Understanding the Power of Distributed Coordination for Dynamic Spectrum Management

Lili Cao and Haitao Zheng
Department of Computer Science
University of California, Santa Barbara, CA 93106 U.S.A
Email: {lilicao,htzheng}@cs.ucsb.edu

Abstract

This paper investigates a distributed and adaptive approach to manage spectrum usage in dynamic spectrum access networks. While previous works focus on centralized provisioning, we propose a distributed low complexity approach where nodes self-organize into coordination groups and adapt their spectrum assignment to approximate the global optimal assignment. We design a distributed coordination protocol to regulate the coordination format and achieve fast system convergence. The proposed approach achieves similar performance in spectrum assignment compared to that of the conventional centralized approaches, but significantly reduces the number of computations and message exchanges required to adapt to topology changes. As a case study, we investigate the detailed coordination strategy to improve proportional fairness in spectrum assignment, and derive a theoretical lower bound on the minimum spectrum/througput each node can get from coordination. Such bound can be utilized to guide the coordination procedures. We also show that the difference between the proposed approach and the global optimal approach is upper-bounded. Experimental results demonstrate that the proposed approach provides similar performance as the topology-based optimization but with more than 50% of reduction in complexity.

I. Introduction

Wireless spectrum is a finite and scarce resource. Current (and historical) spectrum policies assign static spectrum bands to wireless technologies to prevent interference among them. Over time, such static assignments have led to over-allocation and under-utilization of licensed bands, creating an artificial spectrum scarcity problem [33], [50].

One promising solution to overcome the artificial scarcity is dynamic spectrum access. Future wireless devices will use cognitive radios [2], [20], [27], [32], [35], [45], the next generation software-reconfigurable frequency-agile radios, and dynamically access locally available spectrum. Further, many have advocated for Open Spectrum Systems [5], [10], [28], [41], [39] where cognitive radios become secondary devices and opportunistically access licensed spectrum, all without disrupting operations of any existing license owners, *i.e.* primary users.

The key challenge in dynamic spectrum access networks is how to maintain efficient spectrum sharing among (secondary) users. We hereby refer to secondary users as nodes. While maximizing spectrum utilization is the primary goal of dynamic spectrum networks, we also need a good sharing mechanisms to provide fairness across nodes. A node seizing spectrum without coordinating with others can cause harmful interference to neighbors and hence reduce spectrum utilization.

Design of dynamic spectrum access networks can be divided into two types: power based spectrum sharing where nodes use the same spectrum band but transmit at different powers to minimize interference [12], [21], [22]; and channel sharing where nodes transmit at orthogonal channels to avoid interference. The mechanisms for channel sharing can be further divided into two categories.

1) *System-utility driven* – Nodes share spectrum to maximize a predefined system utility [6], [7], [9], [24], [29], [40], [42], [49], [53]. This type of sharing also refers to collaborative sharing where nodes have signed agreements or are deployed by the same service providers to maximize system wide performance regardless of individual benefits.
2) *Self-gain driven* – Nodes share spectrum to maximize self-benefit. Existing works have applied game theory to design and analyze spectrum usage strategies [38], [37]. This type of sharing refers to non-collaborative sharing where nodes are rational (and competitive), and are only interested in maximizing self-benefits.

In this paper, we focus on the *system-utility driven, collaborative* channel sharing. A motivating scenario for our work is spectrum sharing among WiMAX/WiFi access points[1], shown in Figure 1. These access points are deployed across geographic locations to support their subscribers, such as Googlenet [1] in San Francisco. Access points must share spectrum to avoid interference and maintain reliable communication links, and to maximize a system utility function. Access points opportunistically exploit unused spectrum by nearby primary users, *i.e.* analog TV broadcast/receivers, and hence their available spectrum changes over time and location.

The problem of spectrum management can be reduced into a variant of the graph coloring problem [7], [40], which is NP-hard. Given a small fixed network topology, existing approaches have proposed good heuristics based on *centralized systems* to obtain conflict free spectrum assignments that closely approximate the global optimum. In this case, a server collects information of node demand and assigns spectrum to a limited number of nodes to maximize system utility. However, network topology, node demand and available spectrum change over time. Hence the system needs to completely recompute spectrum assignments for all nodes after each change, resulting in high computational and communication overhead. This costly operation needs to be repeated frequently to maintain utilization and fairness.

In this paper, we consider a distributed approach to spectrum allocation that can potentially address the coexistence of a large number of nodes, and quickly adapt to network dynamics. We propose a distributed coordination based approach where nodes affected by network dynamics self-organize into coordination groups and adapt their spectrum usage to approximate the new optimal conflict free assignment. In particular, nodes start from the last spectrum assignment and perform local improvements to gradually improve system utility. To avoid conflict among concurrent local modifications, we design regulations to restrict coordination formats and achieve fast system convergence.

The key contributions of this paper are three-fold:

Distributed Coordination Framework. We propose a distributed framework where nodes coordinate locally to refine spectrum assignment. The proposed system enables concurrent coordination efforts to reduce coordination delay, and applies regulations to prevent conflicts and cascading effects due to concurrent coordinations. We propose two simple coordination formats: *one-to-one fairness coordination* and *feed poverty coordination* to maximize fairness based system utility,

Analytical Bounds. We evaluate the proposed approach analytically in terms of the spectrum assignment performance, the distance to the global optimum and the coordination overhead/complexity. First, we derive a theoretical lower bound on the amount of spectrum a particular node can get from coordination, referred to as its *Poverty Line*. This lower bound represents the level of fairness enforced by the proposed approach, and can be used to guide the coordination procedures. Second, we derive a upper bound on the performance distance between the proposed approach and the global optimal solution, referred to as *price of anarchy*. Finally, we derive a upper bound on the coordination complexity and overhead.

Simulation of Efficiency and Complexity. We conduct extensive simulations to quantify the performance of distributed

[1]We assume these access points are equipped with Cognitive Radios, and can utilize multiple non-consecutive channels concurrently to provide high speed wireless connections.

coordination. Results indicate the proposed approach performs similarly to the centralized solution[54], [40] but at significantly reduced complexity. We also validate the *poverty line*. Moreover, we examine the impact of network topology to the efficiency and complexity of our approach, using both randomly generated topologies and measured AP deployment traces.

The rest of the paper is organized as follows. We begin in Section II by defining the spectrum allocation problem and the centralized solutions. Next, we present the distributed coordination framework in Section III. As a case study, we propose specific coordination formats to improve system fairness in Section IV. In Section V we evaluate the proposed system theoretically. In Section VI, we conduct experiments to evaluate the performance of coordination and validate the theoretical lower bound. Finally, we summarize related work in Section VIII, discuss implications and future directions and conclude in Section IX.

II. The Problem of Spectrum Allocation

The two key goals of allocating spectrum are spectrum utilization and fairness. Combinations of these two goals form specific utility functions that can be customized for different types of network applications. As background, we describe in this section the general problem of spectrum allocation. We start with the theoretical model used to represent the general allocation problem, and two utility functions that maximize spectrum utilization and fairness. We describe how to reduce the optimal allocation problem to a variant of a graph multi-coloring problem and describe existing centralized solutions that optimize the spectrum allocation for a given topology.

A. Problem Model and Utility Functions

As shown in Figure 1, we assume a network of N nodes indexed from 0 to $N-1$ competing for M spectrum channels indexed 0 to $M-1$. We consider the case where the collection of available spectrum ranges forms a spectrum pool, divided into non-overlapping orthogonal channels. Nodes select communication channels to avoid interfering with legacy (*i.e.* primary) users. Equipped with cognitive radios, nodes can utilize non-consecutive channels concurrently for high throughput transmissions. The channel availability and throughput for each node can be calculated based on the location and channel usage of nearby primaries. We assume that each node transmits using a predefined combination of radio parameters (*e.g.* power and modulation), and focus on achieve conflict-free spectrum allocation by assigning conflicting nodes with orthogonal channels[2]. The spectrum access problem becomes a channel allocation problem, *i.e.* to obtain a conflict free channel assignment for each node that maximizes system utility. The key components of our model are:

Channel availability $\mathbb{L}$: We define $\mathbb{L} = \{l_{m,n} | l_{m,n} \in \{0,1\}\}_{M\times N}$ as a M by N binary matrix representing the channel availability: $l_{m,n} = 1$ if and only if channel m is available at node n. In general, $l_{m,n} = 0$ when channel m is occupied by a primary user who conflicts with node n, so that the transmissions of n on this channel will interfere with the primary's activity if they use channel m concurrently. Let $L(n) = \{0 \leq m \leq M-1 | l_{m,n} = 1\}$ be the set of channels available at n.

Interference constraint $\mathbb{C}$: Let $\mathbb{C} = \{c_{n,k} | c_{n,k} \in \{0,1\}\}_{N\times N}$, a N by N matrix, represents the interference constraints among nodes. If $c_{n,k} = 1$, nodes n and k would interfere with each other if they use the same channel. The interference constraint depends on the signal strength of transmissions and the distance between nodes. A simple model of interference constraint is the binary geometry metric, *i.e.* two transmissions conflict if they are within π distance from each other. In Section VII we discuss extensions to more complex interference models.

[2] While the proposed approach can be integrated with other parameter-tuning techniques such as power control, we will defer this investigation to a later study.

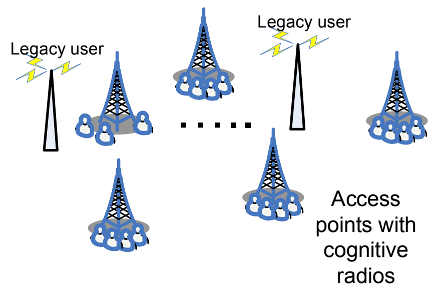


Fig. 1. An illustrative scenario of distributed spectrum sharing. Access points identify locally available channels that will not disrupt the operations of legacy spectrum owners such as TV broadcasts. They also share these available spectrum with peers, and access spectrum to connect their subscribers.

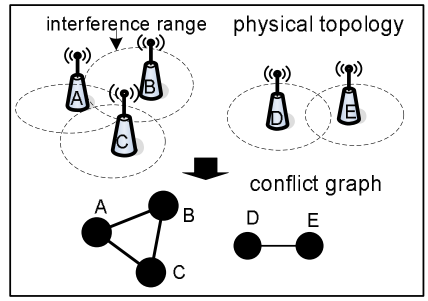


Fig. 2. A sample conflict graph.

Node dependent channel throughput $\mathbb{B}$: Let $\mathbb{B} = \{b_{m,n} > 0\}_{M\times N}$ describe the reward that a node gets by successfully acquiring a spectrum band, i.e. $b_{m,n}$ represents the maximum bandwidth/throughput that node n can acquire through using spectrum band m (assuming no interference from other neighbors). Let $b_{m,n} = 0$ if $l_{m,n} = 0$. So that B represents the bandwidth weighted node available spectrum.

Conflict free assignment $\mathbb{A}$: Let $\mathbb{A} = \{a_{m,n}|a_{m,n} \in \{0,1\}\}_{M\times N}$ where $a_{m,n} = 1$ denote that spectrum band m is assigned to node n, otherwise 0. A satisfies all the constraints defined by C, that is,

$$a_{m,n} + a_{m,k} \leq 1,\ if\ c_{n,k} = 1, \forall\, n, k < N, m < M.$$

Let $\Lambda_{N,M}$ denote the set of conflict free spectrum assignments for a given set of N nodes and M spectrum bands.

