

Robust Clustering of Ad-hoc Cognitive Radio Networks under Opportunistic Spectrum Access

Di Li and James Gross

Abstract—The time and space varying nature of channel availability among cognitive radio nodes challenges connectivity and robustness of ad-hoc cognitive radio networks. Clustering of neighboring cognitive radio nodes is a suitable approach to address this challenge, which enables cooperative spectrum sensing, supports a coordinated channel switching and simplifies routing in ad-hoc cognitive radio networks. However, the sudden appearance of primary nodes can lead to the loss of connectivity within a cluster or between clusters. This impact can be mitigated to some extent by the way clusters are formed. In this work we discuss two distributed, lightweight clustering algorithms that strengthen robustness of clusters by improving inter- and intra-cluster connectivity, in which ROSS-DGA is proven to converge within game theoretic framework. Numerical evaluation shows significant improvement achieved by both algorithms on robustness compared with related work.

I. INTRODUCTION

Cognitive radio (CR) is a promising technology to solve the spectrum scarcity problem [1]. In CR systems, primary users access their allocated spectrum band whenever there is information to be transmitted. In contrast, CR users (forming cognitive radio networks, abbreviated as CRN) can only access primary channels after validating the channel is idle. This refers to the process of sensing a particular channel and verifying (with a previously specified probability of error) that it is not used by a primary user currently. This form of spectrum sharing is also referred to as opportunistic spectrum access.

For cognitive radio networks it is well known that clustering leads to a more stable operation of the network. This is due to multiple issues. First of all, by forming clusters for sensing, the sensing reliability can be increased [2]. This prevents mainly interference originating from CR users to primary users which is highly desirable. Also, it prevents CR users from using channels that are occupied by primary users. Secondly, sensing needs to be coordinated within a set of CR users to enable clean results. This typically requires all CR users¹ within some cluster to stop payload transmission on the operating channel and initiate the sensing process. Furthermore, by clustering (and shifting the sensing process) the potential for collisions (when vacating the channel due to primary node appearance) among neighboring clusters is reduced [3]. Finally, routing becomes simplified if clusters are formed in cognitive ad-hoc

networks [4]. Hence, there has been some interest in clustering for cognitive radio networks recently [5], [6], [7].

However, as the activity of primary users is generally not known to CR users in advance, the connectivity between CR nodes in a CRN is not guaranteed. For a pair of communicating CR nodes, whenever a primary user is detected to be using the working channel, CR nodes have to switch to other idle channels. This can lead potentially to a connectivity cut off if there is no such alternative channel. As clustering leads to a dependency between the used working channel and the availability of the working channels for all CR nodes in a cluster, the clustering algorithm has a big impact on stability. Furthermore, the clustering algorithm determines also the connectivity between several clusters which ultimately determines the robustness of the entire network with respect to connectivity. Hence, a desirable feature for a clustering algorithm is connectivity robustness, which means to build clusters such that more common channels are shared by cluster members and more common channels are maintained between neighboring clusters.

This paper contributes by proposing a new clustering algorithm which is especially designed to support robustness. Compared to previous work, the scheme has a much lower complexity while providing a significantly better robustness both in terms of joint channels within a cluster as well as total channels connecting to neighboring clusters. This is accomplished by building more homogeneous clusters with respect to their size and forcing nodes with a high connectivity degree to the border of a cluster (making the cluster therefore more robust regarding connectivity loss to its neighbor). For our scheme we can prove convergence in cluster formation phase and resolve ambiguities with respect to cluster membership in a game-theoretic setting.

The rest of paper is organized as follows. After reviewing related work in Section II, we present our system model in Section III. Then we introduce our clustering scheme and performance evaluation in Section IV and V. Finally, we conclude our work and point out direction future research in Section VI.

II. RELATED WORK

Many clustering algorithms have been proposed in the literature for ad-hoc network [8], [9], [10] and sensor networks [4]. In ad-hoc networks, the major focus of clustering is to preserve connectivity (under static channel conditions) or to improve routing. In the context of sensor networks, the emphasis of clustering has been on longevity and coverage. Hence, none of this work takes into account the channel availability and the issue of robustness in CRNs.

