Resource Allocation for Cognitive Radio Networks
Report2

By
Li ZHU Zheng LI
Caoxie ZHANG Yun LI

Contents

Chapter 1

Introduction

1.1 Cognitive Radio:A Extension of Software Radio

Cognitive radio(CR) refers to a paradigm in which either a network or a wireless node changes its transmission or reception parameters to communicate efficiently avoiding interference with licensed or unlicensed users. This intelligent radio is also viewed also as novel approach for improving the utilization of the radio electromagnetic spectrum. Before we delve into the content of CR, a glimpse of the origin of CR is presented here in order that a clear roadmap on how CR develops from previous technology which lays the foundation for it can be made. The key technology, which CR is based on, is Software-defined radio, sometimes shortened to software radio(SR). Joseph Mitola, who is internationally recognized as the "Godfather" of the software radio, coined the term in 1991 and he then promoted the term cognitive radio in 1998. SR is generally a multiband multimode radio that supports multiple air interfaces and protocols and is reconfigurable through software run on DSP or general-purpose microprocessors[1], which provides an ideal platform for the realization of cognitive radio. Build on SR, the goal of CR is to develop software agents that have such a high level of competence in radio domains that they may accurately be call "cognitive". In general, "cognitive radio is a particular extension of software radio that employs model-based reasoning about users, multimedia content, and communications context".[3]

1.2 Dynamic Spectrum Access in a Opportunistic Manner

While cognitive radio represents a much broader paradigm where many aspects of communication systems can be improved via cognition, in this report we mainly focus on a important application of CR – dynamic spectrum access.

textnormalThe idea of dynamic spectrum access is promoted in response to the limited available spectrum and inefficiency in spectrum usage now. According to Federal Communications Commission(FCC)[6], a large portion of the assigned spectrum is used sporadically and geographical variations in the utilization of assigned spectrum ranges from 15% to 85% with high variance in time. To deal with the underutilization of spectrum caused by a fixed spectrum policy, engineers, economists, and regulation communities are taking actions in searching for better spectrum management and techniques. In contrast to the current static spectrum management policy, the term dynamic spectrum access has broad connotations that encompass various approaches to spectrum reform. The diverse ideas presented at the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks(DySPAN) suggest the extent of this term and a discussion about the categorization of the dynamic spectrum access can be found in [5].
textnormalIn this report, we focus on the overlay approach under the hierarchical access model. The opportunistic manner will be adopted throughout. Since most of the spectrum is already assigned, the most important challenge is to share the licensed spectrum without interfering with the transmission of other licensed users.The cognitive radio enables the usage of temporally unused spectrum, which is referred to as itspectrum hole or itwhite space[18]. If this band is further used by a licensed user, the cognitive radio moves to another spectrum hole or stays in the same band, altering its transmission power level or modulation scheme to avoid interference.

1.3 A New Networking Paradigm

As mentioned above, the spectrum scarcity and inefficiency in its usage necessitates a new communication paradigm to exploit wireless spectrum opportunistically. From a networking perspective, this new model is referred to as NeXt Generation (xG) Networks as well as Dynamic Spectrum Access (DSA) and cognitive radio networks.[4]
textnormalThe concept of xG networks is actually bound up with the technology of cognitive radio and dynamic spectrum access. "Cognitive radio techniques provide the capability to use or share the spectrum in an opportunistic manner. Dynamic spectrum access techniques allow the cognitive radio to operate in the best available channel. More specifically, the cognitive radio technology will enable the users to (1) determine which portions of the spectrum is available and detect the presence of licensed users when a user operates in a licensed band (spectrum sensing), (2) select the best available channel (spectrum management), (3) coordinate access to this channel with other users (spectrum sharing), and (4) vacate the channel when a licensed user is detected (spectrum mobility)."[4]
In summary, the main functions for cognitive radios in cognitive networks can be summarized

as follows:[4]

- Spectrum sensing: Detecting unused spectrum and sharing the spectrum without harmful interference with other users.
- Spectrum management: Capturing the best available spectrum to meet user communication requirements.
- Spectrum mobility: Maintaining seamless communication requirements during the transition to better spectrum.
- Spectrum sharing: Providing the fair spectrum scheduling method among coexisting cognitive network users.

The main focus of this report is on the algorithm optimization of spectrum sharing, which is detailed in the following.

1.4 Spectrum Sharing

In cognitive networks, one of the main challenges in open spectrum usage is the spectrum sharing. In some respects, spectrum sharing can be regarded to be similar to generic medium access control (MAC) problems in existing systems. However, substantially different challenges exist for spectrum sharing in cognitive networks. "The coexistence with licensed users and the wide range of available spectrum are two of the main reasons for these unique challenges."[4]

Here we outline the discussion on spectrum sharing in [4]. We first enumerate the steps in spectrum sharing in cognitive networks. The spectrum sharing process consists of five major steps:[4]

1. Spectrum sensing: An cognitive network user can only allocate a portion of the spectrum if that portion is not used by an unlicensed user. Accordingly, when an cognitive network node aims to transmit packets, it first needs to be aware of the spectrum usage around its vicinity.
2. Spectrum allocation: Based on the spectrum availability, the node can then allocate a channel. This allocation not only depends on spectrum availability, but it is also determined based on internal (and possibly external) policies. Hence, the design of a spectrum allocation policy to improve the performance of a node is an important research topic.

3. Spectrum access: In this step, another major problem of spectrum sharing comes into picture. Since there may be multiple cognitive network nodes trying to access the spectrum, this access should also be coordinated in order to prevent multiple users colliding in overlapping portions of the spectrum.

4. Transmitter-receiver handshake: Once a portion of the spectrum is determined for communication, the receiver of this communication should also be indicated about the selected spectrum. Hence, a transmitter-receiver handshake protocol is essential for efficient communication in cognitive networks. Note that the term ithandshake by no means restricts this protocol between the transmitter and the receiver. A third party such as a centralized station can also be involved.