Node throughput $\mathbb{R}$: Let $R_{\mathbb{A}}(n)$ represent the throughput that node n gets under assignment A, *i.e.* $R_{\mathbb{A}}(n) = \sum_{m=0}^{M-1} a_{m,n} \cdot b_{m,n}$.

Given this model, the goal of spectrum allocation is to maximize network utilization, defined by U. We can define the spectrum assignment problem by the following optimization function:

$$\mathbb{A}^* = \max_{\mathbb{A}\in\Lambda_{N,M}} arg\max U(\mathbb{A}),$$

We can obtain utility functions for specific application types using sophisticated subjective surveys. An alternative is to design utility functions based on traffic patterns and fairness inside the network. In this paper, we consider and address fairness based system utility. Consistent with prior work[36], [23], [48], we address fairness for single-hop flows since they are the simplest format in wireless transmissions. We postpone the discussion of routing related utility functions to a future paper. Similar to [36], we define fairness in terms of maximizing total logarithmic node throughput, refereed to as proportional fairness. The utility can be expressed as

$$U(\mathbb{A}) = \sum_{n=0}^{N-1} \log R_{\mathbb{A}}(n) = \sum_{n=0}^{N-1} \log \sum_{m=0}^{M-1} a_{m,n} \cdot b_{m,n}. \tag{1}$$

As a reference, another utility function is the total spectrum utilization in terms of total node throughput, $U(\mathbb{A}) = \sum_{n=0}^{N-1} R_{\mathbb{A}}(n)$. Maximizing utilization does not consider fairness and can lead to node starvation.

B. Color-Sensitive Graph Coloring

Conflict graphs have been widely used to characterize interference constraints [7], [26], [29], [40], [44], [42]. We start from an illustrative example in Figure 2 where nodes A to E are for example access points[3] that provide network access for their subscribers. Since A, B and C are located closely to each other, their subscribers will receive signals from all three APs. Signals from non-associated APs are interference that could disrupt communications. To minimize interference, A, B and C should not use the same spectrum concurrently. Those interference constraints can be modeled by a conflict graph $G = (V, L)$. Each vertex $v \in V$ represents a node and an edge $l(u, v) \in L$ exists between any two nodes if their communications interfere. Any two connected vertices should not use the same spectrum concurrently. The example of Figure 2 reduces to one triangle (A,B,C) and one link (D,E). In this paper, we assume that nodes perform conflict discovery to construct local conflict graph.

The corresponding spectrum management problem is to color each vertex using a number of colors from its color list, and find the color assignment that maximizes system utility. The coloring is constrained by that if an edge exists between any two distinct vertices, they can't be colored with the same color. Most importantly, the objective of coloring is to maximize system utility. This is different from traditional graph color solutions that assign one color per vertex. Notice that the solution to this graph coloring problem is to maximize system utility for a given graph, *i.e.* a given topology and channel availability. This characterizes the optimal solution for a static environment.

C. Existing Spectrum Management Solutions

The optimal coloring problem is known to be NP-hard [15]. Existing efforts lead to efficient algorithms that closely approximate the optimum spectrum allocation for a given network topology. There are multiple complementary ways of addressing the problem of spectrum management, each applicable to different scenarios.

The work in [24] proposed a demand responsive pricing framework to use iterative bidding to maximize the expected revenue. They use exhaustive search to derive the optimal channel allocation for a small number of nodes. The work in [7], [6], [42] proposed the use of regional server model, and developed several heuristics based centralized approximations for a limited number of nodes. Results in [40], [54] show that the heuristic based algorithms perform similarly to the global optimum (derived off-line for simple topologies), and the centralized and distributed algorithms perform similarly. Further, the work in [47] proposed a hybrid pricing model - use simple auctions during peak period with a reserved price while applying a uniform price to all buyers during off-peak. Finally, another approach uses power based allocations and applies price-driven power control to minimize interference [21]. In this case, all buyers use the same spectrum band.

III. Distributed Coordination Framework

The approach described in Section II globally optimizes spectrum allocation for a given topology. In dynamic spectrum networks, mobility, traffic dynamics and primary user dynamics lead to constant changes in network topology and spectrum availability. Using existing approaches, we can reapply the spectrum allocation algorithm after each change in the conflict graph. This approach assumes no prior allocation information, and incurs high computation and communication overheads. To reduce these overheads, we propose the use of an adaptive and robust distributed algorithm that takes prior allocation into account in new spectrum assignments.

An efficient dynamic allocation algorithm can run every time network condition changes. Therefore, an adaptive algorithm needs to only compensate for small changes affecting a local network region. The algorithm starts from a non-optimal spectrum

[3] A node is a component of wireless networks, *e.g.* cellular/WiMAX base stations, mesh routers.

allocation, which can be constructed from the allocation prior to the topology change. Consider a conflict graph with N nodes (indexed from 0 to $N-1$) and M channels (indexed from 0 to $M-1$), where the optimized assignment is $\mathbb{A}_{M \times N}$. When a new node (indexed as N) joins the network, the assignment after introduction of the node becomes $\mathbb{A}'_{M \times (N+1)}$ where

$$\mathbb{A}'_{m,n} = \begin{cases} \mathbb{A}_{m,n} & : \quad 0 \le n \le N-1,\ 0 \le m < M \\ 0 & : \quad n = N,\ 0 \le m < M \end{cases}$$

Or if a primary user i enters the network and wants to use channel m_0, the nodes within impact of primary i (denoted by $Nbr(i)$) need to stop using channel m_0 within a given time. Hence, the assignment becomes $A'_{M \times N}$ such that

$$\mathbb{A}'_{m,n} = \begin{cases} 0 & : \quad m = m_0 \; and \; n \in Nbr(i) \\ \mathbb{A}_{m,n} & : \quad otherwise \end{cases}$$

A. The Concept of Distributed Coordination

Assuming the spectrum allocation was near optimal before the topology/spectrum availability change, distributed coordination between affected vertices can quickly optimize allocations for utilization and fairness. During distributed coordination, sets of neighboring vertices, each of which form a connected component of the conflict graph, self-organize into coordination groups. Each group modifies spectrum assignment within the group to improve system utility while ensuring that local changes in spectrum assignment do not conflict with nodes outside the group.

In this paper, we propose a simple coordination format where a node who wants to improve its spectrum assignment broadcasts a coordination request to its k-hop neighbors, where k is the ratio of interference range to transmission range. These neighbors are connected to the node in the corresponding conflict graph. Neighbors willing to coordinate reply to the sender and form a coordination group. Note that it is possible that two connected neighbors in a conflict graph might not be able to communicate directly with each other, *i.e.* when $k > 1$. The coordination information can be relayed by the nodes in between, although these relay nodes do not necessary participate in the coordination group.

B. Regulated Coordination for Fast System Convergence

To perform coordination, we must first determine the size and membership of coordination groups. Large groups increase the complexity of coordination due to high synchronization and communication costs. In addition, interactions might occur between coordination groups if they share neighboring nodes. Careless coordination will lead to conflicts among concurrent coordination groups and cascading effects that prevent system from converging. To facilitate coordination, we propose two constraints to regulate the coordination procedure [9].

Constraint 1: Limited Neighbor Coordination .

While pair-wise coordination is already a hard convex optimization problem, coordination within a large group implies even higher computation and communication overheads. Coordinating around a central leader per group can greatly simplify spectrum assignment. For each coordination group, the requester becomes the group coordinator and performs the coordination computation. The coordination strategies can be divided into the following formats:

- *One-to-one coordination*- The node n_1 who initiates the coordination can choose to coordinate with only one neighbor n_2 at a time. They exchange some channels to improve system utility while complying with the conflict constraints from the other neighbors. The requester only needs approval from one of his neighbors to perform the coordination. When multiple neighbors *e.g.* n_2 and n_3 acknowledge the coordination request, n_1 can perform coordination sequentially or expand the coordination group to (n_1, n_2, n_3). However, this also requires that n_1 chooses a sequential coordination order and gets

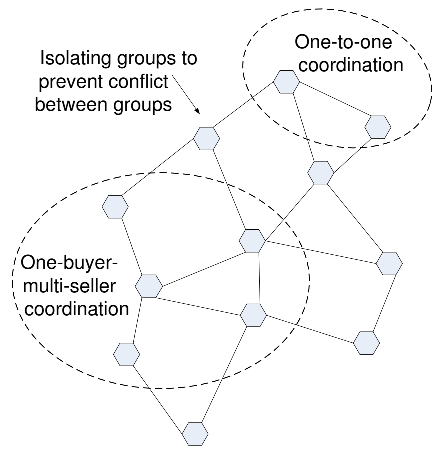


Fig. 3. An example of Coordination Groups.

approval from all the group members on the order before conducting coordination. Hence, for simplicity, we restrict this format to only one-to-one coordination. Fig. 3 illustrates an example of one-to-one coordination.

- *One-buyer-multi-seller coordination*- A buyer node n_1 purchases a set of channels M_0, from its neighbors who are currently using any channel in M_0, such that to improve system utility. In this case, the coordination requires concurrent approval from multiple neighbors. As we will show later, this type of coordination is necessary to eliminate node starvation. Fig. 3 illustrates an example with one buyer and four sellers.

Constraint 2: Self-contained Group Coordination

Once the coordination groups are organized, the coordination inside each group should not disturb the spectrum assignment at nodes outside the group. That is, after the coordination, the modified channel assignment should not lead to any conflict with nodes outside the group. This helps to maintain system stability, so that a coordination may not invoke a series of reactions due to violations in interference constraints. More importantly, this guarantees that if a coordination improves the utility in a local area, it also improves the system utility. Or in other words, a local improvement will lead to a system improvement. This constraint has two components.

- *Restricted Exchangeable Channels*– This restricts the set of channels that are exchangeable between nodes inside each coordination group, such that when a node gets one channel from its neighbor, the assignment does not conflict with its neighbors outside the coordination group.
- *Isolated Coordination Group*– This not only restricts each node to participate in at most one coordination group at any time, but also requires that the members of any two coordination groups can not be directly connected. Having nodes between groups regulate spectrum adjustment and prevents conflict between groups. The necessity of this requirement can be explained by the following example. Assume there are two neighboring nodes A and B (two nodes in the conflict graph connected with an edge) who are the members of two different coordination groups. Before assignment, A and B are not using channel 0. After the coordination, both A and B are granted with channel 0 from their coordination neighbors. However, as A and B conflict with each other if using the same channel, the coordination produces interference conflict among nodes. The detailed procedure to form isolated coordination groups will be introduced in Section III-C. An example of isolation between coordination groups is shown in Fig. 3.