⁰Di Li and James Gross are with the Mobile Network Performance Group, UMIC Research Center, RWTH Aachen University, Germany, {li|gross}@umic.rwth-aachen.de; This work was supported in part by NRW State, Germany, within the B-IT Research School and by the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC), German Research Foundation grant DFG EXC 89.

¹User and node are used interchangeably in this paper

There have been several clustering schemes tailored for CRNs. In [5], the channel available to the largest set of one-hop neighbors is selected as common channel which yields a partition of the CRN into clusters. Their approach minimizes the set of distinct frequency bands (and hence, the set of clusters) used as common channels within the CRN. However, bigger cluster sizes generally lead to less options within one cluster to switch to if the common channel is reclaimed by a primary node. Hence, this scheme does not provide robustness. In [7] the authors consider the balance between the number of idle common channels within cluster and cluster size and propose an algorithm that increases the number of common channels within clusters. However, this work neglects the issue of connectivity between clusters. Furthermore, the scheme leads to a high variance on the size of clusters. Finally, Cogmesh [6] provides a practical MAC protocol for clustering but leaves the set of common channels and sizes of clusters unconsidered.

III. SYSTEM MODEL

Let us consider an area in which a certain spectrum is shared by primary and CR users. The spectrum is divided into $|P|$ non-overlapping channels. There are $|J|$ primary users in the area and $|I|$ CR users. While primary users are assumed to be fixed, CR users can be mobile. Spectrum sharing is implemented by the opportunistic access paradigm. Hence, primary users access their allocated channels whenever they need without any explicit notification to CR users. We assume that primary users have a relatively low variation in activity (periods of activity and inactivity in the range of seconds or minutes). In contrast, CR users are only allowed to access channels after validating these. Validation refers to the process of ensuring that no primary transmission is actually taking place on the respective channel. This is achieved by spectrum sensing by which the CR users validate channels with a certain probability of detection as well as with a certain periodicity, i.e., every T_{sense} time the currently used channels need to be validated again. Denote the spectrum sensing result of each CR user i by $V_i = \{v_1, v_2, \dots, v_{p_i}\}$ with $p_i = |V_i| \leq P$ indicating the total number of available channels for CR user i .

If active, primary and CR users both have a certain transmission range. Any user outside this transmission range can not receive data from the transmitter (i.e., any CR user outside this range can not detect the primary user that transmits). Therefore, different CR users have a different view on the occupancy of the spectrum (apart from the fact that there might be false negatives in the sensing process), i.e., $V_i \neq V_j$. Also, not all CR users can communicate directly with each other. In the following, we do not consider the primary users further and focus on the operation of cognitive radio network (CRN). All $|I|$ CR users form an ad-hoc network in which data is to be transmitted potentially from any node to any other node. Due to the assumed 0/1 nature of connectivity and primary user interference, this CRN can be represented by a connectivity graph $\mathcal{G}(I, \mathcal{E})$, where $\mathcal{E} = \{(i, j, v) | i, j \in I \wedge v \in V_i \wedge v \in V_j\}$ is the set of currently available wireless links between any CR node i and its neighbor j . Availability means here that the

pair of nodes i, j is both located in each other's transmission range and both share a validated channel. Due to relatively low primary user dynamics, time index is omitted here.

To maintain the CRN, CR users form clusters after sensing the spectrum. In order to perform clustering, CR users first need to establish their neighborhood. For CR node i , its neighborhood N_i contains all the CR nodes located within its transmission range (links are assumed to be reciprocal), which have at least one common channel with node i each, i.e. $j \in N_i \Rightarrow V_i \cap V_j \neq \emptyset$. Neighborhood establishment and maintenance are done by a neighborhood discovery protocol which is out of scope of our work. CR nodes exchange their spectrum sensing results V_i over a control channel². Then, the clustering phase is initialized during which any control message is again conveyed by the control channel. In the rest of the paper, the word channel only refers to the spectrum opportunity used for payload communication. A cluster is formally defined by four items: A cluster head, cluster members, common channels of the members with the head and the one channel currently used for payload data transmission. In this paper we only pursue the first three elements and leave the last to future work of payload channel selection. As the CR users are potentially mobile, clustering is performed with some periodicity, but obviously not more often than spectrum sensing.