5. Spectrum mobility: cognitive network nodes are regarded as "visitors" to the spectrum they allocate. Hence, if the specific portion of the spectrum in use is required by a licensed user, the communication needs to be continued in another vacant portion. As a result, spectrum mobility is also important for successful communication between cognitive network nodes.

In addition, a classification of spectrum sharing techniques and the fundamental results about these techniques is given in [5]. Considering the tradeoff between system complexity and performance, hybrid techniques may be considered for the spectrum technique. In this report, we discuss a algorithm that is distributed, cooperative and overlay, namely in opportunistic manner.

Chapter 2

Related Work

2.1 Opportunistic Spectrum Access(OSA)

Motivated by the conflict between finite spectrum resources and increasing number of wireless devices, open spectrum policy is employed which enables secondary users to share under-utilized spectrums with primary users (legacy users) opportunistically [7],[8],[9],[10].
[7], [8] focus on the algorithm design in Opportunistic Spectrum Access (OSA) assuming that each secondary users have full knowledge of the availability of all channels. In the context of open spectrum, the primary goal is to maximize utilization and provide fairness among different devices [7], [8] and the main problem lies in dealing with the fluctuation in spectrum availability (spectrum heterogeneity) and avoiding interference with primary users, which calls for coordination between users. Considering the case where the collection of spectrums forms a spectrum pool, algorithms can be designed to find an appropriate distribution among secondary users while minimizing interference. In a slow varying scenario where user location topology and available spectrum remains unchanged during the allocation, by modeling with graph theory and describing three utility functions, the spectrum allocation problem is reduced to a graph-coloring problem and proved to be PN-hard. As the centralized algorithm requires a central allocation server which is almost impossible to implement, a distributed version of the algorithm is built to fulfill the approximation. [7] While, in a mobile environment where numerous users keeps changing in position, this topology-optimized allocation algorithm requires huge amount of computation because the network has to perform global reassignment after any change in topology in order to maintain spectrum utilization and fairness among users. Actually, prior information can be obtained from previous spectrum assignment and used in distributed algorithms which further reduce the workload to adapt to topology change. A local bargain framework is introduced by [8] where users make self-organization into bargain groups and these groups make spectrum assignment independently to reach an approximation of optimal solution. Simulation

validates this approach in maximizing the fairness-based spectrum utilization but with less complexity. [9], [10] focus on the physical layer and media access control layer (MAC) of OSA technology. Typically, OSA includes spectrum sensor at physical layer, sensing policy at MAC layer and access policy at MAC layer. [10] Considering the power-consuming nature of full spectrum sensing, it is unrealistic for battery-powered wireless nodes to perform full spectrum sensing. So we could only optimize our design based on the assumption that every secondary user only has access to a subset of the full spectrum. Keeping the interference perceived by primary users under a certain threshold, [9] proposes an analytical framework for OSA based on the theory of partially observed Markov decision process (POMDP). The solution to optimal POMPDA has exponential complexity, so a suboptimal greedy approach to POMDP is then proposed as a tradeoff to reduce complexity to linear level. It is proved that the design of spectrum sensor and access policy can be decoupled from that of sensing policy without losing optimality. [10] Based on this, the joint OSA design can be formulated as an unconstraint POMDP which leads to insight of the best tradeoff between false alarm and miss detection.

2.2 Power allocation

In [13],Michael J. Neely developed a dynamic control strategy for minimizing energy expenditure in a time varying wireless network with adaptive transmission rates. The algorithm operates without knowledge of traffic rates or channel statistics, and yields average power that is arbitrarily close to the minimum possible value achieved by an algorithm optimized with complete knowledge of future events. Proximity to this optimal solution is shown to be inversely proportional to network delay. Neely then presented a similar algorithm that solves the related problem of maximizing network throughput subject to peak and average power constraints. The techniques used by Neely are novel and establish a foundation for stochastic network optimization. And in [11], Neely and Modiano made the formulation of a general power control problem for time-varying wireless networks, the characterization of the network layer capacity region, and the development of capacity achieving routing and power allocation algorithms that offer delay guarantees and consider the full effects of queueing.

2.3 Utility optimization

Modern data networks consist of a variety of heterogeneous components, and continue to grow as new applications are developed and new technologies are integrated into the existing communication infrastructure. While network resources are expanding, the demand for these resources is also expanding, and it is often the case that data links are loaded with more traffic than they were designed to handle. In order to provide high speed connectivity for future personal computers, hardware devices, wireless units, and sensor systems, it is essential to develop fair networking techniques that take full advantage of all resources and system capabilities. Michael J. Neely, Eytan Modiano and Chih-Ping Li designed a set of decoupled algorithms for resource allocation, routing, and flow control for general networks with both wireless and wireline data links and time varying channels in [12]. And they have presented a fundamental approach to stochastic network control for heterogeneous data networks. Simple strategies were developed that perform arbitrarily close to the optimally fair throughput point, with a corresponding tradeoff in end-to-end network delay. The strategies involve resource allocation and routing decisions that are decoupled over the independent portions of the network, and flow control algorithms that decoupled over dependent control valves at every node. Such theory-driven networking strategies will impact the design and operation of future data networks.

2.4 Application of Maximum Weighted Matching (MWM)

The resource allocation problem can be reduced to Maximum Weighted Matching (MWM) if secondary users transmit on the channel without interference on other channels, which is the case of orthogonal channels for secondary users[14]. A matching is to link two groups of nodes and no two links share the same node. A weigh of a matching is the sum of all the weight of the links belonging to the matching. MWM is to find the maximal weigh of a matching, with an $O(N^3)$ complexity algorithm found in presence[15]. Recent works [16],[17] have investigated Greedy Maximal Match Scheduling(GMS) to achieve near optimal results in a much simpler implementation. GMS firstly try to find the largest weight in the available links and remove all the links that have same nodes as in the first link. It then starts to find the largest weight of link in the remaining links. Same procedure is continued until no link is left. GMS algorithm has an O (LlogL) complexity with low overhead and the total weight is at least 1/2 of the weight of the MWM [16].