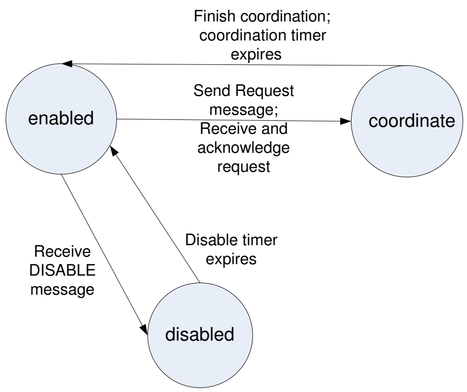


Fig. 4. Node State and Transitions

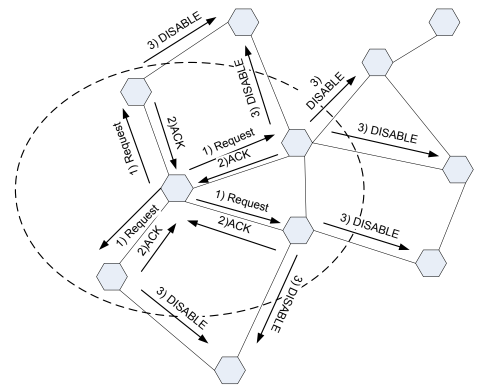


Fig. 5. Messages exchanged during coordination.

C. Coordination Protocol Design

We design a coordination protocol based on the above constraints. We propose a *distributed, iterative grouping and coordination process*. We assume that nodes periodically broadcast their current channel assignment and interference constraints to their neighbors. Each node switches among three states: *coordination*, *disabled* and *enabled* (see Fig. 4). Only *enabled* nodes can perform coordination. The actual coordination involves the following 4 steps, and repeats until no further coordination can improve system utility. Here nodes refer to the vertices in the conflict graph, and two "connected" nodes might be physically k $(k > 1)$ hops away. Information exchange between them is done through relay.

(1) Initialize Coordination Request– In general, nodes affected by mobility events initialize coordination. Based on broadcasts of channel assignments and interference constraints from neighbors, an *enabled* node determines if coordination with a neighbor will lead to an improvement in system utility. If such neighbors exist, the node broadcasts a coordination request to the neighbors along with its current channel assignment and interference constraints. Such broadcasts reduces communication overhead. As we will show in Section IV-C, additional criterion exists to guide nodes on generating coordination requests.

(2) Acknowledge Coordination Request– Neighbors who are *enabled* and willing to coordinate reply an *ACK* message with its current channel assignment and interference constraints. We assume that nodes are willing to collaborate to improve system utility, and accept requests that improve system utility even if it might degrade their individual channel assignments. Incentive systems to encourage such collaboration will be investigated in a future paper. If a node receives multiple concurrent requests from its neighbors, it acknowledges the request that leads to the highest coordination gain calculated from information embedded in the request.

(3) Coordination Group Formation– When the requester receives the replies, it selects the members of the coordination group, and broadcasts this information along with the proposed modification of the channel assignment to neighbors. Once the coordination group is set, its members enter *coordination* state. They broadcast a *DISABLE* message with a timer equal to the estimated duration of the coordination process to neighbors not in the coordination group. Note that the *DISABLE* message can be embedded in the *ACK* messages to reduce overhead. Nodes receiving the message enter *disabled* state for the duration of the timer. This procedure prevents nodes who are neighbors of existing coordination group to participate in any future coordination before the timer expires. Following this, all coordination groups are isolated.

(4) Coordination– Once all members acknowledge the changes to the channel assignment, each member updates its local channel assignment. This is straightforward for one-to-one coordination. For one-buyer-multiple-seller coordination, interactions among

members can be coordinated by the coordination requestor. After coordination, each member enters *enabled* state. Fig. 4 and 5 illustrate the node state transition and messages during coordination.

D. Choosing Coordination Strategies

Once the distributed coordination procedure is set, the specific coordination strategy may be customized for different utility functions. It is easy to show that for utilization based utility (total node throughput), the optimization can be reduced to solving M optimization problems for each color respectively. On each color, the corresponding optimization problem is exactly a Weighted Independent Set (WIS) problem [3]. WIS problem is a special case of Weighted Set Packing problem, and can be approximated by a local improvement heuristic algorithm, generally called $t-improvement$ [16], [3]. It is straightforward to convert this algorithm to distributed coordination among neighbors. We omit the coordination procedure due to space constraints, and next focus on the distributed coordination strategy for the fairness-based utility.

IV. A Case Study: Distributed Coordination to Improve Fairness

In this section, we focus on the coordination strategy optimizing for fairness. Based on its definition in (1), the optimization aims to maximize the total logarithmic node throughput, *i.e.* the product of node throughput or the amount of spectrum assigned. Therefore, the global fairness utility increases if nodes with many assigned channels "give" some channels to nodes with few assigned channels. In this section, we start by describing basic one-to-one coordination where two unbalanced nodes exchange channels to improve the local throughput product. We show that such coordination is limited by the number of exchangeable channels and thus not effective against the node starvation problem. We then develop a special case of one-buyer-multi-seller coordination, referred to as *Feed Poverty* to eliminate node starvation. We also derive a theoretical lower bound of node throughput using distributed coordination under a simplified network configuration.

We first define the following notations.

n: a node n in the conflict graph ($0 \leq n \leq N-1$);

$Nbr(n) = \{v \in V | (n,v) \in E\}$: neighbors of n;

$Nbr(X) = \bigcup_{n \in X} Nbr(n) \setminus X$: neighbors of node set X;

$f_A(n) = \{0 \leq m \leq M-1 | a_{m,n} = 1\}$: the set of channels assigned to node n under current assignment A.

A. One-to-One Fairness Coordination

As we described, one-to-one coordination allows two neighboring nodes n_1 and n_2 to exchange channels to improve system utility while complying with conflict constraints from the other neighbors. For n_1 and n_2 to coordinate, they need to first obtain the channels that are exchangeable to avoid disturbing other neighbors, referred to as $C_b(n_1, n_2)$:

$$C_b(n_1,n_2) = L(n_1) \cap L(n_2) \cap \{\{0..M-1\} \setminus \bigcup_{n \in Nbr(n_1,n_2)} f_A(n)\}.$$

Given $C_b(n_1, n_2)$, we can define the one-to-one coordination regarding fairness as follows:

Definition 1: For an assignment $A_{M \times N}$, an *One-to-One Fairness Coordination* finds nodes n_1 and n_2, and their coordination channel set $C_b(n_1, n_2)$, and modifies $A_{M \times N}$ to $A'_{M \times N}$ related to n_1, n_2 and channels $C_b(n_1, n_2)$, such that $R_{A'}(n_1) \cdot R_{A'}(n_2) > R_A(n_1) \cdot R_A(n_2)$.

The One-to-One Fairness Coordination increases the product of the coordination nodes while other nodes' throughput values remain unaffected. Hence, the system fairness increases with each coordination. The improvement between each pair of nodes

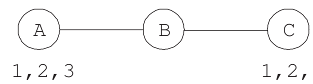


Fig. 6. An example of Starvation

(n_1, n_2) can be calculated as $G(n_1, n_2) = \frac{R_{A'}(n_1)R_{A'}(n_2)}{R_A(n_1)R_A(n_2)} - 1$. Nodes use this criterion to (in Section III) to determine whether a coordination can improve system utility.

Given (n_1, n_2), assigning channels to n_1 and n_2 to maximize their throughput product is a difficult task. This is because node throughput depends on all channels (including non-exchangeable ones) assigned to a node, and the available bandwidth on a channel differs between nodes. The problem is shown to belong to the class of convex programming problems [51]. When the number of exchangeable channels ($|C_b(n_1, n_2)|$) is small (*e.g.* $|C_b(n_1, n_2)| < 10$), exhaustive search may be feasible. Otherwise we need to use approximations based on heuristics such as the one given in [51]: first sort channels in $C_b(n_1, n_2)$ (by channel bandwidth), then use a two-band partition to determine the allocation.

The effectiveness of One-to-One coordination is constrained by the size of $C_b(n_1, n_2)$. In general, due to heavy interference constraints among neighboring nodes, $C_b(.)$ could be very small. Figure 6 illustrates an example where the conflict graph is a chain topology consisting of three nodes A,B,C. Node B is not assigned with any channel and the system utility is zero. We refer to this as node *starvation*. Node a and b cannot modify their spectrum usages due to the constraint from c (*i.e.* $C_b(a, b) = \emptyset$), while node b and c also cannot coordinate due to the constraint from a (*i.e.* $C_b(b, c) = \emptyset$). Hence, the *Fairness Coordination* is not effective to eliminate node *starvation*.

B. Feed Poverty Coordination

We observe that node *starvation* in most cases is a result of the lack of flexibility in coordination. As for the example in Figure 6, by allowing A and C to give up channel 1 at the same time and feed it to B, we can remove the starvation at B. This is an example of one-buyer-multi-seller coordination. In this paper, we propose a special one-buyer-multi-seller coordination, called *Feed Poverty* where if a node (buyer) has very poor channel assignment, the neighboring nodes can collaborate together to feed it with some channels.

Definition 2: For an assignment $\mathbb{A}_{M\times N}$, a *Feed Poverty Coordination* is to find some node n_0 and channel m_0, modify $\mathbb{A}_{M\times N}$ to $\mathbb{A}'_{M\times N}$, such that

$$\mathbb{A}'_{m,n} = \begin{cases} 1 & : \quad m = m_0 \; and \; n = n_0 \\ 0 & : \quad m = m_0 \; and \; n \in Nbr(n_0) \\ \mathbb{A}_{m,n} & : \quad otherwise \end{cases}$$

(intuitively, the assignment let some of n_0's neighbors give up channel m_0 and feed it to n_0) and

$$\begin{aligned} G_{FP}(n_0) &= (R_{\mathbb{A}'}(n_0)) \cdot \prod_{n \in Nbr(n_0) \wedge \mathbb{A}_{m_0,n}=1} (R_{\mathbb{A}'}(n)) - (R_{\mathbb{A}}(n_0)) \cdot \prod_{n \in Nbr(n_0) \wedge \mathbb{A}_{m_0,n}=1} (R_{\mathbb{A}}(n)) \\ &> 0. \end{aligned}$$

This means the product-throughput of the nodes involved in the coordination is locally increasing, while the other nodes' throughput are not affected. So generally the coordination improves system utility, except that, in case of starvation of other nodes, the system utility remains $-\infty$. A special case of Feed Poverty is when $\mathbb{A}_{m_0,n} = 0$ for all $n \in Nbr(n_0)$. This means none of n_0's neighbors are using channel m_0, and n_0 simply seizes it.

When there is no feasible *one-to-one Fairness Coordination*, *i.e.* $|C_b| = \emptyset$, the requestor initializes a *Feed Poverty Coordination* on all neighbors who acknowledge the request. The requestor sequentially selects multiple channels to maximize group utility.

C. FC-Optimal Assignment

We combine *one-to-one Fairness Coordination* and *Feed Poverty Coordination* into a *Fairness Coordination with Feed Poverty* (FC). Each node who wants to improve its spectrum usage starts with *one-to-one Fairness Coordination* with its neighbors to improve system utility. If there is no exchangeable channels, a *poor* node can broadcast a Feed-Poverty request to its neighbors to initialize *Feed Poverty Coordination*. Overall, a channel assignment A is said to be *FC-optimal* if no further coordination can improve system utility.

V. Theoretical Analysis

In this section, we perform theoretical analysis on the proposed coordination approach. We examine system utility, node throughput and algorithm complexity for FC-optimal assignments. The analytical results provide insights to fairness provided and system efficiency.

A. Lower Bound on Node Throughput/Spectrum

We first examine the node throughput under any FC-optimal assignment, *i.e.* the node performance when system stabilizes. With the goal of maximizing system proportional fairness, we show that each node's throughput is low-bounded.