This paper addresses the problem of how to group CR users into clusters and maintain the connectivity of the CRN. In particular, the clustering should make the CRN as robust to primary user interference as possible. Connectivity within the CRN is guaranteed as long as there is at least one channel available within each cluster but also between neighboring clusters. As it is difficult to predict the activity of primary users, the goal of clustering is to establish as many common channels within each cluster as well as between neighboring clusters. This needs to be achieved by a distributed algorithm.

In the following, we refer to the common channels within a cluster C by the term *inner common channels* (ICC) and denote them by set K_C . Also, we refer to the common channels between neighboring clusters by the term *outward common channels* (OCC). We define the set of OCCs of cluster C to be the set of available common channels between any member of C and any other CR user of a neighboring cluster:

$$R_C = \bigcup_{j \in C, k \in N_j, k \notin C} (V_j \cap V_k)$$

The goal of the proposed clustering schemes is to obtain a set of clusters with big $|K_C|, |R_C|$.

IV. DISTRIBUTED COORDINATION FRAMEWORK: CLUSTERING ALGORITHM

In this section, we present the new clustering scheme named ROSS (RObust Spectrum Sharing). It is based on the local sensing results V_i of all CR users i and utilizes local similarity of the available channels to form clusters. ROSS consists of two

²for example, ISM band or other reserved channels which are exclusively used for transmitting control messages

TABLE I

| Symbol | Description |
|-----------|--|
| I, J | set of CR and primary nodes in the scenario |
| P | set of non-overlapping channels in the scenario |
| N_i | node i ' neighborhood |
| V_i | set of available channels on CR node i |
| p_i | number of available channels on CR node i |
| $CH(C)$ | cluster head of cluster C |
| C_i | a cluster composed by CR nodes and with CR node i as CH |
| S_i | set of clusters, each of which includes debatable node i after phase I |
| K_{C_i} | set of common channels within cluster C_i |
| R_{C_i} | set of outward common channels of cluster C_i |

phases: *cluster formation* and *membership clarification*. We will describe both phases individually.

A. Phase I - Cluster Formation

After initial sensing and neighborhood discovery, the CR nodes are ready for cluster formation. We define two values to characterize the channel availability in neighborhood.

Spectrum Connectivity Degree: $D_i = \sum_{j \in N_i} |V_i \cap V_j|$, which denotes the sum of the pairwise common channels of node i , and is an indicator of node i 's adhesive property to the CRN.

Local Connectivity Degree: $G_i = |\bigcap_{j \in N_i} V_j|$, which is the number of common channels in N_i . G_i represents the ability of a neighborhood to form a robust cluster. Figure 1 illustrates an example CRN where the corresponding *Spectrum/Local Connectivity Degree* are specified for each node. The algorithm of

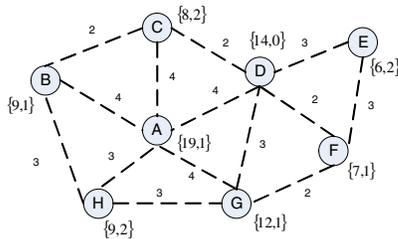


Fig. 1. Connectivity graph and the connectivity vector $\{D_i, G_i\}$ on each node. The available channels sensed by each CR node are: $V_A = \{1, 2, 3, 4, 5, 6, 10\}$, $V_B = \{1, 2, 3, 5, 7, 10\}$, $V_C = \{1, 3, 4, 10\}$, $V_D = \{1, 2, 3, 5, 10\}$, $V_E = \{2, 3, 5, 7\}$, $V_F = \{2, 4, 5, 6, 7, 10\}$, $V_G = \{1, 2, 3, 4, 8\}$, $V_H = \{1, 2, 5, 8\}$. Dashed lines indicates two end nodes are within transmission range of each other. Each edge is labeled by the number of common channels between the two ends.

phase I can be sketched like this: cluster heads are determined first, then clusters are formed second.