Chapter 3

Models and Algorithms on Cognitive Network

3.1 A Graph-theoretical Model to Characterize Opportunistic Spectrum Access in Cognitive Network

We present this theoretical model defined in [19],[21] to represent the general allocation problem, and describe three utility functions that trade off spectrum utilization and fairness are described. We then show that this optimal allocation problem can be reduced to a variant of a graph-coloring problem.

3.1.1 Definitions in the model

- In a network waiting for spectrum assignment, there are N users or entities indexed from 0 to $N-1$ competing for M spectrum bands indexed 0 to $M-1$.
- *Channel availability*: $L = l_{n,m} | l_{n,m} \in 0, 1_{N\times M}$ is a N by M binarymatrix representing the channel availability: $l_{n,m} = 1$ if and only if channel m is available at user n.
- *Channel reward*: $B = b_{n,m_{N\times M}}$, a N by M matrix representing the channel reward: $b_{n,m}$ represents the maximum bandwidth/throughput that can be acquired (assuming no interference interference from neighbors) by user n using channel m.
- *Interference constraint*: Let $C = c_{n,k,m} | c_{n,k,m} \in 0, 1_{N\times N\times M}$, a N by N by M matrix, represents the interference constraints among secondary users. If $c_{n,k,m} = 1$, users n and k would interfere with each other if they use channel m simultaneously. The constraint depends on channel availability, i.e., $c_{n,k,m} \leq l_{n,m} \times l_{k,m}$ and $c_{n,n,m} = 1 - .l_{n,m}$. Again,

this constraint is channel specific: two users might be constrained on one channel but not another.

- *Conflict free channel assignment*: $A = a_{n,m} | a_{n,m} \in 0, 1, a_{n,m} \leq l_{n,m}{}_{N \times M}$ is a N by M binary matrix that represents the assignment: $a_{n,m} = 1$ if channel m is assigned to user n. A conflict free assignment needs to satisfy all the interference constraints defined by C, that is,

$$a_{n,m} + a_{k,m} \leqslant 1, if c_{n,k,m} = 1, \forall n, k < N, m < M. \tag{3.1}$$

Let $\Lambda(L, C)_{N,M}$ denote the set of conflict free spectrum assignments for a given set of N users and M spectrum bands and constraints C.

3.1.2 Utility Functions for Optimization

- *User reward*: $\mathfrak{R} = \beta_n = \sum_{m=0}^{M-1} a_{n,m} \cdot b_{n,m}{}_{N1}$ represents the reward vector that each user gets for a given channel assignment.
- *Network utilization*: The channel allocation is to maximize network utilization $U(\mathfrak{R})$.

Given the model above, this spectrum assignment problem is equivalent to a optimization problem. In general, the optimization function is as follows:

$$A^* = argmax_{A \in \Lambda(L,C)_{N,M}} U(\mathfrak{R}). \tag{3.2}$$

Specific utility functions based on traffic patterns and fairness inside the network are presented as follows:

- *Max-Sum-Reward*: This maximizes the total spectrum utilization in the system regardless of fairness. The optimization problem is expressed as:

$$U_{sum} = \sum_{n=0}^{N-1} \beta_n = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} a_{n,m} \cdot b_{n,m}. \tag{3.3}$$

- *Max-Min-Reward*: This maximizes the spectrum utilization at the bottleneck user, or the user with the least allotted spectrum. The optimization problem is expressed as:

$$U_{\min} = \min_{0 \leqslant n \leqslant N} \beta_n = \min_{0 n < N} \sum_{m=0}^{M-1} a_{n,m} \cdot b_{n,m}. \tag{3.4}$$

Roughly, Max-Min-Reward driven allocation gives the most poorly treated user (i.e. the user who receives the lowest reward) the largest possible share, while not wasting any network resources. This is the simplest notion of fairness.

- *Max-Proportional-Fair*: In [19],[21] and the reference therein, the fairness for singlehop flows is considered. The corresponding fairness-driven utility optimization problem is expressed as:

$$U_{fair} = \sum_{n=0}^{N-1} \log(\beta_n) = \sum_{n=0}^{N-1}) \log(\sum_{m=0}^{M-1} a_{n,m} \cdot b_{n,m}). \tag{3.5}$$

3.1.3 Color-sensitive Graph Coloring

To solving this complex optimization problem, [19], [21] reduce it to a variant of the graph coloring problem by mapping spectrum channels into colors, and assigning them to users (vertices in a graph).

We present a bidirectional graph $G = (V, L, E)$ defined in [19], [21], where V is a set of vertices denoting the users that share the spectrum, L is the available spectrum or the color list at each vertex; defined in subsection 3.1.1, and E is a set of undirected edges between vertices representing interference between any two vertices. For any two vertices $u, v \in V$, a m-colored edge exists between u and v if $c_{u,v,m} = 1$. The edges depend on the interference constraint C (see subsection 3.1.1).

The spectrum allocation problem is equivalent to coloring each vertex using a number of colors from its color list to maximize system utility. The coloring scheme is constrained by that if a m colored edge exists between any two vertices, they cannot simultaneously use color m. This problem is called *color-sensitive graph coloring* (CSGC).

3.1.4 Spectrum Allocation Algorithm

The optimal coloring problem is known to be NP-hard[19], [21]. In this subsection, we discuss a set of heuristic based approaches given in [19], [21] that produce good coloring solutions.