Theorem 1: Under a FC-optimal assignment A, for each vertex n in the conflict graph G, $0 \le n \le N-1$, its throughput $R(n)$ has a lower bound, i.e.

$$\begin{aligned} R(n) &> \frac{B(n)}{d(n)+1} - MB(n) \\ B(n) &\triangleq \sum_{m=0}^{M-1} b_{m,n}, \quad MB(n) \triangleq \max_{m=0}^{M-1} b_{m,n} \end{aligned} \tag{2}$$

where $d(n)$ represents the number of conflicting neighbors of n or the degree of n in the conflict graph, $B(n)$ represents n's total available bandwidth, and $MB(n)$ represents n's maximum channel bandwidth.

The proof is included in Appendix A.

Theorem 1 shows that the low bound of each node's throughput depends on its interference condition. First, the bound of node n scales inversely with the number of competing nodes in n's neighborhood, *i.e.* $d(n)+1$. A node should obtain less throughput in crowded areas (with more conflicting neighbors) than in sparse areas. This scaling provides an immediate intuition of fairness.

Second, the bound scales linearly with the node's total available throughput $B(n)$. This trend demonstrates that the proposed strategy provides similar level of fairness to nodes regardless of its channel conditions. If a node improves its channel bandwidth using sophisticated physical layer techniques, without changing its interference condition, the throughput bound increases linearly, but the number of assigned channels remains the same. This also shows that our proposed strategy will not favor nodes in good channel conditions and starve nodes in bad channel conditions.

B. Poverty Line on Node Spectrum

Under certain circumstances, channel quality fluctuates due to fading, shadowing and environmental factors, making it impractical to collect channel quality of neighbors in real time. Hence, a reasonable approach is to assume all channels are identical (i.e. with bandwidth 1 if it is available, and with bandwidth 0 if it is not). Next, we show that Theorem 1 can be reduced to the following:

Theorem 2: Under a FC-optimal assignment A, for each vertex n in the conflict graph G, $0 \leq n \leq N-1$ with degree $d(n)$ and channel availability list $L(n)$, its spectrum usage $R(n)$ has a lower bound, i.e.

$$R(n) \geq \left\lfloor \frac{|L(n)|}{d(n)+1} \right\rfloor \triangleq PL(n).$$

The proof is straightforward using Theorem 1, included in Appendix B. Theorem 2 shows that the proposed *Fairness Coordination with Feed Poverty* guarantees a *poverty line* $PL(n)$ to each vertex n. The *poverty line* provides a guideline in coordination in real systems where a vertex is entitled to request coordination if its current throughput is below its poverty line. We refer to this as the *Poverty guided coordination*.

It is easy to show that if channels are fully available at each vertex, *i.e.* $|L(n)| = M$, a FC-optimal assignment can eliminate node starvation if the number of channels $M \geq \Delta + 1$, where Δ is the maximum degree in the graph $\Delta = \max_{0 \leq n < N} d(n)$. This matches to the well-known conclusion in graph coloring where the chromatic number of a graph is at most $\Delta + 1$ [11]. It can also be shown that the bound in Theorem 2 is tight, for several types of network topology: clique, ring, star, and straight line. We omit the proof due to space constraints. For randomly generated topologies, we obtain statistical results on the tightness of the bound using experiments in section VI-C.

C. Upper Bound on Coordination Complexity

Using *Poverty guided coordination*, only the node below its *poverty line* is qualified to initiate a coordination. The requestor n can sequentially select the best (*i.e.* providing the highest improvement to system utility) channels, and negotiate with the neighboring nodes to take over these channels. Following the detailed proof for Theorem 2, we can show that after each coordination iteration, one node will reach its poverty line. When the system reaches an equilibrium, every node is guaranteed with its poverty line. This also allows us to derive an upper bound on the number of coordination iterations.

Theorem 3: In a system with N nodes, with uniform channel bandwidth, the Poverty guided coordination *will reach an equilibrium after an expected number of at most $O(N^2)$ iterations. By optimizing the order of coordination, the system can reach an equilibrium in at most N iterations.*

The proof is in Appendix C.

It should be noted that the $O(N^2)$ bound in Theorem 3 is a *theoretical upper bound* on the complexity of the coordination system, which does not necessarily mean that the complexity of the system will scale in a $O(N^2)$ trend in practice. Actually as we observed in our simulation (Section VI-D), in average cases the complexity of the system scales linearly as the number of nodes increases.

D. Theoretical Distance to Social Optimal

We also compare the performance of the proposed coordination strategy, to the social optimal assignment. A social optimal assignment maximizes the global system utility. We focus on the *price of anarchy (POA)* [30], defined as the ratio of the system utility of social optimal assignment over the worse case of the proposed assignment strategies. However, without restriction on channel bandwidth or number of nodes, the POA is unbounded in general. Therefore, we restrict ourselves to the case with uniform channel bandwidth, *i.e.* $b_{m,n} = 1$ for all m, n, and a system of N nodes. Next we derive the upper bound of the POA.

Theorem 4: For a topology with uniform channel availability and channel bandwidth, i.e. $b_{m,n} = 1$*,* $l_{m,n} = 1$*, and* $M > \Delta$[4]*, the price of anarchy for a FC-optimal channel assignment is at most*

$$\frac{M^N \cdot \prod_{\langle u,v \rangle} \left(\frac{d_v}{d_u+d_v}\right)^{\frac{1}{d_u}} \cdot \left(\frac{d_u}{d_u+d_v}\right)^{\frac{1}{d_v}}}{\prod_{n=0}^{N-1} \left\lfloor \frac{M}{d_n+1} \right\rfloor}. \tag{3}$$

The proof can be found in [9].

VI. Experimental Results

A. Experiment Setup

We conduct experiments to quantify the performance of distributed coordination against both the centralized approaches and the theoretical lower bounds. For default scenarios, we assume that channels are equally weighted and all the channels are available for each node, *i.e.* $l_{n,m} = 1,\ b_{n,m} = 1$. We assume that all active nodes have backlogged traffic.

We simulate a dynamic spectrum network by randomly placing a set of nodes on a 100×100 area. Each node represents a WiFi or WiMAX access point, who serves a large number of subscribers and have backlogged traffic. We further abstract the network into a *Conflict Graph* where each vertex represents a transmitting node. Any two nodes interfere with each other (*i.e.* connected in the conflict graph) if they are within distance of 20[5]. For default scenarios, we assume that channels are equally weighted and all the channels are available for each node, *i.e.* $l_{n,m} = 1,\ b_{n,m} = 1$.

We model topology and traffic dynamics by turning nodes on and off. However, the on-off period is much longer than the coordination period so that the system can converge to the optimal spectrum assignment. We divide time into slots, and in each time slot, a random $p\%$ of nodes that are currently off become on and $p\%$ of nodes that are currently on become off. In each time slot, after the topology change, nodes adjust their channel usage. We also randomly deploy primary users and introduce dynamics to spectrum availability at each node. Finally, we introduce heterogeneous channel bandwidth and examine the impact on the tightness of poverty line. We use these simple scenarios to demonstrate the impact of network dynamics and examine the system convergence time and coordination overhead. We use two metrics to evaluate the performance.

System utility – We consider fairness defined in (1). Note that if there exists a node with no channel assigned, the utility becomes $-\infty$. For better representation, we modify the utility to $U(A) = \sqrt[N]{\prod_{n=0}^{N-1} R_A(n)}$ and $U(A) = 0$ if there is any $R_A(n) = 0$.

Communication overhead – We quantify algorithm complexity as the communication overhead, *i.e.* total number of messages exchanged among nodes, since transmission and handling of messages will likely dominate computations for channel assignment. In both distributed coordination and graph coloring approaches, each iteration of spectrum assignment or coordination involves a 4-way handshake between neighbors, *i.e.* (request, acknowledgement, action, acknowledgement).

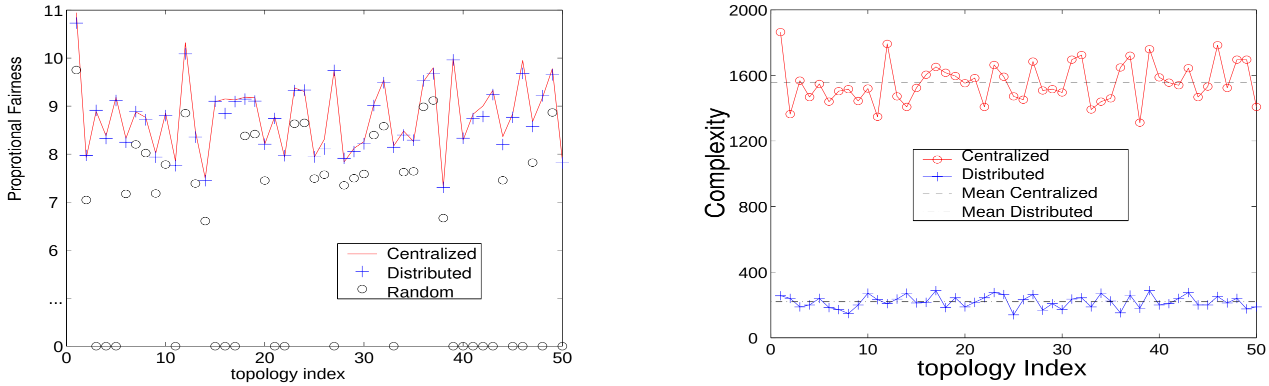
B. Comparison with Centralized Graph Coloring Approach

We compare the proposed distributed coordination to the centralized graph-coloring approach of [40]. We randomly deploy 40 links with 30 channels in a given area and produce the corresponding conflict graph.

(1) Static Networks – In this case, distributed coordination starts from a random allocation and gradually improve system utility until the system converges. Figure 7 compares the system utility and algorithm complexity using *centralized* graph coloring, *distributed* coordination and *random* assignment. Node starvation is common when using random assignment, resulting in

[4] otherwise there may exist starvation and the ratio may be meaningless.

[5] We use 20 as an illustrative example. The actual distance threshold depends on the choice of transmission power and radio hardware.



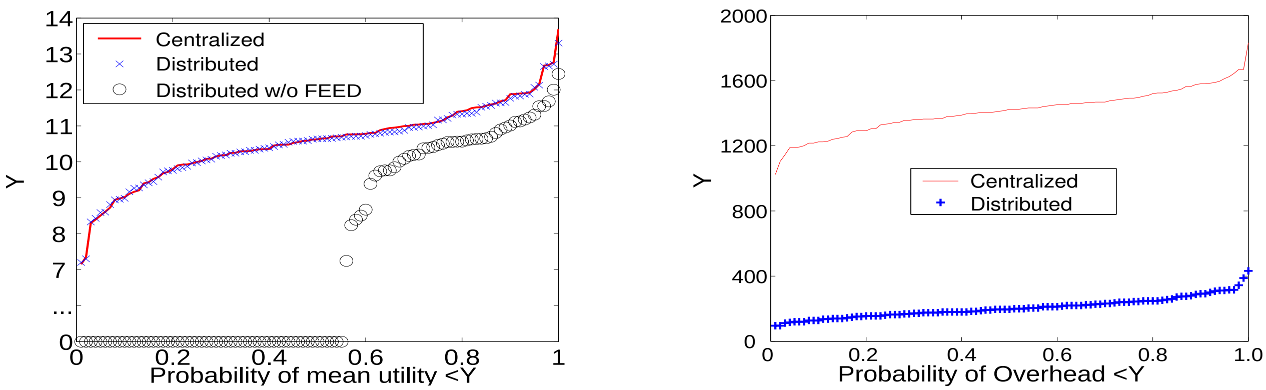
(a) System Utility

(b) Algorithm Complexity

Fig. 7. Performance comparison of *centralized* graph coloring, *distributed* coordination and *random* spectrum assignment for various topologies.

many zero values for system utility. Distributed coordination can effectively eliminate node starvation and performs only slightly worse compared to graph coloring approach. Figure 7(b) shows that distributed coordination can significantly reduce communication overhead.