1) *Determining Cluster Heads*: All nodes check whether they can become cluster head or cluster members periodically. For a CR node i , if $D_i < D_k, \forall k \in N_i \setminus CHs$ (CHs donate the cluster heads existing in N_i), then i is cluster head. If there is another CR node j in its neighborhood with $D_j = D_i$, and $D_j \leq D_k, \forall k \in N_j \setminus CHs$, then the node with better *Local Connective Degree* becomes cluster head, and the other one becomes member of it. If $G_i = G_j$ as well, then the one with smaller node ID takes precedence and becomes cluster head.

2) *Initial Cluster Formation*: After becoming cluster head, CR node i forms the initial cluster $C_i = (N_i \setminus CHs) \cup i$.

Figure 1 depicts an example how CR nodes decide cluster heads. Node B and H have same local robustness, $d_B = d_H$, but as $g_H = 2 > g_B = 1$, node H becomes cluster head. In Figure 1, the cluster C_H is $\{H, B, A, G\}$. It is possible that the nodes in C_i have no common channel. This can be handled in the following way. As a smaller cluster size can increase the number of common channels within the cluster, some nodes are eliminated until there is at least one common channel. The elimination of nodes is performed according to an ascending list of nodes sorted by their number of common channels with the cluster head. If there are nodes having the same number of common channels with cluster head, the node whose elimination brings in more common channels will be chosen and excluded. If this criterion meet a tie, then the tie will be broken by deleting the node with smaller ID. At the end of this procedure every cluster has at least one common channel. For the nodes eliminated, they can possibly become cluster heads or get included by other clusters later on.

After becoming a member of a cluster, the *Spectrum Connectivity Degree* on the CR user is changed to a big positive value M (can be regarded as a positive infinite value), which is bigger than all the possible connectivity degrees calculated in the CR network. Then this CR user broadcasts this new *Spectrum Connectivity Degree* to all its neighbors. If a CR node i is associated to multiple clusters, D_i is still set to M .

The algorithm of phase I makes sure that every CR node either becomes cluster head or a member of at least one cluster, as formulated by the following lemma.

Lemma 1. *Every node in CRN will be included into at least one cluster in phase I in finite steps.*

To see this, assume there are some nodes not assigned to any cluster and node α is one of them. As node α is not contained in any cluster, there must be at least one node $\beta \in N_\alpha$, with $D_\beta < D_\alpha$. Thus, node β has at least one neighboring node γ with $D_\gamma < D_\beta$, and this series of nodes with monotonically decreasing D_i might continue but finally ceases because the total number of nodes is limited. Now we find the last node ω in this series, because ω is the end node and does not have neighboring nodes with smaller connectivity degree D , so ω will become a cluster head and embrace all its one-hop neighbors, including the node before it in the node series (here we assume that every new formed cluster has common channels). After that, the node recruited into cluster will set its connectivity degree D to M , which enables the node further down in the list to become a cluster head. In this way, all the nodes in the series are included in at least one cluster in an inverse sequence. This clearly contradicts the initial assumption and proves the claim stated above. The proof implicitly shows that, within $|I|$ steps, all nodes will become a part of certain clusters and so phase I converges.

Figure 2 shows the clusters formed after phase I based on the example of Figure 1. Notice that there are several nodes, like

A, B, D , are included by more than one cluster. We refer to these nodes as *debatable nodes* because their cluster heads are not clear, and the clusters which include debatable node i are named as *claiming clusters*, the set of which is represented by S_i . The members of a cluster and the set of available channels in it is known by debatable nodes (which lie in that cluster). The second phase of ROSS clarifies cluster membership of the debatable nodes. Especially, it fosters the connectivity between the clusters. It can do so, as the nodes with larger connectivity degree are not cluster heads but members.

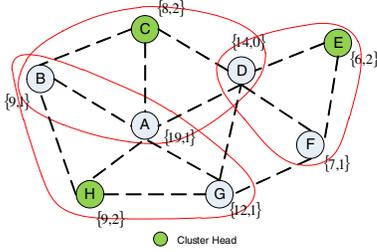


Fig. 2. Clusters formation after the first phase of ROSS. Some nodes remain debatable nodes after the first phase.

