A Divide-and-Conquer Scheme for Assigning Roles in Multi-Channel Wireless Mesh Networks

Benoît Darties, Fabrice Theoleyre, and Andrzej Duda
Grenoble Informatics Laboratory, Grenoble, France
Email:{darties, theoleyr, duda}@imag.fr

Abstract—A multi-channel MAC is a promising approach for improving network throughput by multiplexing transmissions over orthogonal channels. Molecular MAC has recently adopted this approach by proposing to modify the standard IEEE 802.11 DCF. It requires role and channel assignment to nodes; some of them use a static channel while others dynamically switch to neighbor channels on-demand. To assign roles and channels, we extend the notion of the Weakly Connected Dominating Set, a structure already used in clustering. We adapt it by introducing new constraints to define what we call a reversible WCDS (r-WCDS), which is particularly suitable for Molecular MAC. We propose a divide-and-conquer scheme that partitions the network into clusters with one leader per cluster solving a MILP formulation to assign roles in its cluster. By appropriately defining the roles at the border of clusters, we maintain global connectivity in the r-WCDS. Our simulations show that the performance of our strategy is close to a centralized algorithm.

Index Terms—Wireless Mesh Networks, Multi-channel MAC, WCDS, MILP, distributed algorithms.

I. INTRODUCTION

We consider wireless mesh networks composed of wireless routers providing connectivity to mobile nodes. They begin to emerge in some regions to provide cheap network connectivity to a community of end users. Usually they grow in a spontaneous way when users or operators add more routers to increase capacity and coverage.

To improve network capacity, wireless mesh networks can use multiple radio channels. Nodes with multiple interfaces can statically allocate channels to achieve high spatial reuse and good performance [1]. Nodes with single interfaces can also benefit from multiple channels by switching channels on a per frame basis [2]. Molecular MAC [3] has recently proposed to modify the standard IEEE 802.11 DCF access method to use dynamic channel switching for efficient packet forwarding over multiple hops. It solves the deafness problem inherent to multi-channel schemes by assigning a static channel for one part of nodes (nuclei of spatially distinct atoms) and letting other nodes (electrons) dynamically switch between channels. Electrons initiate transmissions and nuclei notify other nodes about pending packets. Molecular MAC outperforms classical strategies like MMAC [2] with respect to throughput, fairness, and end-to-end delay. The authors have left the role (nucleus or electron) and channel assignment problem for future work.

In this paper, we propose a protocol for organizing a wireless mesh network according to a suitable structure associated with Molecular MAC. We adapt the well-known Weakly Connected Dominating Set (WCDS) structure [4] and introduce new constraints to define what we call a reversible WCDS (r-WCDS), particularly suitable for networks operating under Molecular MAC. We provide a formal definition of the r-WCDS and propose a new divide-and-conquer scheme for constructing such an r-WCDS in a distributed way.

The paper is organized as follows. First, we introduce notations and define the problem of constructing an r-WCDS. Section III shortly describes a Mixed Integer Linear Programming formulation (MILP) to find an r-WCDS. Then, we present a divide-and-conquer scheme for constructing an r-WCDS in a scalable way. We evaluate the performance of this scheme in Section V and discuss the related work in Section VI.

II. PROBLEM FORMULATION AND NOTATION

We model the network as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ in which vertices $\mathcal{V}(G)$ are the set of nodes and edges $\mathcal{E}(G)$ are all pairs of nodes able to directly communicate. We adopt the following classical notation:

- $n = |\mathcal{V}|$ defines the number of nodes in the mesh network,
- \{u, v\} denotes the edge between vertices u and v, i.e. \{u, v\} $\in$ $\mathcal{E}$, while $N(u)$ is the set of neighbors of u with cardinality $\Delta(u) = |N(u)|$,
- $CH$ is the set of all available channels and $nbCH = |CH|$. Finally, $BW$ denotes channel capacity.

We define the reversible WCDS (denoted r-WCDS in the rest of the paper) as follows: nodes are splitted into two non-overlapping subsets dominators and dominatees. We only keep the edges (dominator,dominatee), i.e. edges between two dominators or two dominatee are removed, and the resulting structure has to remain connected. Formally, let D be the set of dominators such that:

$$\forall u \in \{V - D\}, \exists v \in D \mid v \in N(u) \quad \mathcal{G} = (\mathcal{V}, \mathcal{E}') \text{ connected} \mid \mathcal{E}' = \{\{u, v\}, u \in D, v \in \{V - D\}\}.$$ 

We are interested in the reversible WCDS, because Molecular MAC [3] requires such a structure. Molecular MAC divides a wireless mesh network into spatially distributed atoms so that each atom uses a fixed channel different from its neighbors. An atom is composed of a nucleus and electrons. A nucleus chooses a channel for its atom and sticks to this channel all the time. Nodes at the border of atoms have the role of electrons bonding neighboring atoms: they forward traffic between atoms by dynamically switching their channel to communicate with neighboring nuclei.
In Molecular Mesh, we need to assign a role to each node (a nucleus or an electron) so that the resulting network has the following properties:
1) a node can communicate with any other node via multi-hop forwarding;
2) only nuclei and electrons can communicate with each other, i.e. there is no direct communication between two electrons or two nuclei;
3) the capacity of the network should be maximal. In particular, two neighboring atoms, which can interfere, need to use different channels.