In this work, the heterogeneity in both the color list and also the color rewards (bandwidth, throughput) are considered. The colors are assigned in a greedy fashion[19]. In each stage, the algorithm labels all the vertices with a non-empty color list according to a labeling rule. Each label is associated with a color. The algorithm picks the vertex with the highest label, and assigns the color associated with the label, e.g. color m. The algorithm then deletes the color from the vertexs color list, and also from the color lists of the m color-constrained neighbors. It should be noted that the neighborhood of a vertex keeps on changing as other vertices are processed. The labels of the colored vertex and his neighbor vertices are modified according to the new graph. The algorithm enters the next stage until every vertexs color list becomes empty. Intuitively, this algorithm chooses to color the most valuable vertices first, i.e. the vertices that contribute to the system utility the most.

In the following, we examine a set of heuristics based labeling rules that are proposed in

[19], [21]. We claim that a rule is collaborative if it considers the impact of interference to the neighbors when performing labeling and coloring.

- *Collaborative-Max-Sum-Bandwidth (CMSB) rule*
  This rule aims to maximize the sum of bandwidth weighted color usage, corresponding to U_{sum} optimization defined in (3.3). When a vertex n is assigned with a color m, his contribution to the sum bandwidth in a local neighborhood can be computed as $b_{n,m}/D_{n,m}$ since his neighbors can not use the color. Here $D_{n,m}$ represents the number of m color constrained neighbor of a vertex n in the current graph. In [19], [21], the rule to label the vertex is:

$$l_n = \max_{m \in l_n} b_{n,m}/(D_{n,m}+1), \tag{3.6}$$

$$color_n = \arg\max_{m \in l_n} b_{n,m}/(D_{n,m}+1). \tag{3.7}$$

  where l_n represents the color list available at vertex n at this assignment stage. This rule considers the tradeoff between spectrum utilization (in terms of selecting the color with the largest bandwidth) and interference to neighbors. This rule enables collaboration by taking into account the impact to neighbors. If two vertices have the same label, then the vertex with lower assigned bandwidth weighted colors will get a higher label.[19], [21]

- *Non-collaborative-Max-Sum-Bandwidth (NMSB) rule*
  This rule aims to maximize the sum of bandwidth weighted color usage without considering the impact of interference to neighbors. The vertex with the maximum bandwidth-weighted color will be colored, i.e. a vertex n is labeled with

$$label_n = \max_{m \in l_n} b_{n,m}, \tag{3.8}$$

$$color_n = \arg\max_{m \in l_n} b_{n,m}. \tag{3.9}$$

  When colors have the same property, this corresponds to a random labeling. Comparing to CMSB rule, this rule is relatively selfish or non-collaborative.[19], [21]

- *Collaborative-Max-Min-Bandwidth (CMMB) rule*
  This rule aims to assign equal number of colors to vertices in order to improve the minimum bandwidth weighted colors that a vertex can get, while considering interference to neighbors. It is targeted to solve MMB optimization defined in (3.4). In each stage, the vertices are labeled according to

$$label_n = -\sum_{m=0}^{N-1} a_{n,m} \cdot b_{n,m}, \tag{3.10}$$

$$color_n = \arg\max_{m \in l_n} b_{n,m}/(D_{n,m}+1). \tag{3.11}$$

If two vertices have the same label, then the vertex with larger $\max_{m \in l_n} b_{n,m}/(D_{n,m}+1)$ value gets a higher label.[19], [21]

- *Non-collaborative-Max-Min-Bandwidth (NMMB) rule*
  This rule is a non-collaborative version of CMMB rule where the impact of interference is not considered in the vertex labeling, and coloring. In each stage, the vertices are still labeled according to (3.10), but the associated color is determined as $\arg\max_{m \in l_n} b_{n,m}$. If two vertices have the same label, then the vertex with larger $\max_{m \in l_n} b_{n,m}$ is assigned with a higher label.[19], [21]

- *Collaborative-Max-Proportional-Fair (CMPF) rule*
  This rule aims to achieve a specific fairness among vertices, corresponding to MPF optimization defined in (3.5). It is well known that proportional fair scheduling [19], [21] assigns resource (time slot) to the user with the highest r_n/R_n, where r_n represents the reward generated by using the resource and R_n is the average reward that the user n gets in the past. The concept of proportional fair scheduling is applied to this problem by viewing color as time slot. In each stage, the vertices are labeled according to

$$label_n = \frac{\max_{m \in l_n} b_{n,m}/(D_{n,m}+1)}{\sum_{m=0}^{M-1} a_{n,m} \cdot b_{n,m}}, \tag{3.12}$$

$$color_n = \arg\max_{m \in l_n} b_{n,m}/(D_{n,m}+1). \tag{3.13}$$

  where $label_n$ represents the ratio of the interferenceweighted bandwidth using one color and the accumulated bandwidth in the past. This rule is in general different from the traditional proportional fair rule as it captures the difference in the impact of interference generated by a color (resource) assignment.[19], [21]

- *Non-collaborative-Max-Proportional-Fair (NMPF) rule*
  This is a non-collaborative version of the CMF rule. Each vertex n is labeled according to

$$label_n = \frac{\max_{m \in l_n} b_{n,m}}{\sum_{m=0}^{M-1} a_{n,m} \cdot b_{n,m}}, \tag{3.14}$$

$$color_n = \arg\max_{m \in l_n} b_{n,m}. \tag{3.15}$$

- *Random (RAND) rule* Each vertex is assigned with a random label, and the chosen vertex is colored with a randomly picked color from his color list. [19], [21]

The implementation of the above coloring algorithm can be divided into two categories.[19], [21]

- *Centralized:* If there is a central controller who makes decisions on color assignment, the corresponding implementation is quite straightforward. The controller collects spectrum and interference information from all the vertices, and executes the rule to distribute colors among vertices and broadcast the assignment.