(2) Dynamic Networks – We randomly deploy 40 links with 30 channels in a given area and produce the corresponding conflict graph. We use the graph coloring approach to derive an initial spectrum assignment for the given conflict graph. We simulate dynamic events in the next 100 time slots, one event per time slot where up to 6 nodes turn on and 6 nodes turn off. After each event, we apply both distributed coordination and graph-coloring approaches to derive the new spectrum allocation.



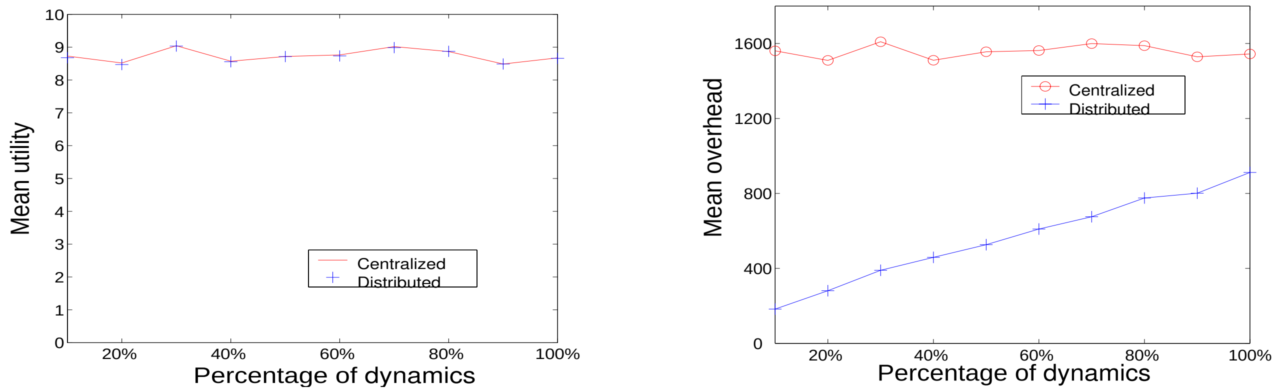
(a) System Utility

(b) Algorithm Complexity

Fig. 8. Performance comparison of centralized graph coloring and distributed coordination.

Figure 8(a) illustrates the sorted fairness utility using both approaches, and distributed coordination performs nearly as good as the graph coloring approach. The graph coloring approach makes decisions with the knowledge of global topology, while using distributed coordination, each node makes decisions based on only neighbor information. Figure 8(b) compares the communication overhead in each time slot. We observe that distributed coordination achieves similar performance while incurring much lower complexity in terms of messages exchanged. This significant overhead reduction allows quick adaptation to network dynamics. In Figure 8(a), we also examine the performance of distributed coordination using only *one-to-one fairness coordination*, without *Feed Poverty Coordination*, referred to as *distributed w/o feed*. The results confirm that *Feed Poverty Coordination* is critical to eliminate *starvation*.

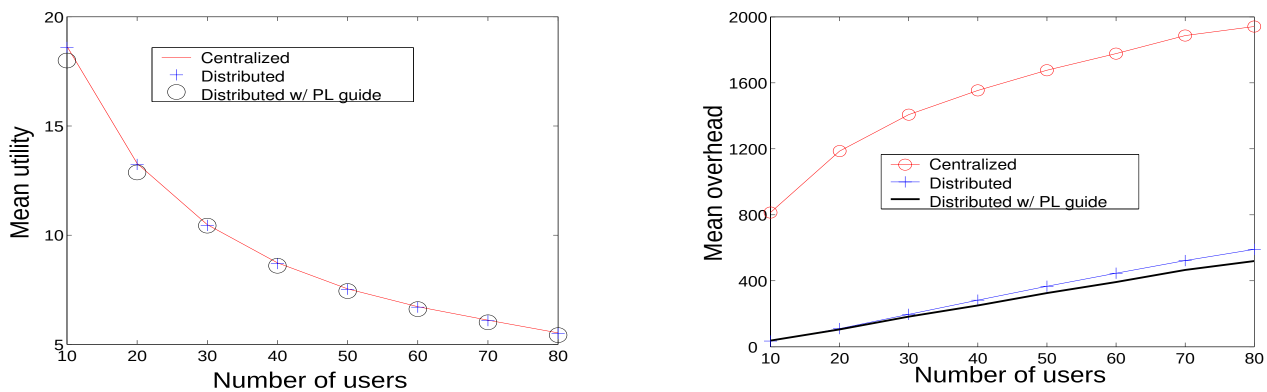
Next, we extend the simulation to allow $p\%$ of currently on vertices to turn off and the same number of vertices to turn



(a) System Utility

(b) Algorithm Complexity

Fig. 9. Performance comparison of centralized graph coloring and distributed coordination under different rate of network dynamics.



(a) System Utility

(b) Algorithm Complexity

Fig. 10. Performance comparison of centralized and distributed approaches under different vertex density.

on from random locations. In general, larger p implies more disturbance to the conflict graph and thus more vertices will perform coordination to adapt their spectrum usage to the new topology. Figure 9 illustrates the system utility and algorithm overhead for distributed coordination and graph coloring for increasing values of p. The utility is geometrically averaged over 100 time slots, and overhead is averaged over 100 time slots. As before, distributed coordination performs similarly to graph coloring approach in system utility. The overhead of graph coloring is not sensitive to the value of p as it mainly depends on the number of vertices and channels. The overhead complexity of distributed coordination increases with p as more vertices need to perform coordination. We observe that even under 100% network dynamics, distributed coordination reduces overhead by half compared to the graph coloring approach. Therefore, distributed coordination appears to be an attractive alternative to graph coloring for spectrum allocation on a given network topology.

In Figure 10, we examine the impact of network size at $p = 20\%$ dynamics and 40 channels. Increasing vertex density results in additional interference and increases average vertex degree in the conflict graph. Therefore, system utility scales inversely with the number of vertices while the algorithm complexity increases. As before, results show that distributed coordination compares favorably to graph coloring in quality of allocation while incurring significantly less overhead.

C. Tightness of the Poverty Line Bound

We now examine the appropriateness of the node poverty bound derived in Theorem 2.

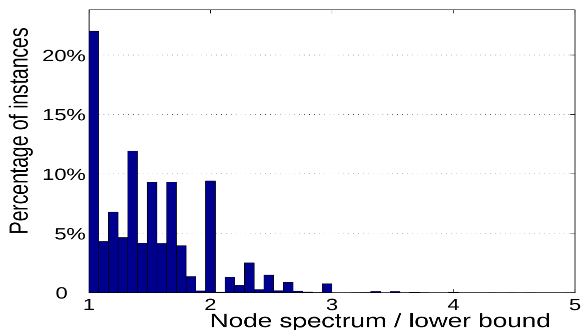


Fig. 11. Histogram of the ratio between the actual node throughput and the poverty bound.

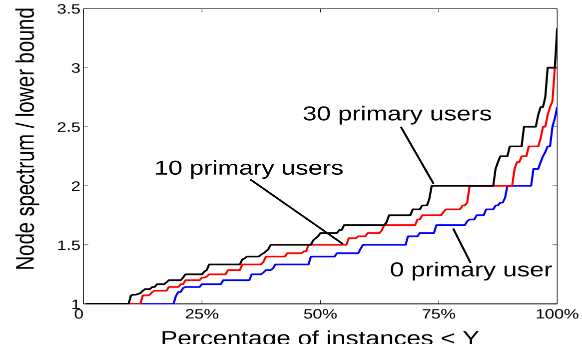


Fig. 12. Tightness of the bound when there are primary users.

(1) Static Networks – Figure 11 illustrates the histogram of the ratio of the actual node throughput and the poverty bound assuming 40 vertices and 100 time slots. Results show that the theoretical bound is valid and fairly tight. As we described, nodes can use the poverty line to decide whether further distributed coordination is necessary. A vertex with an assignment below the poverty line should bargain with additional neighbors to acquire additional channels. In Figure 10, we also compare the performance of *Poverty guided coordination* to the graph coloring and coordination approaches. We show that *Poverty guided coordination* performs close to that of the coordination but with 10% less overhead, again demonstrating the tightness of the poverty line bound.

(2) Dynamic Spectrum Availability – Next, we examine the impact of primary user deployment on the bound. We randomly deploy 10 and 30 primary users to our previous experiments. Primary users have the same interference range as secondary nodes, and occupy one randomly chosen channel. Figure 12 shows the CDF of the ratio between actual node throughput and lower bound. We observe that the bound becomes slightly looser as the number of primary users increases. This is mainly due to the mismatch between $L(n)$ and $d(n)$. We calculate $d(n)$ to include all the interfering neighbors who has at least one channel in common with n. Therefore, $d(n)$ is the upper bound of the conflicting neighbors on each channel.

(3) Heterogenous Spectrum Bandwidth – When channel bandwidth is non-uniform, we compute the bound following Theorem 1. Figure 13 compares the tightness of the bound when channel bandwidth varies between 1 to 3 and 1 to 5. The result is relatively smoother by eliminating the truncation effect in Theorem 2. We see that the bound is looser compared to that of the uniformed bandwidth. This is mainly due to the fact that the lower bound is only a sub-bound, *i.e.* a node's throughput is always *strictly* larger than the bound. The figure also shows that the bound is sensitive to the variance in bandwidth across channels.

D. Scalability of Distributed Coordination

We also evaluate the complexity of coordination in large scale networks. We measure the total system overhead, *i.e.* the total number of message exchanges for a system to reach an equilibrium. We keep the node density constant and vary the system scale from 200 to 1000, and assume 20% of node dynamics. Results in Figure 14 together with those in Figure 10(b) indicates that the system overhead scales linearly with the number of nodes. Hence, the average overhead per node is roughly constant, 8 messages or 2 coordination iterations under the above system configurations. This result demonstrates the efficiency of the proposed approach.

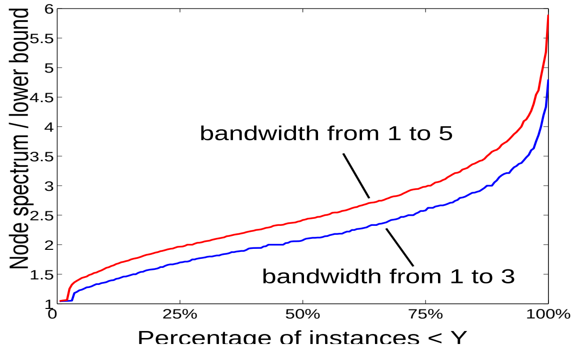


Fig. 13. Tightness of the bound in theorem 1 when channel bandwidth is not uniform.