We can note that links between two electrons are not used, however we need to keep the number of unused links small to allow for redundant paths in the network for better connectivity and failure tolerance. Clearly, the constraints for assigning roles to nodes lead to a reversible-WCDS: each dominator corresponds to a nucleus and an electron to a dominatee.

Constructing the molecular structure requires that each nucleus node selects a fixed channel. Simultaneous assignment of roles and channels so to maximize network capacity is a difficult problem. A good tradeoff between computation time and network performance consists first of determining the role of a node to obtain a connected component regardless of channel usage. When a node becomes nucleus, it chooses a channel according to a greedy approach by scanning available channels and choosing the one with the minimum load.

### III. MILP FORMULATION

We use a MILP (Mixed Integer Linear Programming) formulation already defined in our previous work [5] to assign roles (i.e. finding an r-WCDS) while maximizing the network capacity. Its solution leads to the optimal assignment of roles (nucleus or electron) and channels.

Basically, the MILP formulation assumes that each node communicates with each other node (total $n(n - 1)$ flows) and assigns bandwidth per flow assuming a max-min objective (guaranteed bandwidth per flow). We have formulated classical flow conservation constraints and added constraints on radio resource sharing to model interference. Finally, some constraints forbid nodes with the same role to communicate with each other.

### IV. POTATOES: A DIVIDE AND CONQUER SCHEME

Solving the MILP is computationally expensive for large networks (more than 40 nodes). In this paper, we propose a divide-and-conquer scheme: we divide the network into clusters with one leader per cluster that solves the MILP formulation for its cluster. The small size of clusters helps to efficiently obtain the local MILP solution. However, we need to enforce additional constraints so that the union of multiple local r-WCDSs results in a global connected r-WCDS: we have to define clusters in a certain manner and fix the roles of some node. We provide below the main lines of the mechanisms for constructing clusters and achieving the global r-WCDS (for more details see [6]).

#### A. Approach

We first construct a rooted cluster tree, i.e. a tree in which vertices are clusters and a link exists between clusters if and only if they share a node. The rooted cluster tree supports distributed role assignment: one leader per cluster computes the optimal local assignment in its cluster. We need to limit the dependence between two clusters, i.e. a node should receive its role from only one leader. Thus, we add the following constraints with respect to classical cluster-trees:

1) One node belongs to at most two clusters. A node that belongs to exactly two clusters is a cluster member of the cluster higher in the tree and the leader in the other;
2) two clusters are connected via at most one node.

Figure 2 illustrates the cluster-tree structure we could obtain.

The leader of each cluster collects the information on radio links of all cluster members: each node includes in its hellos all its radio links and the radio links of its descendants in the tree. Then, it computes the local optimal solution with the MILP formulation: all the constraints are translated into linear inequalities and the global objective consists in maximizing the throughput. After computing the roles for its cluster, the leader has just to notify its cluster members about their roles.
The assignment is optimal in one cluster, but the union of local assignments does not necessarily leads to a global optimal r-WCDS, because leaders find the optimal role assignment inside clusters and not among clusters. We need to enforce that nodes belonging to different clusters end up with the same role in the global r-WCDS. To achieve this goal, we use the hierarchy of the cluster-tree and proceed in the following way:

1) the leader assigns a role to all its cluster members;
2) a node belonging to one cluster uses the assigned role;
3) a node that belongs to more than one cluster receives its role from the leader upper in the hierarchy (e.g. B receives its role from A in Figure 2) and uses this role in its own cluster (e.g. B in cluster 2).

Finally, at least one path exists in the rooted cluster-tree that uses the hierarchy of clusters. We will now define more formally this algorithm.

B. Cluster-Tree Construction

We will use the following definitions:

- a cluster is a connected subgraph Cluster of network graph \( G \) induced by the subset of nodes \( S \subseteq V(G) \) with \( V(Cluster) = S \) and \( E(Cluster) = \{(u,v)\mid u,v \in V(G) \text{ and } \{u,v\} \in E(G)\} \);
- \( T \) denotes the spanning tree used to build the cluster-tree;
- \( CT \) represents the cluster-tree structure;
- network leader node denoted by RootLeader is the root of spanning tree \( T \);
- \( V(Cluster) \) is the set of nodes belonging to \( Cluster \);
- Leader(Cluster) denotes the leader of \( Cluster \). If RootLeader \( \notin \) Cluster, Leader(Cluster) is the node belonging to Cluster and to the upper cluster in the cluster-tree. Formally, Leader(Cluster) = \( V(Cluster) \cap V(parent(Cluster)) \) with parent(Cluster) being the upper cluster in \( CT \);
- intra-cluster edges are links between two nodes of the same cluster. Any other link is an inter-cluster edge.
- role\((u)\) is the role of node \( u \) (dominator or dominatee).