B. Phase II - Membership Clarification

1) *Problem Description*: In this phase, debatable nodes need to be uniquely associated to one cluster and removed from the other claiming clusters finally. In particular, note that the process of eliminating debatable nodes from a cluster can possibly increase the set K_C of ICCs of cluster C (at the cost of potentially decreasing R_C , the set of OCCs).

2) *Distributed Greedy Algorithm (DGA)*: Each debatable node i clarifies its membership with the purpose of maximizing the total number of ICC in S_i according to Algorithm 1:

Debatable node i can either execute this algorithm periodically or triggered by the change of membership of $C \in S_i$. Notice that, as to debatable node $i \in S_i$, cluster $C \in S_i$ could have several debatable nodes whose choices may change cluster C 's membership and further trigger node i to alter its previous decision. Thus, the question arises whether the process converges if all the debatable nodes implement ROSS-DGA, and if it converges, how good such a distributed scheme performs. Below we show that this problem can be converted into an equivalent congestion game, and a stable state is reached by the greedy procedure within a finite number of updates.

3) *Convergence of DGA*: To formulate the problem of membership clarification for the debatable nodes in a game-theoretic framework, we use a different perspective to describe this process. In the new perspective, the debatable nodes are regarded as isolated and don't belong to any cluster. Thus, the clusters they used to belong to become their neighboring clusters. So for each debatable node, the original problem of deciding which clusters to leave becomes in turn the problem which cluster to join. In this new problem, debatable node i (note now $i \notin S_i$) greedily updates its choice that channel availability of clusters in which cluster $C \in S_i$ to join so that the decrement

Algorithm 1 update belongings (for debatable node i)

- 1: **Input**: S_i , membership of every cluster $C \in S_i$
- 2: **Begin procedure**:
- 3: Find out cluster $C \in S_i, i \in C$, so that
 $\arg \max_{C \in S_i} \sum_{C' \in S_i, C' \neq C} |K_{C'}|$, with $C' = C' \setminus i$
- 4: **if** There exists only one C achieving above condition **then**
- 5: $C' = C' \setminus i$. Go to Line 14
- 6: **else**
- 7: The set of C satisfying the condition are donated as \mathcal{C}
 Find $C \in \mathcal{C}$, so that $\arg \max_{C \in \mathcal{C}} (V_{CH(C)} \cap V_i)$
- 8: **if** There exists only one C achieving above condition **then**
- 9: $C' = C' \setminus i$. Go to Line 14
- 10: **else**
- 11: The set of C satisfying the condition are donated as \mathcal{C}' , i will belong to $C \in \mathcal{C}'$, which has smallest cluster size, with cluster head's ID to break tie.
- 12: **end if**
- 13: **end if**
- 14: **if** $\sum_{C \in S_i} |K_C|$ calculated is bigger than its current value **then**
- 15: adapt the new strategy and notify all $C \in S_i$
- 16: **else**
- 17: remain the current strategy
- 18: **end if**
- 19: **End procedure**

of ICC in S_i is the smallest. The decrement of number of ICC in S_i is $\sum_{C \in S_i} \Delta |K_C| = \sum_{C \in S_i} (|K_C| - |K_{C \setminus i}|)$. This new problem can be formulated as a *player-specific singleton congestion game*, and can be represented by a tuple $\Gamma = (\mathcal{N}, \mathcal{R}, (\sum_{i \in \mathcal{N}} \Delta |K_C^i|)$. Here,

