chooses to be a nucleus, because becoming an electron results in eliminating links with 2 and 3 while being a nucleus only eliminates one link to the leader. Finally, the network forms a valid Max-WECBS: the resulting molecular structure contains at least half of links and it is connected.

D. Fault tolerance

If the network topology changes due to link failures as well as leaving or arriving nodes, the protocol converges to a valid state by reacting to the changes in the same way as for inconsistent decisions. A node just has to lookup its neighbors in `originator-table` and re-apply its decision rule as soon as it detects any change. Besides, a topology change may require reordering decisions: it happens when a node has the local maximum `decision-date`, but it is not the leader. It would potentially break the connectivity of the bipartite graph. In this case, the node looks for the minimum `decision-date` among its neighbors and increments it by a random value ($rand \in (0..\Delta_{\texttt{decision-date}})$). Then, it just has to decide on its new role according to the same rule and updates `originator-table`. Using the minimum (and not the maximum) `decision-date` limits the number of decision changes among the neighbors: if the role does not change, only the node with the minimum `decision-date` has to re-consider its decision.

As mentioned previously, a node detects the departure of the leader: if the sequence number does not increase in the received `hellos`, a node can safely consider the leader dead. Thus, nodes will re-consider their decision after the propagation of the new maximum id.

We can observe that this protocol is self-stabilizing, but we omit here the proof due to space limitation. However, it is quite straightforward by considering all the cases that may occur and proving the protocol convergence in each situation if no topology or decision change occurs during a sufficiently long time.

VII. Performance Evaluation

We have simulated the 2-approximation protocol in Ws-Net [17] using the COIN-CBC linear programming library [18]. We randomly place nodes in a circular simulation area. We assume that nodes use IEEE 802.11a network cards with the radio range of 10 units and the interference range of 30 units. We adjust the simulation area to maintain an average density of 10 neighbors.

The results correspond to statistics averaged over 10 different simulations of 240 seconds. The graphs present averaged values with 95% confidence intervals. We compared the proposed 2-approximation with a centralized optimal (OPT) solution, the Maximum Independent Set protocol (MIS), and a self-stabilizing Spanning Tree (ST) (introduced in the earlier work [13]). Note that the OPT strategy results in the best solution, however it is intractable for medium to large networks (> 40 nodes).

Since all these protocols use uniquely `hellos` for their decisions, the reader can note that they present exactly the same overhead.

First, we present in Figure 3 the route stretch factor (i.e. the ratio between the length of the shortest route via the links

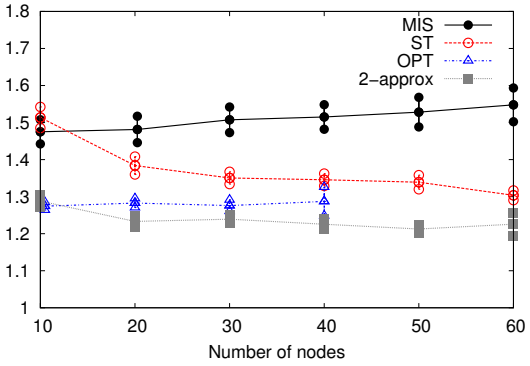


Fig. 3. Route stretch factor for a varying number of nodes in a random mesh network.

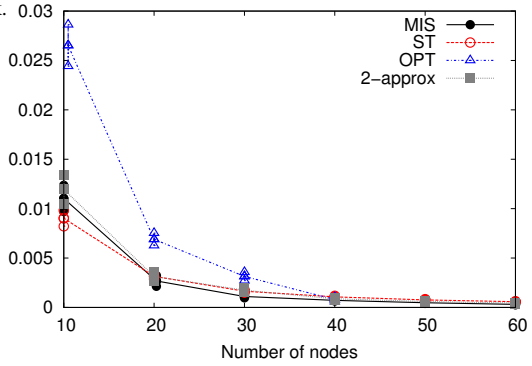


Fig. 4. Minimum throughput T_{min} for a varying number of nodes in a random mesh network.

$(edge, nucleus)$ and the length of the shortest route in the original graph). We can note that the MIS and ST strategies use long routes while 2-approx achieves the lowest route stretch factor since it maximizes the number of radio links to keep in the resulting structure. The OPT strategy uses a little longer routes than the 2-approximation since it directly maximizes the global throughput: although both objectives are correlated, they are not entirely identical.