- *Distributed:* In this case, each vertex executes the rule to select the appropriate color(s). The colors are assigned in a greedy fashion. In each stage, each vertex labels itself according to one of the above labeling rules, and broadcasts the label to his neighbors. A vertex with the maximum label within his neighborhood gets to grab the color associated with his label and broadcasts the color assignment to his neighbors. After collecting assignment information from surrounding neighbors, each vertex updates his color list and recalculates the label. This process is repeated until the color list at each vertex is exhausted or all the vertices are satisfied.

3.2 A Game Theory Model to Characterize Spectrum Sharing in Cognitive Network

We present this theoretical model defined in [22] to solve the spectrum sharing problem. Spectrum sharing is an inherently distributed problem, with no central authority to coordinate and arbitrate channel allocation. It is important that spectrum sharing be efficient, allowing as many users as possible to use the network. With this in mind, we have modeled spectrum sharing as a game between providers, and analyzed the price of anarchy. We view the channel assignment problem as a game, where the players are the service providers and APs are acquired sequentially. And it is shown that if we assume that providers are able to use easily implementable bargaining procedures, the price of anarchy is bounded by a constant if users are distributed uniformly or every AP uses the same transmission power.

3.2.1 Definitions in the model

- *Game Theory*:Game theory is a branch of applied mathematics that is used in the social sciences (most notably economics), biology, political science, computer science and philosophy. Game theory attempts to mathematically capture behavior in strategic situations, in which an individual's success in making choices depends on the choices of others. While initially developed to analyze competitions in which one individual does better at another's expense (zero sum games), it has been expanded to treat a wide class of interactions, which are classified according to several criteria.

- *Price Of Anarchy*:the radio between the total coverage of the APs in the worst Nash

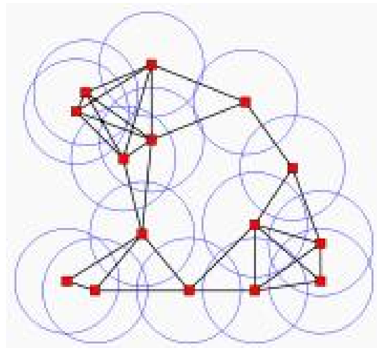

Figure 3.1: Potential interference between two APs

equilibrium of the game and what the total coverage of the APs would be if the channel assignment were done by a central authority. It shows how far a Nash equilibrium can be from the socially optimal solution to the problem.

- *Nash Equilibrium*: In game theory, the Nash equilibrium is a solution concept of a game involving two or more players, in which no player has anything to gain by changing only his or her own strategy unilaterally. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute a Nash equilibrium.

- *Unit Disk Graph*: In geometric graph theory, a unit disk graph is the intersection graph of a family of unit circles in the Euclidean plane. That is, we form a vertex for each circle, and connect two vertices by an edge whenever the corresponding circles cross each other.

3.2.2 Interference graph induced by the game

The small circle, denoted $R_t(u)$, represents u's transmission range. All messages sent by u can be correctly received by users in $R_t(u)$. The larger circle, denoted $R_s(u)$, represents u's sensing range. In practice, the radius of $R_s(u)$ is about twice that of $R_t(u)$. The actual size of $R_s(u)$ and $R_t(u)$ depends on the transmission power used by u. AP u's within $R_s(u)$ if they share the same channel. To avoid such interference, the distance d(u,v) between u and v has to be greater than $R_t(u)$+$R_t(v)$+max$\{R_s(u)$,$R_s(v)\}$. That is, if APs u and v are greater than $R_t(u)$+$R_t(v)$+max$\{R_s(u)$,$R_s(v)\}$apart, they can transmit using the same channel, since then no user in v's transmission range will be able to sense a message from u or its users, and vice versa.

We can represent the game using a labeled Graph G=(V,E), where the vertices in V are

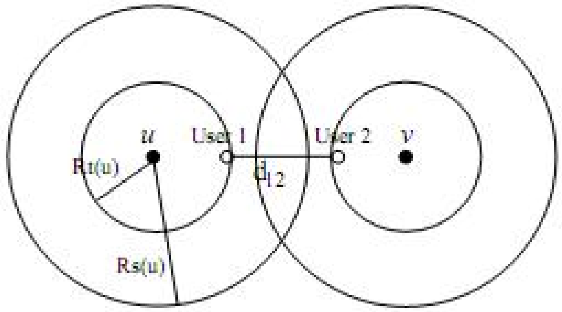

Figure 3.2: A collection of unit circles and the corresponding unit disk graph

the APs, and two vertices u and v are joined by an edge if they potentially interfere, i.e., if $d(u, v) \leq R_t(u)+R_t(v)+\max\{R_s(u),R_s(v)\}$. Each vertex also has a label, which represents the utility of the AP associated with that vertex being assigned a channel. G is called the *interference graph* induced by the game.

3.2.3 k-coloring of the graph

An association of some vertices to colors such that two adjacent edges are labeled with different colors. Clearly this corresponds to a feasible assignment of channels to APs. All APs that are assigned colors can safely communicate on the channel associated to the color without interference. The APs that are assigned a channel in a Nash equilibrium of the game correspond to a maximal subset of vertices that has been colored with k colors. A maximal k-colored subset of the induced graph is defined to be a subset of nodes with specific coloring such that no additional nodes can be colored. If there are any other vertices in the graph that can be colored, then the corresponding AP should have been assigned a channel. Conversely, given a maximal k-colored subset of the interference graph, there is a Nash equilibrium of the game where these are precisely the APs that are assigned a channel. In particular, this will be the case if the APs in the maximal set are set up before any other APs are set up. Thus there is a 1-1 correspondence between maximal k-colored subsets of the graph and Nash equilibria of the game. Moreover, a socially optimal assignment corresponds to a maximal k-colored subset of maximum weight. Thus, the price of anarchy is simply the ratio of the total weight of a maximal k-coloring of minimum weight to the k-coloring of maximum weight.