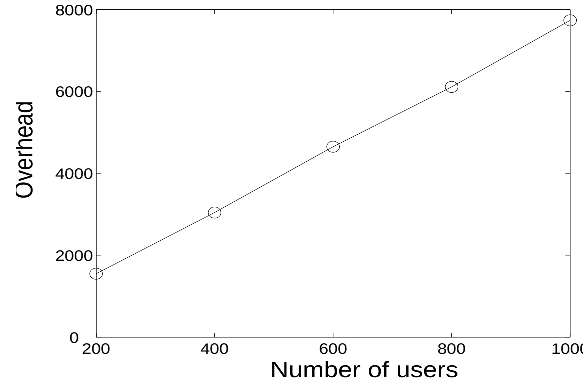


Fig. 14. System overhead in large scale networks

E. Impact of Network Topology

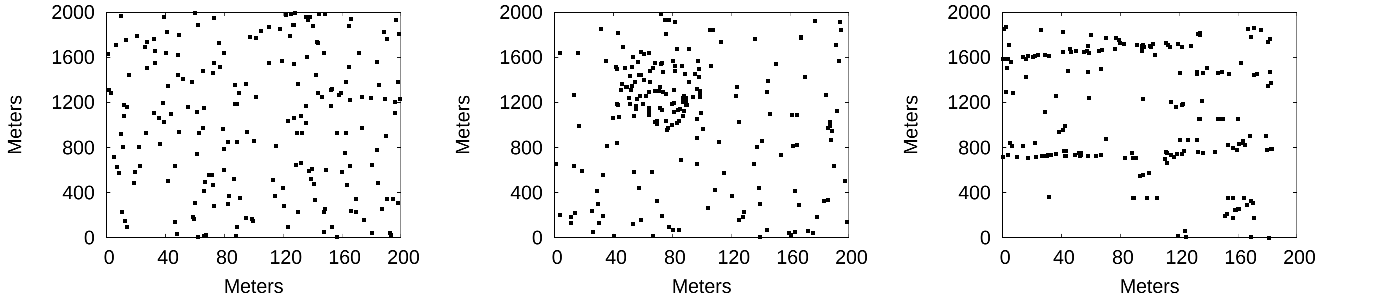
To examine the impact of network topology, we use both randomly generated topologies and measured AP deployment traces.

- *Random network.* We place nodes randomly in an area.
- *Clustered network.* We simulate a hotspot scenario by deploying a set of nodes densely in a small area of the random network. We use this topology to examine the impact of conflict degree on system performance.
- *Real network trace.* We extract a set of actual AP deployments using data traces collected by Placelab (http://placelab.org/).

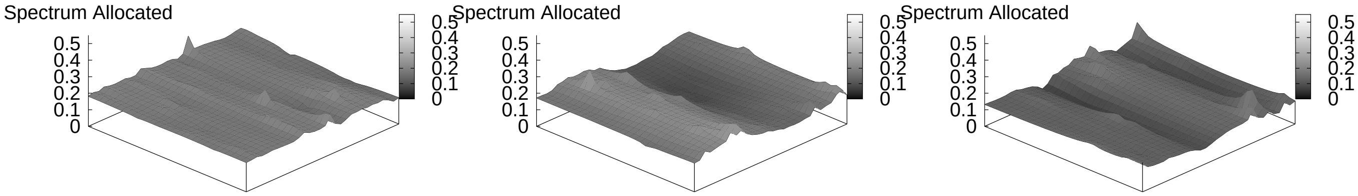
Figure 15(a) illustrates three sample topologies corresponding to random, clustered and measured networks, respectively. To make the size of the network consistent with the size of the real network, we set the network to be 200×2000, and two nodes conflict if they are within distance of 90. Every topology contains 200 nodes, and the number of channels is 100. We use 100 channels since most nodes in clustered and measured networks have degree larger than 30. Note that for this experiment the initial allocation is *the empty allocation*. Figure 15(b) plots the amount of spectrum assigned to each node using the explicit coordination. We observe that any node n obtains spectrum inversely proportional to its conflict degree $d(n)$, as characterized by the poverty line definition.

To further investigate this dependency, we plot in Figure 15(c) the amount of spectrum assigned at each node divided by its poverty line, as a function of the conflict degree $d(n)$. It can be observed that in the clustered network and the measured network, there are three kinds of nodes: 1) the nodes in the non-clustered area, which is consistent with the nodes in the random network; 2) the nodes at the clustered area, whose degrees are high; 3) the nodes near the edge of the clustered area, whose conflict degrees are between that of the clustered area and the non-clustered area (around 10-20). The results show that the Poverty Line bound is relatively tighter for the first two types of nodes, and becomes looser for the third type. In other words, the Poverty Line bound is more accurate for homogenous conflict conditions, which coincides with the theoretical results where the Poverty Line is tight for the line, clique, and ring topologies.

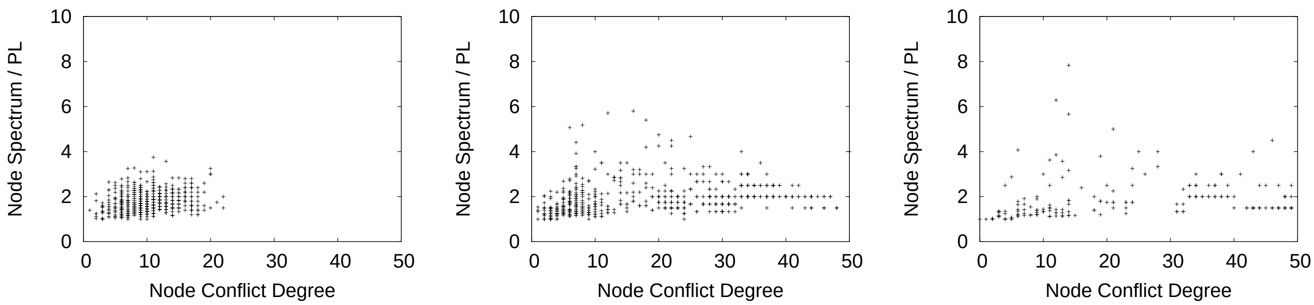
Figure 15(d) shows the communication overhead at each node as a function of the conflict degree $d(n)$. In the clustered network, the results show that the nodes at the edge of the cluster have more communication overhead. Further, we also observe that although the total complexity scales linearly with the network size, it also scales with the number of channels. However, we could not obtain the relationship of the complexity and the number of channels. We plan to study the individual complexity in a future work.



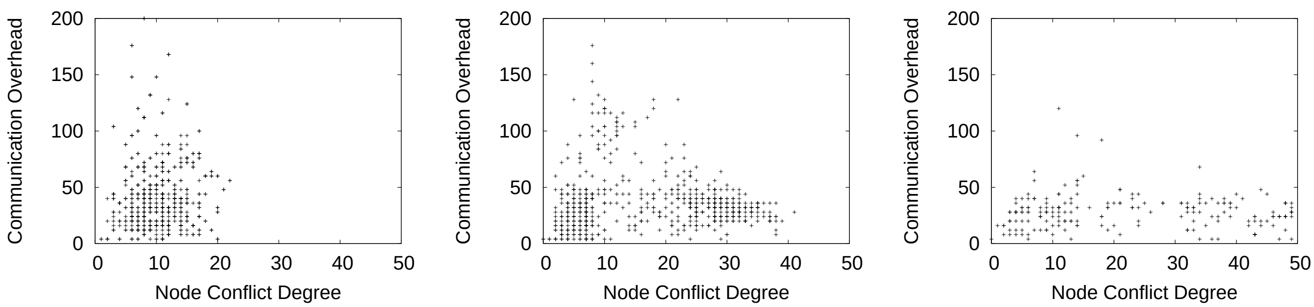
(a) Network topology



(b) The amount of spectrum assigned to each node.



(c) The tightness of Poverty Line bound.



(d) The communication overhead.

Fig. 15. The amount of spectrum assigned to each node for (left) random network, (center) clustered network and (right) placelab network.

VII. Practical Considerations

In this section, we discuss practical issues associated with our network model and problem definition.

Identifying channel availability and interference constraint. The proposed spectrum allocation system requires nodes (access points) to keep track of the channel availability matrix $\mathbb{L}$ and the interference constraint $\mathbb{C}$. We list three complementary mechanisms to obtain this information.

- Access points perform spectrum measurements to determine $\mathbb{L}$ and $\mathbb{C}$, using techniques like spectrum sensing [8], [14] and RSSI measurement-based probing [46].
- Access points can broadcast "hello" beacons [52] periodically and help peers construct the interference matrix.
- Clients associated with APs can assist the interference detection by sensing radio signals and provide feedback on findings of interfering access points [34]. This mechanism has been shown to help refine the interference map.

Extending to complex interference characterizations. In this paper, we use a binary matrix to model the pair-wise interference among access points. This model is widely used in existing works (e.g. [25], [34], [40]). We note that more realistic models are based on aggregated SNR measurements, referred to as the physical model [25]. However, under this model the corresponding channel allocation problem becomes significantly complex, because in order to precisely decide a node's incoming interference, every node in the whole network needs to be considered. We are currently investigating spectrum allocation schemes under this complex interference model, by reducing the global interference condition to a local interference condition, and then apply our local coordination approach. This reduction may be carried out by first bounding the interference coming from all the nodes with a distance more than a predefined distance.

Finally, in this paper, we assume interference conditions are homogeneous across channels. Our model is a reasonable abstraction of the scenario where channels are divided from a single band pool and have frequencies close to each other. When interference conditions are heterogeneous across channels, we can model the interference constraint channel-dependent, *i.e.* $c_{n,k,m}$, and then extend our local coordination approach to the channel-dependent case. For example, when using Feed Poverty Coordination, a node can calculate the impact of grabbing a specific channel from other nodes by taking this channel-dependent factor into account, *i.e.*, grabbing different channels will impact different set of neighbors. We are currently investigating analytical bounds on individual node spectrum usage under this case. Alternatively, we can always use an aggregated measure of interference across channels to define the interference constraint and apply the proposed solution, *i.e.* any two nodes conflict if they conflict on at lease one channel. The drawback of this approach is that this reduction is overly conservative and can degrade spectrum utilization.

VIII. Related Work

Optimal conflict-free channel assignment satisfying a global optimal objective is often NP-hard, even when global topology information is available [13]. We have summarized several existing works using centralized servers in Section II-C. In the following, we will summarize other related research results.

Practical strategies have been proposed for sharing a single channel. Contention based schemes invoke a random access protocol like ALOHA and CSMA, where nodes contend in time to share a common channel [23], [31], [36]. While this scheme provides fairness and utilization on a single channel system probabilistically, its application to a multi-channel system requires each node to know how many and which channel(s) to access. Another approach, conflict free time slot scheduling, provides guaranteed channel usage by reserving time slots for each flow. Solutions in [4], [43], [44] assign exactly one time slot to each flow. This approach can be used in multi-channel systems if each node uses only one channel. Another solution [48] allows nodes to use multiple slots/channels to achieve Max-Min-fair, but does not consider interference from neighbor transmissions.