Then, we consider the minimum guaranteed throughput for an increasing number of nodes (cf. Figure 4). The OPT strategy computes this throughput as the objective of the MILP formulation (cf. [13]). We directly apply here the same MILP formulation with different roles computed by different strategies and extract the objective value. We can observe that maximizing the number of radio links in the molecular structure really improves the throughput of the structure. The OPT strategy obtains the highest throughput, but it is intractable for larger networks. Our 2-approximation outperforms the ST and MIS protocols. Even if the objective of Max-WECBS is to maximize connectivity, it also results in improved throughput of the molecular structure.

VIII. Conclusion and Future Work

In this paper, we have formally defined the structure associated with the Molecular MAC and proposed a centralized 2-approximation protocol that maximizes the number of radio links in the molecular structure. We then have extended this protocol so that it can operate in a distributed way still providing the same guarantee. Moreover, it is self-stabilizing thus robust to topology changes.

We are currently working on comparing the performance of this approach with other strategies that do not provide guarantees. Moreover, unlike the present work in which we assign roles first and then allocate channels, we consider assigning jointly channels and roles to develop protocols with the global guarantee of performance.

References

[1] C. Chaudet, D. Dhoutaut, and I. Guerin Lassous, "Performance Issues with IEEE 802.11 in Ad Hoc Networking," *IEEE Communications Magazine*, vol. 43, no. 7, 2005.

[2] H. L. Nguyen and U. T. Nguyen, "Channel Assignment for Multicast in Multi-channel Multi-radio Wireless Mesh Networks," *Wireless Communications and Mobile Computing*, 2008.

[3] S. Avallone and I. Akyildiz, "A Channel Assignment Algorithm for Multi-Radio Wireless Mesh Networks," *Computer Communications*, vol. 31, no. 7, 2008.

[4] J. So and N. H. Vaidya, "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver." in *MOBIHOC*. Japan: ACM, 2004.

[5] M. Nassiri, F. Theoleyre, M. Heusse, and A. Duda, "Molecular architecture for autonomic wireless mesh networks," in *MASS*. Macau, Macau: IEEE, October 2009.

[6] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[7] A.-S. K. Pathan and C. S. Hong, "A key-predistribution-based weakly connected dominating set for secure clustering in dsn," in *HPCC*, LNCS, Ed., vol. 4208, Munich, Germany, September 2006, pp. 270–279.

[8] J. E. Dunbar, J. W. Grossman, J. H. Hattingh, S. T. Hedetniemi, and A. A. McRae, "On weakly connected domination in graphs," *Discrete Mathematics*, vol. 167-168, 1997.

[9] Y. P. Chen and A. L. Liestman, "Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks," in *MOBIHOC*. Lausanne, Switzerland: ACM, June 2002, pp. 165–172.

[10] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets." *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[11] Y. P. Chen and A. L. Liestman, "Maintaining weakly-connected dominating sets for clustering ad hoc networks," *Ad Hoc Networks*, vol. 3, pp. 629–642, 2005.

[12] B. Han and W. Jia, "Clustering wireless ad hoc networks with weakly connected dominating set," *Journal of Parallel and Distributed Computing*, vol. 67, no. 6, pp. 727–737, June 2007.

[13] F. Theoleyre, B. Darties, and A. Duda, "Assignment of roles and channels for a multichannel mac in wireless mesh networks," in *ICCCN*. San Francisco, USA: IEEE, August 2009.

[14] A. Kosowski and L. Kuszner, "On Greedy Graph Coloring in the Distributed Model." in *Euro-Par*, ser. LNCS, vol. 4128. Dresden, Germany: Springer, August 2006.

[15] P. Alimonti, D. I. Sistemistica, and V. Kann, "Hardness of Approximating Problems on Cubic Graphs," in *Theoretical Computer Science*. Springer-Verlag, 1997, pp. 288–298.

[16] P. Crescenzi, R. Silvestri, and L. Trevisan, "To Weight or Not To Weight: Where is the Question?" in *ISTCS*. Jerusalem, Israel: IEEE, June 1996, pp. 68–77.

[17] G. Chelius, A. Fraboulet, and E. Fleury, "Wsnet: a modular event-driven wireless network simulator," http://www.worldsens.net/, 2006.

[18] Coin-or branch and cut, "https://projects.coin-or.org/Cbc."