- $\mathcal{N} = \{1, \dots, n\}$, the set of players (debatable nodes).
- $\mathcal{R} = \{1, \dots, m\}$, the set of resources which player can choose, which are all the clusters in our model.
- $\sum_i \subseteq 2^{[S_i]}$, $i \in \mathcal{N}$, is the strategy space of player i . S_i is the set of claiming clusters of node i . Note that only one resource is allocated for i when it makes decision (thus the game is a singleton game).
- We use function $\Delta |K_C^i|$ to represent the decrement of ICCs in cluster C caused by debatable node i ' joining in it. For one cluster $C \in S_i$, the decrement of ICCs caused by enrollment of debatable nodes is $\sum_{i: C \in S_i} \Delta |K_C^i|$. Note that this function is non-decreasing.
- The Rosenthal's potential function [11] of this congestion game is given by:

$$\phi(S) = \sum_{C \in \mathcal{R}} \sum_{i: C \in S_i} \Delta |K_C^i|$$

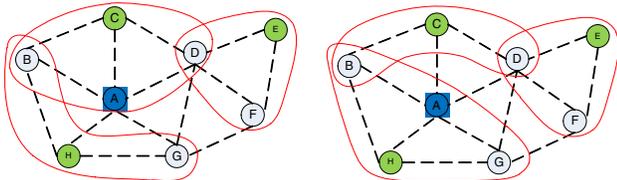
All the players in this game greedily update their strategy to minimize the potential function (congestion), this process is exactly the same with the network behavior under *Distributed Greedy Algorithm*.

Singleton congestion game is a special type of matroid game [12], [13]. It is known that player-specific matroid congestion game admit pure equilibrium, and the number of steps towards *Nash Equilibrium* is upper-bounded³ by $n^2 \cdot m$. In our context, n is the number of debatable nodes, m is number of clusters in CRN, so the total time complexity to achieve the *Nash Equilibrium* using greedy approach is $O(n^2m)$. This is upper-bounded (in the worst case) by $O(|I|^3)$. Based on above model and analysis, phase II can converge if debatable nodes utilize the DGA.

4) *Distributed Fast Algorithm (DFA)*: The complexity of DGA is quite large recalling that the formation of clusters takes at most $|I|$. Here we propose a faster algorithm DFA which is especially suitable for CRN where channel availability might change dynamically and re-clustering is possible. In DFA, each debatable node executes only one iteration of Algorithm 1 (by setting 'the current value' in Line 14 to zero). Every cluster includes all its debatable nodes, thus the membership is static and debatable nodes can make decisions simultaneously without considering the change of membership of its claiming clusters. For example, for node A in Figure 2, the membership of cluster $C_C, C_H \in S_A$ are $\{A, B, C, D\}$ and $\{A, B, H, G\}$ respectively.

The two possible strategies of node A 's clarification is illustrated in Figure 3. In Figure 3(a), node A staying in C_C and leaving C_H brings 2 more ICC in S_A , as it is more than that brought by another strategy showed in 3(b), A 's membership is clear. After the decisions made similarly by the other debatable nodes B and D , the final clusters formed are shown in Figure 4.

Using DFA in phase II, the time complexity is decreased drastically to 1. Thus, the total complexity of ROSS-DFA is $|I|$, while, ROSS-DGA's complexity is $|I|^3$ in the worst case.



(a) Node A stays in cluster C_C , (b) Node A stays in cluster C_H , quits C_H , $\Delta|R_{C_C}| + \Delta|R_{C_H}| = 2$ quits C_C , $\Delta|R_{C_C}| + \Delta|R_{C_H}| = 1$

Fig. 3. Membership clarification: Node A 's possible strategies

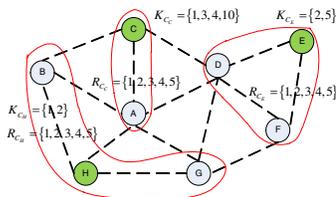


Fig. 4. Final cluster formation.

³Here we present this with modifying the original conclusion in [13] according to our model.

V. PERFORMANCE EVALUATION

In this section, we compare the performance of ROSS-DGA, ROSS-DFA and one other approach by means of simulations. To the best of our knowledge, SOC [7] is the only work emphasizing on the robustness of clustering structure from all previous work on clustering in CRN. The authors of [7] compared SOC with other schemes based on the average number of common channels within each cluster, on which SOC outperforms other schemes by 50%-100%. This is because the schemes except for SOC are designed either for ad hoc network without consideration of channel availability [10], or for CRN but just considering basic connection among CR nodes [5]. Hence, we only compare our scheme with SOC to demonstrate the advantage of DGA and DFA on robustness of clustering structure and cluster size.