Multi-channel assignment strategies were developed mostly for cellular networks. The work in [29] provides solutions to assign frequency bands among base stations to minimize call blocking probability for voice traffic. There is no notion of fairness as the traffic determines the number of channels each base station should use. In [17], the authors proposed a graph-theoretic model and discussed the price of anarchy under various topology conditions such as different channel numbers and bargaining strategies. The main difference between [17] and the proposed work is that the proposed model allows multi-coloring of a vertex, while in [17] each vertex can only be assigned with at most one color.

Cooperative/non-cooperative bargaining is also used in previous research to optimize channel allocation for cellular networks. The work in [18], [19] proposed a set of bargaining strategies for OFDMA based network, focusing on one-to-one bargaining. In these cases, nodes are mutually interfered, *i.e.* the corresponding conflict graph is fully-connected. Forming bargaining group is to find any two nodes in the network and let them exchange certain channels to improve system performance. The main difference between these work and the proposed work is that the proposed work provides solutions for general conflict graphs where group setup needs to consider local topology (*i.e.* isolated group and self-contained channel adjustment). We propose a feed-poverty coordination to eliminate node starvation which can not addressed by one-to-one coordination. In addition, the proposed work derives a lower bound for each node's channel assignment based on general topologies rather than all-connected conflict graph topologies.

IX. Conclusion and Future Work

In this paper, we present an adaptive and distributed approach for spectrum sharing in dynamic spectrum networks. We propose a distributed coordination approach where nodes self-organize into coordination groups and adapt their spectrum assignment to approximate the global optimal assignment. We propose a *Fairness Coordination with Feed Poverty* to improve fairness in spectrum assignment. We also derive a lower bound on the spectrum assignment that each node can get from coordination, referred to as the *poverty line*. It reflects the level of fairness enforced by the coordination, and serves as a guideline to organize coordination. Experimental results show that the proposed approach performs similarly as the centralized topology-optimized approach but with much less complexity. We also verify the correctness of the *poverty line* and the effectiveness of the *poverty guided coordination*.

While we only proposed a specific coordination strategy to maximize fairness based system utility, the proposed coordination framework can be extended towards other utility functions or optimization goal. We intend to examine this in a future study.

References

[1] Business 2.0 magazine: Get ready for googlenet, 2005.

[2] Akyildiz, I. F., Lee, W. Y., Vuran, M., and Mohanty, S. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks Journal (Elsevier)* (September 2006).

[3] Arkin, E. M., and Hassin, R. On local search for weighted k-set packing. In *Proc. of ESA, LNCS 1284* (1997), pp. 13–22.

[4] Bao, L., and Garcia-Luna-Aceves, J. J. Hybrid channel access scheduling in ad hoc networks. In *Proc. of ICNP* (October 2002).

[5] Berger, R. J. Open spectrum: a path to ubiquitous connectivity. *ACM Queue 1*, 3 (May 2003).

[6] Buddhikot, M. M., Kolody, P., Miller, S., Ryan, K., and Evans, J. DIMSUMNet: New directions inwireless networking using coordinated dynamic spectrum access. In *Proc. of IEEE WoWMoM* (June 2005).

[7] Buddhikot, M. M., and Ryan, K. Spectrum management in coordinated dynamic spectrum access based cellular networks. In *Proc. of IEEE DySPAN* (November 2005).

[8] Cabric, D., Mishra, S. M., and Brodersen, R. W. Implementation issues in spectrum sensing for cognitive radios. In *Proceedings of Asilomar conference on signals, systems and computers* (2004).

[9] Cao, L., and Zheng, H. Spectrum allocation in ad hoc networks via local bargaining. In *Proc. of SECON* (September 2005).

[10] Challapali, K., Mangold, S., and Zhong, Z. Spectrum Agile Radio: Detecting Spectrum Opportunities. In *International Symposium on Advanced Radio Technologies (ISART)* (March 2004).

[11] Chartrand, G. *A Scheduling Problem: An Introduction to Chromatic Numbers.* New York: Dover, pp.202-209, 1985.

[12] Clemens, N., and Rose, C. Intelligent power allocation strategies in an unlicensed spectrum. In *Proc. of IEEE DySPAN* (November 2005).

[13] Ephremides, A., and Truong, T. Scheduling broadcasts in multihop radio networks. *IEEE Trans. on Communications 38*, 4 (April 1990), 456–460.

[14] G. GANESAN, Y. L. Implementation issues in spectrum sensing for cognitive radios. In *Proc. IEEE DySPAN 2005* (November 2005).

[15] GAREY, M. R., AND JOHNSON, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1990.

[16] HALLDÓRSSON, M. M. Approximations of independent sets in graphs. In *Proc. of APPROX '98: Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization, LNCS 1444* (1998), Springer-Verlag, pp. 1–13.

[17] HALLDÓRSSON, M. M., HALPERN, J. Y., LI, L. E., AND MIRROKNI, V. S. On spectrum sharing games. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing* (2004), ACM Press, pp. 107–114.

[18] HAN, Z., JI, Z., AND LIU, K. R. Low-complexity OFDMA channel allocation with Nash bargaining solution fairness. In *Proc. of Globecom* (November-December 2004).

[19] HAN, Z., JI, Z., AND LIU, K. R. Power minimization for multi-cell OFDM networks using distributed non-cooperative game approach. In *Proc. of Globecom* (November-December 2004).

[20] HAYKIN, S. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications 23*, 2 (Feburary 2005), 201–220.

[21] HUANG, J., BERRY, R., AND HONIG, M. Auction mechanisms for distributed spectrum sharing. In *Proc. of 42nd Allerton Conference* (September 2004).

[22] HUANG, J., BERRY, R. A., AND HONIG, M. L. Spectrum sharing with distributed interference compensation. In *Proc. of IEEE DySPAN* (November 2005).

[23] HUANG, X., AND B.BENSAOU. On Max-Min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation. In *Proc. of MobiHoc* (2001), ACM.

[24] ILERI, O., SAMARDZIJA, D., AND MANDAYAM, N. B. Demand responsive pricing and competitive spectrum allocation via spectrum server. In *Proc. of IEEE DySPAN* (November 2005).

[25] JAIN, K., PADHYE, J., PADMANABHAN, V., AND QIU, L. Impact of interference on multi-hop wireless network performance. In *Mobicom 2003*.

[26] JAIN, K., PADHYE, J., V.N.PADMANABHA, AND QIU, L. Impact of interference on multi-hop wireless network performance. In *Proc. of MobiCom* (September 2003).

[27] JONDRAL, F. K. Software defined radio basics and evolution to cognitive radio. *EURASIP Journal on Wireless Communications and Networking 2005* (2005), 275–283.

[28] JONES, S. D., MERHEB, N., AND WANG, I.-J. An experiment for sensing-based opportunistic spectrum access in CSMA/CA networks. In *Proc. of IEEE DySPAN* (Baltimore, MD, Nov 2005).

[29] KATZELA, I., AND NAGHSHINEH, M. Channel assignment schems for celluar mobile telecommunication systems. *IEEE Personal Communications 3*, 3 (June 1996), 10–31.

[30] KOUTSOUPIAS, E., AND PAPADIMITRIOU, C. Worst-case equilibria. In *Proc. 16th Annual Conf. Theoretical Aspects of Computer Science* (1999), vol. 1563 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 404–413.

[31] LUO, H., LU, S., AND BHARGHAVAN, V. A new model for packet scheduling in multihop wireless networks. In *Proc. of MobiCom* (August 2000).

[32] MANGOLD, S., ZHONG, Z., CHALLAPALI, K., AND CHOU, C. T. Spectrum agile radio: Radio resource measurements for opportunistic spectrum usage. In *Proc. of Globecom* (November-December 2004).

[33] MCHENRY, M. Spectrum white space measurements. *New America Foundation Broadband Forum* (June 2003).

[34] MISHRA, A., SHRIVASTAVA, V., AGARWAL, D., BANERJEE, S., AND GANGULY, S. Distributed channel management in uncoordinated wireless environments. In *Mobicom 2006*.

[35] MITOLA III, J. Wireless architectures for the 21st century. http://ourworld.compuserve.com/homepages/jmitola.

[36] NANDAGOPAL, T., T.KIM, X.GAO, AND BHARGHAVAN, V. Achieving MAC layer fairness in wireless packet networks. In *Proc. of MobiCom* (August 2000).

[37] NEEL, J., REED, J., AND GILLES, R. The role of game theory in the analysis of software radio networks. In *Proc. Software Defined Radio Forum Technical Conference and Product Exhibition (SDR 02)* (San Diego, CA, USA, November 2002), vol. 2, pp. pp. NP–3–02.

[38] NIE, N., AND COMANICIU, C. Adaptive channel allocation spectrum etiquette for cognitive radio networks. In *Proc. of IEEE DySPAN* (November 2005).

[39] PEHA, J. M. Approaches to spectrum sharing. *IEEE Communications Magazine 43* (Feburary 2005), 10–12.

[40] PENG, C., ZHENG, H., AND ZHAO, B. Y. Utilization and fairness in spectrum assignemnt for opportunistic spectrum access. *Mobile Networks and Applications (MONET) 11* (May 2006), 555–576.

[41] POWELL, M. K. Broadband migration III: New directions in wireless policy. Remarks at the Silicon Flatirons Telecommunications Program, October 2002.

[42] RAMAN, C., YATES, R., AND MANDAYAM, N. Scheduling variable rate links via a spectrum server. In *Proc. of IEEE DySPAN* (November 2005).

[43] RAMANATHAN, S. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks 5*, 2 (March 1999), 81–94.

[44] RAMANATHAN, S., AND LLOYD, E. Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking 1*, 2 (April 1993), 166–177.

[45] RAYCHAUDHURI, D. Adaptive wireless networks using cognitive radios as a building block. MobiCom 2004 Keynote Speech, September 2004. Philadephia, PA.

[46] REIS, C., MAHAJAN, R., RODRIG, M., WETHERALL, D., AND ZAHORJAN, J. Measurement-based models of delivery and interference in static wireless networks. In *Proc. of ACM SigComm* (2006).

[47] RYAN, K., ARAVANTINOS, E., AND BUDDHIKOT, M. M. A new pricing model for next generation spectrum access. In *Proc. of TAPAS* (2006).

[48] SALONIDIS, T., AND TASSIULAS, L. Distributed on-line schedule adapation for balanced slot allocation in wireless ad hoc networks. In *Proc. of IWQoS* (June 2004).

[49] SANKARANARAYANAN, S., PAPADIMITRATOS, P., MISHRA, A., AND HERSHEY, S. A bandwidth sharing approach to improve licensed spectrum utilization. In *Proc. of IEEE DySPAN* (November 2005).

[50] STAPLE, G., AND WERBACH, K. The end of spectrum scarcity. *IEEE Spectrum 41*, 3 (2004), 48–52.

[51] YU, W., AND CIOFFI, J. FDMA capacity of gaussian multi-access channels with ISI. *IEEE Trans. Commun. 50*, 1 (Jan 2002), 102–111.

[52] ZHAO, J., ZHENG, H., AND YANG, G. Distributed coordination in dynamic spectrum allocation networks. In *Proc. of IEEE DySPAN* (November 2005).

[53] ZHENG, H., AND CAO, L. Device-centric spectrum management. In *Proc. of IEEE DySPAN* (November 2005).

[54] ZHENG, H., AND PENG, C. Collaboration and fairness in opportunistic spectrum access. In *Proc. of ICC* (June 2005).