3.2.4 Spectrum Sharing Games Model

We model the channel assignment problem as a game, where the players are the service providers, APs are set up or acquired by service providers sequentially. When an AP is set up or acquired, a channel must be chosen that does not interfere with the channels chosen for APs that were previously set up; if there is no such channel, the AP cannot be used. The order that the APs are set up is determined exogenously and is arbitrary. We assume that when a service provider sets up an AP, it knows about the APs that have already been set up and might interfere with it, but we do not make any assumptions about what the service providers know about other APs that have already been set up. For simplicity, we also assume that when a service provider sets up and AP, it does not know what APs will become available in the future. The only information it has is the Aps that currently exit.

The utilities of the service providers depend on how many users they can serve. We assume that there is a commonly known distribution of users. The utility to a service provider of setting up an AP u that is assigned a channel is the expected number of users in $R_t(u)$; if AP u is not assigned a channel, then its utility to the service provider is 0. The utility of a provider at the end of a game is just the sum of the utilities of the APs that it sets up.

In some special cases players converge to a Nash equilibrium after polynomial number of steps. But in the general case, we show that there exists an exponentially long path of improvements to a Nash equilibrium.

3.2.5 Two Kind of Bargains In The Spectrum Sharing Games

- The first is a generalization of the situation described initially with APs v_1,v_2, and v_3. If v_2 is colored (i.e., has a channel assigned), v_1 and v_3 are not, v_1 and v_3 could be colored if v_2 were not colored, and the sum of the weights of v_1 and v_3 is greater than the weight of v_2, then we assume that the providers that own APs v_1 and v_3 can always offer the owner of v_2 sufficient utility, so that v_2 is uncolored, while still themselves coming out ahead. We do not go into the details of exactly what the offers are. All that matters is that, in equilibrium, the exchange will be made. We call this a *local 2-buyer-1-seller bargain.*

- The second occurs if an AP is uncolored but its weight is greater than the sum of weights of all its neighbors of a particular color. In this case, we assume that the owner of that AP can offer the owners of the interfering APs sufficient utility so that the interfering APs will be uncolored. Again, we do not go into the details of exactly what the offers are. We call this *local 1-buyer-multiple-seller bargain.* Note that although many sellers may be involved, this really is a collection of 2-way arrangement, since the buyer can negotiate separately with each of the sellers.

3.2.6 some theorems and propositions of PoA

In [22], the author proved some theorems and propositions of price of anarchy(PoA).

- *Suppose the price of anarchy if there is only one channel for a spectrum-sharing game that allows a certain type of bargaining is ρ. Then, for all k, the price of anarchy for the same game with k channels is at most $\rho + max(0, 1 - \rho/k)$ and at least ρ.*

- *The price of anarchy is unbounded in the basic spectrum-sharing game, no matter how many channels or players there are, even if all vertices have equal weight.*

- *If all APs transmit with the same power and users are uniformly distributed, then the price of anarchy of the spectrum-sharing game is at most 5+max(0,1-5/k) and at least 5.*

- *If all APs transmit with the same power, users are uniformly distributed, and 2 buyer-1 seller bargains are allowed, then the price of anarchy of the spectrum-sharing game is at most 3+max(0,1-3/k) and least 3.*

- *If APs transmit with the same power but user may not be uniformly distributed, then the price of anarchy is unbounded unless bargains involve at least min(p, τ) sellers, where p is the number of players and the interference graph is $(\tau + 1)$-claw free.*

- *If APs transmit with the same power and 1-buyer-multiple-seller bargains are allowed, then the price of anarchy of the spectrum-sharing game is at most 5+max(0,1-5/k) and at least 5.*

- *In the general case(where APs transmit with different powers and users are not uniformly distributed), then the price of anarchy of the spectrum sharing game is unbounded, even if multiple-buyer-multiple-seller bargains are allowed.*

- *Suppose that distances have been normalized so that, for any pair of nodes u, v, we have $R_t(u)$+$R_t(v)$+max$\{R_s(u)$,$R_s(v)\}\leq$1. Thus, two vertices u, v such that $d(u, v) > 1$ do not have an edge between them in the interference graph. Further suppose that bargains involving arbitrary sets of vertices within distance $\sqrt{2}d$ are allowed. Then the price of anarchy in the spectrum-sharing game is at most $d^2/(d-1)^2$.*

- *Even if users are distributed uniformly, in the spectrum-sharing game with power control, the price of anarchy is unbounded unless bargains involve at least min(p, τ) sellers, where p is the number of players and the interference graph is $(\tau + 1)$-claw free.*

- *If users are distributed uniformly and 1-buyer-multiple seller bargains are allowed, then the price of anarchy of the spectrum-sharing game with power control is at most 9 and at least 7-ϵ, for any $\epsilon > 0$.*

3.2.7 Convergence to Nash equilibria

In the weighted spectrum sharing game, players will converge to a local optimum after finitely many local improvements, no matter what kind of bargains are allowed. Furthermore, if all weights are integers bounded by a polynomial in the number of vertices, then players will converge to a local optimum after a polynomial number of local improvements.

Suppose that local improvements are of two kinds: coloring a new vertex and changing the coloring via a 1-buyer-multiple-seller bargain. In the weighted spectrum sharing game on unit disk graphs, it may take exponentially many local improvements to converge to a Nash equilibrium.

3.3 Network Model for opportunistic scheduling with reliability guarantees in Cognitive Radio

This model presented here aims at maximizing the throughput utility of the secondary users while considering the collision constraint with primary users and the interference with other secondary users(channels may not be orthogonal to secondary users). In order to achieve this, adaptive queuing and Lyapunov Optimization are employed to design an online control, scheduling and resource allocation algorithm for cognitive radio.