The simulation is done by implementing random graph structures in C++. Coinciding with the system model in Section III, primary and CR users are dropped randomly (with uniform distribution) within some area of size A^2 , where we set the transmission ranges of primary and CR users to $A/5$ and $A/10$ respectively. There are $P = 10$ available channels. Each primary user chooses one channel randomly, afterwards CR users are assumed to sense the existence of primary users perfectly if located within transmission range. All primary and CR users are assumed to be static during the process of clustering. Performance results are averaged over 50 randomly generated topologies with equal parameters.

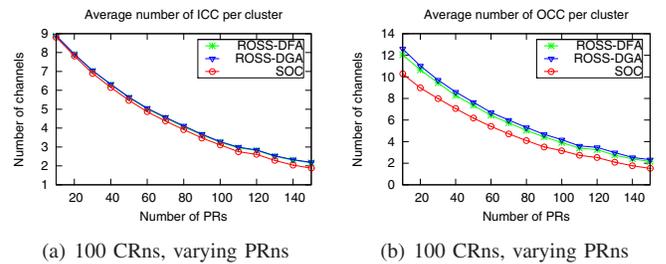


Fig. 5. Connectivity robustness with varying density of primary users.

We present performance results for two different cases. First, we fix the number of CR nodes but increase the number of primary users (from 10 to 150). Second, we keep the number of primary users fixed and vary the number of CR nodes in the area (from 100 to 500). As performance metrics, we consider the average ICC and OCC as well as the empirical distribution function of the ICC and OCC values of the formed clusters by ROSS-DGA/DFA and SOC. The confidence intervals are below 5% of the absolute results for a confidence level of 95% for each point and not shown in the figures.

In the first case, there are 100 CR nodes while primary users are increased from 10 to 150. More primary users lead to fewer idle channels available in the whole network, and thus cause a challenge to the formation of clusters. In Figure 5(a), the average number of inner common channels achieved by the three approaches decreases with increasing the number of

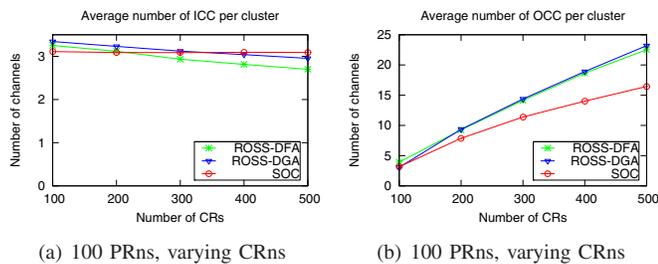


Fig. 6. Connectivity robustness of ICCs and OCCs with varying density of CR nodes.

primary users. ROSS-DGA/DFA outperform SOC by at most 15% in this case. In Figure 5(b), we present the average number of outward common channels. Here, ROSS-DFA outperforms SOC by 20%-40% due to putting CR nodes with bigger connectivity degree at the border of clusters to strengthen the connection among them. ROSS-DGA performs slightly better than ROSS-DFA on the whole range due to its larger number of iterations.

For the second scenario, we vary the number of CR nodes from 100 to 500 while keeping the number of primary users fixed at 100. Hence, we investigate the behavior of the three schemes in sparse and dense situations. Figure 6(a) shows the average number of ICCs. We observe that ROSS-DGA/DFA achieves more ICC in sparse networks while slightly less in dense networks. We attribute this to two reasons, firstly, SOC pursues the maximal product of cluster size and number of ICC, so the product value is assured in many cases by decreasing cluster size to get more ICCs. Actually, there is a large number of clusters with only one member. Secondly, ROSS-DGA/DFA builds clusters on the basis of one-hop neighborhood, and dense network means there are more CR nodes within the neighborhood, thus agree on less ICCs. Figure 6(b) demonstrates an increasing advantage of ROSS-DGA/DFA against SOC on number of outward common channels, because more nodes with bigger connectivity degree are put as border nodes. The distribution of cluster sizes are presented in Figure 7(a) and 7(b) with variation of density. We observe that clusters formed by ROSS-DGA/DFA have similar size. Note in particular that the number of one-node clusters generated by SOC is much bigger than that produced by ROSS-DGA/DFA in both cases. This becomes especially a problem as the density increases.