APPENDIX

A. *Proof of Theorem 1*

We prove the theorem by contradiction: Assume that the bound (2) doesn't hold for node n, *i.e.*

$$R(n) \leq \frac{B(n)}{d(n)+1} - MB(n). \tag{4}$$

In the following we will show that n can request a Feed Poverty Coordination to improve the system utility. This contradicts with the assumption that assignment A is FC-optimal.

We divide the proof into two steps. First, we demonstrate how to find the proper channel to feed n. Next, we prove that system utility increases from this channel feed.

Step 1: channel selection. We assume that user n has $d(n)$ neighbors indexed $\{0, 1, \cdots, d(n)-1\}$. We assume that the channels assigned to user n, $f_A(n)$ are indexed by $\{M-1, M-2, \cdots, M-|f_A(n)|\}$. We define $t \triangleq M - |f_A(n)|$, where t is the number of channels not assigned to n. For each user n, we can organize the assignment matrix A using the following table, where an element is 1 if the channel indexed by the row number is assigned to the user indexed by the column number.

Index	0	$\cdots$	d(n)-1	n	$\cdots$
0	$\cdots$	$\cdots$	$\cdots$	0	$\cdots$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\cdots$	$\vdots$
$t-1$	$\cdots$	$\cdots$	$\cdots$	0	$\cdots$
t	0	$\cdots$	0	1	$\cdots$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
M-1	0	$\cdots$	0	1	$\cdots$

We focus on the first t rows and first $d(n)+1$ columns (i.e. column $0, 1, \cdots, d(n)-1, n$) of matrix A since they represent the channel assignment at user n's neighbors. We also examine $d(n)$ neighbors to exclude any starved user k, *i.e.* $R(k) = 0$. Let d represent the number of non-starved neighbors of n, $d \leq d(n)$, we re-index the neighbors of n to $(0, \cdots, d-1)$. In addition, from (4) we have

$$R(n) \leq \frac{B(n)}{d(n)+1} - MB(n) \leq \frac{B(n)}{d+1} - MB(n). \tag{5}$$

Integrating A with channel bandwidth matrix B, we construct an auxiliary matrix R with t rows and $d+1$ columns. The element of R can be represented as

$$R_{m,i} = \begin{cases} \frac{A_{m,i} \cdot b_{m,i}}{R(i)} & : \ 0 \le i \le d-1, \ 0 \le m \le t-1 \\ \frac{b_{m,i}}{B(i)/(d+1)} & : \ i = n, \ 0 \le m \le t-1. \end{cases}$$

It is straightforward to show that for each column $0 \le i \le d-1$,

$$\sum_{m=0}^{t-1} R_{m,i} = \sum_{m=0}^{t-1} \frac{A_{m,i} \cdot b_{m,i}}{R(i)} = 1. \tag{6}$$

and

$$\sum_{m=0}^{t-1} \sum_{i=0}^{d-1} R_{m,i} = \sum_{i=0}^{d-1} \sum_{m=0}^{t-1} R_{m,i} = d. \tag{7}$$

We can also show that for column n,

$$\begin{aligned} \sum_{m=0}^{t-1} R_{m,n} &= \sum_{m=0}^{t-1} \frac{b_{m,n}}{B(n)/(d+1)} = \frac{\sum_{m=0}^{t-1} b_{m,n}}{B(n)/(d+1)} = \frac{B(n) - R(n)}{B(n)/(d+1)} \\ (by(5)) \quad &\ge \frac{B(n) - (\frac{B(n)}{d+1} - MB(n))}{B(n)/(d+1)} > d. \end{aligned} \tag{8}$$

Combining (7) and (8), we prove that

$$\sum_{m=0}^{t-1} R_{m,n} > \sum_{m=0}^{t-1} \sum_{i=0}^{d-1} R_{m,i}$$

Therefore, there must exist a channel m_0, *s.t.*

$$R_{m_0,n} > \sum_{i=0}^{d-1} R_{m_0,i}. \tag{9}$$

Next we show that we can improve system utility by letting user 0 to $d-1$ give up channel m_0 and feed user n.

Step 2. Verify system utility. First we show that

$$R_{m_0,n} \le 1. \tag{10}$$

From (4),

$$\frac{B(n)}{d+1} - MB(n) \ge R(n) \ge 0 \implies \frac{B(n)}{d+1} \ge MB(n) \ge b_{m_0,n} \implies R_{m_0,n} = \frac{b_{m_0,n}}{B(n)/(d+1)} \le 1$$

Combining (10) with (9), we have

$$\sum_{i=0}^{d-1} R_{m_0,i} < 1 \implies R_{m_0,i} < 1, \quad 0 \le i \le d-1. \tag{11}$$

This shows that for each neighbor i, m_0 is not the only channel assigned. Therefore, after feeding user n with channel m_0, each neighbor has at least one channel assigned. The system utility after the feeding, depends on the value of $R(n)$ before the feeding.

Case 1) $R(n) = 0$. From (11), the feeding will increase $R(n)$ to $b_{m0,n}$ without starving any neighbor (0 to $d-1$). Clearly, the system utility is improved from $-\infty$ if n is the only starved user in the network before the feeding. However, when there are multiple starved users, the system utility remains $-\infty$. In this case, we define increasing system utility by reducing the number of starved users or improving the fairness utility of the non-starved users. Through a similar process, we can continue to remove other starved users to improve system utility.

Case 2) $R(n) > 0$. The ratio of system utility after the feeding to that before the feeding can be derived as

$$
\begin{aligned}
G_{FP} = \quad & \frac{(R(n)+b_{m_0,n})\cdot\prod_{i=0}^{d-1}(R(i)-A_{m_0,i}\cdot b_{m_0,i})}{(R(n))\cdot\prod_{i=0}^{d-1}(R(i))} \\
&= \frac{R(n)+b_{m_0,n}}{R(n)}\cdot\left(\prod_{i=0}^{d-1}\frac{R(i)-A_{m_0,i}\cdot b_{m_0,i}}{R(i)}\right) \\
&= \left(1+\frac{b_{m_0,n}}{R(n)}\right)\cdot\prod_{i=0}^{d-1}\left(1-\frac{A_{m_0,i}\cdot b_{m_0,i}}{R(i)}\right) \\
&= \left(1+\frac{b_{m_0,n}}{R(n)}\right)\cdot\prod_{i=0}^{d-1}(1-R_{m_0,i}) \qquad (12)
\end{aligned}
$$

We can show that

$$
\begin{aligned}
& 1+\frac{b_{m_0,n}}{R(n)} \\
(by(4)) \quad \geq \quad & 1+\frac{B_{m_0,n}}{\frac{B(n)}{d(n)+1}-MB(n)} \geq 1+\frac{B_{m_0,n}}{\frac{B(n)}{d(n)+1}-B_{m_0,n}} = \frac{\frac{B(n)}{d(n)+1}}{\frac{B(n)}{d(n)+1}-B_{m_0,n}} = \frac{1}{1-R_{m_0,n}}. \qquad (13)
\end{aligned}
$$

In addition, using (11) and Lemma 1, we have

$$
\begin{aligned}
\prod_{i=0}^{d-1}(1-R_{m_0,i}) \quad & \geq \quad 1-\sum_{i=0}^{d-1}R_{m_0,i} \\
(by(9)) \quad & > \quad 1-R_{m_0,n} \qquad (14)
\end{aligned}
$$

By combining (13) and (14) into (12), we get

$$
G_{FP} > \frac{1}{1-R_{m_0,n}}\cdot(1-R_{m_0,n}) = 1 \qquad (15)
$$

From the above, the system utility increases after the feeding. This contradicts with the assumption that assignment A is FC-optimal.

Lemma 1: Suppose $a_1, a_2, \cdots, a_n \geq 0$, and $a_1+a_2+\cdots+a_n = p < 1$. Let $f(a_1, a_2, \cdots, a_n) = (1-a_1)(1-a_2)\cdots(1-a_n)$. Then $f(a_1, a_2, \cdots, a_n) \geq 1-p$.

Proof: f is minimized when there is only one non-zero number, *i.e.* $a_i = p$ and $a_j = 0$ for $j \neq i$. To prove this, let's assume that there are two non-zero numbers, i.e. $a_i \neq 0$ and $a_j \neq 0$. We modify $a'_i = a_i + a_j$ and $a'_j = 0$, so $a_i + a_j = a'_i + a'_j$. However,

$$
(1-a'_i)(1-a'_j) = 1-a_i-a_j < 1-a_i-a_j+a_ia_j = (1-a_i)(1-a_j). \qquad (16)
$$

Thus, when there are more than one non-zero numbers in a_i, $1 \leq i \leq n$, we can modify the a_is to reduce f. This shows that f is minimized by having only one positive a_i, and $f = 1-p$. ■

B. Proof of Theorem 2

By Theorem 1, when all channels have bandwidth 1, we have $R(n) > \frac{|L(n)|}{d(n)+1} - 1$. Since $R(n)$ is an integer, this is equivalent to $R(n) \geq \left\lfloor \frac{|L(n)|}{d(n)+1} \right\rfloor$.

C. *Proof of Theorem 3*

We denote a user as "satisfied" if it is on or above its *poverty line*, otherwise "unsatisfied". First we prove a key property of the Poverty Line guided coordination (PGC):

Property 1: In a coordination iteration in which channels are fed to n, if a neighbor n_1 is converted from "satisfied" to "unsatisfied", then $PL(n) < PL(n_1)$.

Proof: First we assume node n is below its *poverty line* $PL(n)$. From the proof of Theorem 1, by combining (12), (13) and (15), we show that for channel m_0,

$$\frac{1}{1 - R_{m_0,n}} \cdot \prod_{i=0}^{d-1} (1 - R_{m_0,i}) > 1.$$

Under uniform channel bandwidth, this shows that before the bargaining,

$$\left(\frac{1}{1 - \frac{d+1}{|L(n)|}} \right) \cdot \prod_{i=0}^{d-1} \left(1 - \frac{1}{R(i)} \right) > 1.$$

Since $(1 - \frac{1}{R(i)}) < 1$ and $\frac{d+1}{|L(n)|} \leq \frac{1}{PL(n)}$, this implies that for any neighbor n_1, $0 \leq n_1 \leq d-1$ involved in the bargaining,

$$\frac{PL(n)}{PL(n) - 1} \cdot \frac{R(n_1) - 1}{R(n_1)} > 1$$

and therefore, $R(n_1) > PL(n)$. Hence, after feeding channel m_0 to n, no neighbor's throughput is reduced to below $PL(n)$. Hence, if the throughput of n_1 is reduced to below $PL(n_1)$, then it must be the case where $PL(n) < PL(n_1)$. ■

Proof for Theorem 3: From property 1, if the PGC can be conducted such that the user with lower *poverty line* among its (one-hop) neighborhood wins the priority to be feeded first, then each "satisfied" user will never become an "unsatisfied" user after it is feeded by the PGC. Hence, the number of iteration is at most N. If the user to be feeded is selected randomly among requestors, then a user with lower *poverty line* among its neighborhood will wait for an expected number of $O(N)$ iterations to win the opportunity, because at each iteration the number of users who request a PGC is bounded by N. The total number of iterations until the system reaches an equilibrium is $O(N^2)$. ■