3.3.1 Underlying Assumptions of this Model

The network we talk about here is a time-slotted model consisting of M primary users and N secondary users as shown below. Primary users are assumed to be static, while secondary users can move around and the channels 'visible' to them also keep changing. But we assume that the topological pattern of the network remains the same during one time slot. Exactly one unique channel is assigned to every licensed user. And all these channels are orthogonal to each other. In order to make things simple and clear, exactly one packet can be transmitted over any channel during a time slot. We also make the assumption that channel state information and channel accessibility of secondary users are Markovian process. Finally, the network here is a distributed one which means that no user knows a whole picture of the network.

3.3.2 Aim of this model

Throughput under the constraint of collision and interference is evidently the ultimate goal of cognitive radio designing. Let R_n be the number of new packets admitted into this queue

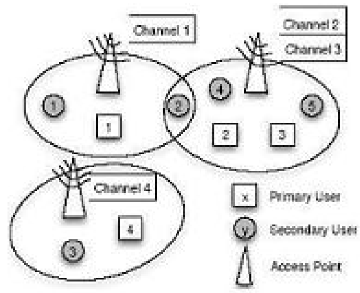

Figure 3.3: Network structure of the cognitive network,cited in[14]

in slot t. Let r_n denote the time average rate of admitted data for secondary user n that means:

$$r_n = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} R_n(\tau) \tag{3.16}$$

Let $\mathbf{r}$=(r_1,...,rN) denote the vector of these time average rates of these N secondary users. Under a specific but common situation, the weight of these N secondary users are the same, so the throughput should be defined as $\frac{1}{N}\sum r_n$. But for a more general purpose, let $\{\theta_1,...,\theta_n\}$ be a collection of positive weights for N secondary users, then the aim of the model is to design a flow control and scheduling policy that yields a $\mathbf{r}$ that maximize $\sum_{n=1}^{N}\theta_n r_n$ while subject to some constraints.

3.3.3 Important Definition and Variables

- Channel accessibility matrix
  $\mathbf{H}$(t)=$\{h_{nm}\}_{N\times M}$
  Where:
  $$h_{nm}(t) = \begin{cases} 1 & \text{if sec. user n can access channel m in slot t} \\ 0 & \text{else} \end{cases} \tag{3.17}$$
  As we have mentioned above, the $\mathbf{H}$(t) process is Markovian and has a well defined steady state distribution.

- Channel occupancy
  Let $\mathbf{S}$(t)=(S_1(t),S_2(t),...,S_M(t)) represent the current primary user occupancy state of the M channels. $\mathbf{S}_i$(t)=0 if channel i is occupied by primary user i in time slot t and $\mathbf{S}_i$(t)=1 if i is idle in time slot t. Because we only have two states(occupied or idle) over a channel and the number of primary user is finite, $\mathbf{S}$(t) evolves according to a

finite state ergodic Markov chain on the space $\{0,1\}^M$. Due to some limitation in carrier sensing, the exact channel state may nit be available to the secondary users. The channel state available to secondary user is described by a probability vector $\mathbf{P}$(t) discussed below.

- Channel state probability vector.
  $\mathbf{P}$(t)=($P_1(t)$,P_2,...,P_M) where P_i is the probability that channel i is idle in time slot t. This vector can be obtained through a knowledge of the traffic statistics of primary users. The statistic nature of $\mathbf{P}(t)$ leads to the inherent sensing measurement errors that no primary transmission detection algorithm could solve. As collision is inevitable, our goal is to constrain it under a pre-given constant ρ_m.

- Channel set of interference
  These M channels mentioned above may not ne orthogonal to secondary uses, so variables are needed to characterize the interference between secondary users. We define $\mathcal{I}_{nm}$ as the set of channels that secondary user n interferes with when it uses channel m. A indicator variable is further defined as:
  $$\mathcal{I}_{nm} = \begin{cases} 1 & \text{if k} \in \mathcal{I}_{nm} \\ 0 & \text{else} \end{cases} \tag{3.18}$$
  Clearly, if there is no interference between secondary users, then $\mathcal{I}_{nm} = \{m\} \forall n$.

- Data receiving process
  Each secondary user n receives data according to an i.i.d arrival process A_n(t) which is upper bounded by a constant value A_{max} that has rate λ_n packet/slot. We will show later that this A_{max} guarantees the worst performance of this model which is very important in practical scenario.

- Backlog **queue** in network layer
  U_n is defined as the backlog queue of secondary user n at the **beginning** of time slot t.

- Virtual collision **queue**
  We define $X_m(t)$ to track the amount by which the number of collisions suffered by a primary user m exceeds its time average collision constraint rate ρ_m.

- New packets admitted
  R_n is the number of new packets admitted into this queue in slot t.

- Number of attempted transmission
  Let $\mu_{nm}(t)$ be the number of attempted packet transmission when a control action allocates channel m to n.

- Collision variable

$$C_m(t) = \begin{cases} 1 & \text{if there was a collison with the primary user in channel m in time slot t} \\ 0 & \text{else} \end{cases} \tag{3.19}$$

Let $c_m(t) = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} C_m(\tau)$

- Control variable
Control variable V offered by the algorithm that we will discuss later enables an explicit trade-off between the average throughput utility and delay.

3.3.4 Modeling the Network with Queuing Dynamics

There are two kinds of queues involved in this model. One is the backlog queue in network layer of secondary users as we described above and the other is the virtual collision queue which is maintained in software.

- The queuing dynamics of the secondary user n is described by:

$$U_n(t+1) = max[U_n(t) - \sum_{m=1}^{M} \mu_{nm}(t) S_m, 0] + R_n(t) \tag{3.20}$$

Which means that the backlog at the beginning of time slot $t+1$ equals to the remaining backlog of time slot t plus the number of new packets admitted in the queue during time slot t. And the constraints are:

  - Constraint on transmission rate: $\mu_{nm}(t) \in 0, 1 \forall m, n$
  - Idle channel: $\mu_{nm}(t) \leq h_{nm}(t)\ \forall m, n$
  - Allocation constraint: $0 \leq \sum_{m=1}^{M} \mu_{nm}(t) \leq 1\ \forall n$
  - Successful transmission: $\mu_{nm}(t) = 1 \Longleftrightarrow \sum_{j=1}^{M} \sum_{i=1, i\neq n}^{N} \mathcal{I}_{ij}^{m} mu_{ij}(t) = 0 \forall m, n$
  - Data rate constraint: $0 \leq R_n(t) \leq A_n(t)$

When the channels are orthogonal for secondary users, these constraints simplifies to $0 \leq \sum_{n=1}^{N} \mu_{nm}(t) \leq 1$.