Nodes periodically broadcast hello messages to discover their neighbors. All leader nodes except RootLeader belong to exactly two clusters. The tree-cluster construction proceeds as follows.

1) The network elects RootLeader.
2) Nodes construct a spanning-tree rooted at the network leader (i.e. RootLeader). A node propagates the minimum id received by neighbors (or its own id if it is lower) by piggybacking this minimum id in hello messages. It also includes in its hello messages the id of its parent in the tree. In this way, any node can maintain the list of its children in \( T \).
3) We define clusters and their leaders based on their position in the spanning tree. A new cluster is created when the distance to the leader upper in the spanning tree is exactly \( R + 1 \) hops, \( R \) representing the radius of each cluster:
   a) RootLeader becomes the first leader;
   b) each node piggybacks in hello the identity of its leader, Leader\((u)\), and its distance in hops via \( T \);
   c) the tree \( T \) is divided so that each node is at most \( R \) hops away from its leader and the leader has the minimum depth in the tree among all its cluster members. To obtain this cluster-tree, a node is self-elected Leader if it is exactly \( \equiv 0 \mod{R} \) hops away from the RootLeader (\( \equiv 0 \mod{R} \) stands for modulo \( R \)).
4) a cluster is finally defined for each leader \( L \) and contains all the nodes with Leader\((u)\) = \( L \).

C. Role Assignment

Each leader needs to know the topology of the whole cluster (we can use tree \( T \) to propagate the topology information and merge it along the tree). Then it computes the local optimal assignment and sends roles to the cluster members. As each leader \( L \) in Cluster (except RootLeader) belongs to another cluster Parent(Cluster), it would normally wait until the leader of Parent(Cluster) gives it a role before computing the roles of nodes in Cluster through MILP. To achieve fully distributed computation, we give a predefined role to each cluster leader with respect to the existence of at least one global r-WCDS. Indeed, if e.g. \( R \) is even, we can safely fix all the leaders to be dominators: a feasible allocation exists to find a path alternating different roles (cf. [6] for more details).

V. PERFORMANCE EVALUATION

We have simulated the proposed protocol in WsNet using the COIN-CBC linear programming library. We randomly place nodes in a simulation area. Nodes use the IEEE 802.11a network interface with the radio range of 10 units and the interference range of 30 units. By default, the network is composed of 50 randomly located nodes with on the average 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 compare the performance of the centralized MILP formulation (OPT), potatoes, the Maximum Independent Set protocol (MIS) and a self-stabilizing Spanning Tree (ST) ([5]). We measure the following performance indices:

- minimum throughput \( T_{min}: \) the minimum throughput guaranteed for each flow extracted from the MILP formulation. We consider the normalized channel capacity;
- average route stretch factor: the average ratio of the length of the shortest route through the r-WCDS and the length of the shortest route in the original graph.

A. Route Stretch Factor

We have first measured the route stretch factor (Fig. 3). For MIS, we discard isolated nodes since the stretch factor would become infinite in this case. OPT results in an average stretch factor around 1.3 for any number of nodes. However, OPT is not scalable—results become difficult to obtain in a reasonable time for more than 40 nodes. This explains why we do not plot results for 50 and 60 nodes under OPT in Fig 3.
B. Capacity

On average, potatoes consumes more bandwidth since it is relayed by more nodes. The stretch factor for MIS is large and a flow approaches the global optimum, our simulations show that its performance is very close to a centralized optimal algorithm. In the future, we plan to explore new strategies to improve the convergence of potatoes. In particular, we can explore redundancy to simplify the MILP formulation.

VI. RELATED WORK

Clusters often make use of the concept of domination: nodes elect a clusterhead and all its neighbors become members of the cluster [7]. In some cases two hops may separate clusterheads so that their interconnection requires gateways. The Weakly Connected Dominating Set (WCDS) is often used for network-wide operations such as clustering or distributing keys in MANET. However, finding a WCDS with a given cardinality is NP-hard [4]. Many researches have focused on computing such WCDS in a distributed way. For example, Alzoubi et al. [8] constructed a Maximum Independent Set, clusterheads being elected based on their depth in a spanning tree. Thus, this algorithm is close to the ST algorithm presented in Section V. In our approach, we build upon the ideas of Chen et al. [9]: they partition the network in zones and each zone executes an algorithm. However, their algorithm is greedy and directly applied to each zone. Moreover, they focused on the original WCDS problem and not on its r-WCDS variant. Besides, we take into account other performance criteria than the cardinality of the WCDS, i.e. network capacity.

ACKNOWLEDGMENTS

This work was partially supported by the European Commission project WIP under contract 27402 and the French Ministry of Research project AIRNET under contract ANR-05-RNRT-012-01.

REFERENCES