Compared with ROSS-DGA, we can find from the simulation that ROSS-DFA has much less complexity by sacrificing a little performance, and both ROSS-DGA and ROSS-DFA are less complex than SOC which has a complexity of the order of $|I|^4$. This was confirmed by the run times of the simulations, which were significantly longer when simulating SOC.

VI. CONCLUSIONS AND FUTURE WORK

We presented two clustering approaches ROSS-DGA and ROSS-DFA that target at robust clusters in CRN. Both algorithms enable clusters to have abundant common channels within themselves and with other clusters. This reduces the

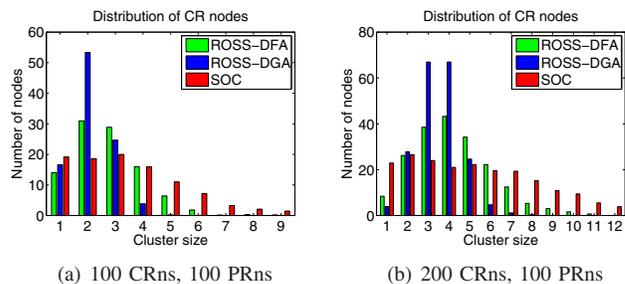


Fig. 7. Distribution of cluster sizes for two fixed scenarios.

risk that all common channels become unavailable because of primary users' appearance. Compared to previous work, clusters generated by ROSS have less clusters composed of only one node, which is advantageous also for cooperative sensing. ROSS has a low complexity and especially ROSS-DFA has a complexity that is linear with the number of nodes of a CRN. Simulation shows ROSS achieves significant improvement on the aforementioned aspects compared with other state-of-art clustering scheme. Future work consists of channel assignment to decrease the co-channel interference within and among CRN clusters.

REFERENCES

- [1] I. Mitola, J. and J. Maguire, G.Q., "Cognitive radio: making software radios more personal," *Personal Communications, IEEE*, vol. 6, pp. 13–18, aug. 1999.
- [2] C. Sun, W. Zhang, and K. B. Letaief, "Cluster-based cooperative spectrum sensing in cognitive radio systems," in *Proc. IEEE ICC '07*, pp. 2511–2515, 2007.
- [3] D. Willkomm, M. Bohge, D. Hollós, J. Gross, and A. Wolisz, "Double hopping: A new approach for dynamic frequency hopping in cognitive radio networks," in *Proc. IEEE PIMRC '08*, Sept. 2008.
- [4] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, no. 14-15, pp. 2826–2841, 2007.
- [5] J. Zhao, H. Zheng, and G.-H. Yang, "Spectrum sharing through distributed coordination in dynamic spectrum access networks," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 9, pp. 1061–1075, 2007.
- [6] T. Chen, H. Zhang, G. Maggio, and I. Chlamtac, "Cogmesh: A cluster-based cognitive radio network," in *Proc. IEEE DySPAN '07*, pp. 168–178, apr. 2007.
- [7] L. Lazos, S. Liu, and M. Krunz, "Spectrum opportunity-based control channel assignment in cognitive radio networks," in *Proc. IEEE SECON '09*, pp. 1–9, jun. 2009.
- [8] V. Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in *Proc. IEEE infocom '03*, pp. 459–469, 2003.
- [9] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1265–1275, 1997.
- [10] S. Basagni, "Distributed clustering for ad hoc networks," in *Proc. I-SPAN '99*, pp. 310–315, 1999.
- [11] R. W. Rosenthal, "A class of games possessing pure-strategy nash equilibria," *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973.
- [12] I. Milchtaich, "Congestion games with player-specific payoff functions," *Games and Economic Behavior*, vol. 13, no. 1, pp. 111–124, 1996.
- [13] H. Ackermann, H. Röglin, and B. Vöcking, "Pure nash equilibria in player-specific and weighted congestion games," in *In Proc. WINE '06*, pp. 50–61, 2006.