- The queuing dynamics for virtual collision queue $X_m(t)$ is:

$$X_m(t+1) = max[X_m(t) - \rho - m, 0] + C_m(t) \tag{3.21}$$

The whole system is rate stable only when $c_m \leq \rho_m$, but the value of queuing dynamics lies in that we can turn the time average constraint into queuing problems if our flow control and resource allocation policies to stabilize all collision queue.

3.3.5 An online algorithm to achieve maximized throughput

This algorithm is a cross-layer strategy which contains two aspects.

- Flow control:
  We aim at minimizing $R_n(t)[U_n(t) - V\theta_n]$ under the constraint of $0 \leq R_n(t) \leq A_n(t)$. We can easily affect the performance/delay tradeoff by changing the parameter V.

- Resource allocation:
  We choose an allocation that maximize $\sum_{n,m} \mu_{nm}(t)[U_n(t)P_m(t) - \sum_{k=1}^{M} X_k(t)(1 - P_k(t))I_{nm}^k]$ This is the difference between the current queue backlog $U_n(t)$ weighted by the probability that primary user m is idle and the weighted sum of all collision queue backlog for the channels that user n interferes with if it uses channel m.

The two maximization requires solving the Maximum Weight Match(MWM) problem on an N×M bipartite graph of N secondary users and M channels which is presented in [14]

3.4 Cognitive Network Control Algorithm(CNC)

Cognitive Network Control Algorithm(CNC) is the methods created on the basis of the 3.3.

3.4.1 Implementation of Cognitive Network Control Algorithm(CNC)

We focus on the the orthogonal channel case in which secondary users transmission on one channel would not interfere with other channels. In this case, the problem is reduced to a Maximal Weight Match(MWM) on a $N \times M$ bipartite graph between N secondary users and M primary users. An edge would exist between nodes n and m if secondary users n can access the channel m in slot t. The topology can be seen in Figure 3.4,and the weight is given by $U_n(t)P_m(t) - X_m(t)(1 - P_m(t))$. The algorithm is to find a match that will maximize

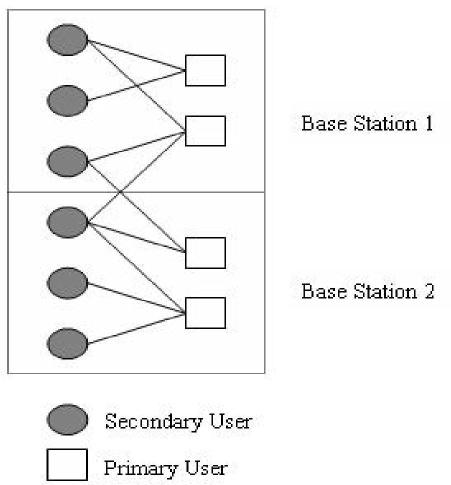

Figure 3.4: Resource allocation can be reduced to maximum weighted matching at the case of orthogonal channels

the sum weights of the match. While the MWM problem can be solved in polynomial time, the complexity of implementation is great. Here we provide another simple idea of Greedy Maximal Match(GMM), which greedily chooses the edge of largest weight at each step. The GMM has the advantage of fast implementation and the obtained result can be as least half as better as the optimal result. The GMM has the property below:

$$\sum_{n,m} \mu_{nm}^{GMM}(t)[U_n(t)P_m(t) - X_m(t)(1 - P_m(t))] \geq \frac{1}{2} \sum_{n,m} \mu_{nm}^{*}(t)[U_n(t)P_m(t) - X_m(t)(1 - P_m(t))]$$

Table 3.1 is the implementation of Greedy Maximal Match.

1: **For** each timeslot **do**
2: Flow control:
For each secondary user i **do**
if U(i)>V **then** packet arrives at input rate λ
else no packet is permitted.
End
3: Update status $S_m(t)$,$\boldsymbol{H}(t)$ and calculate P_m.
4: Resource Allocation:
For each channel m and secondary user n in BS i **do**
if $h_{mn}(t) > 0$ **then**
The weight of the edge (m, n) is:
$w_{mn}(t) = U_n(t)P_m(t) - X_m(t)(1 - P_m(t))$
end if
End
While at lease one remaining edge $w_{mn}(t) > 0$ **Do**
Find the largest weighted edge (i, j)
Channel i is allocated to secondary user j for transmission.
Delete all the edges connected to either channel i or secondary user j
End
5: Update the backlog queue $U_n(t)$ and collision queue $X_m(t)$.
6: **End For**

Table 3.1: Implementation of the Greedy Maximal Match

3.4.2 Simulations on CNC

We simulate our algorithm on two kinds of conditions: single primary user-per cell and multi primary user-single cell. The network is builded on a cell-partitioned architecture. The primary users are confined in their cells with their own licensed channels. A secondary user moves in the network randomly and access to the licensed channel opportunisticly.

The channel states $S_m(t)$ is governed by an ON/OFF Markov chain with the transition probabilities between ON and OFF given by 0.2. The maximum collision rate $\rho = 0.1$.

New packets arrives at the secondary users with the probability λ each slot.

1. Single Primary User-Per Cell

   We establish an example cognitive network consisting of 9 primary users and 8 secondary users as shown in Figure 3.5. The secondary users move from one cell to another

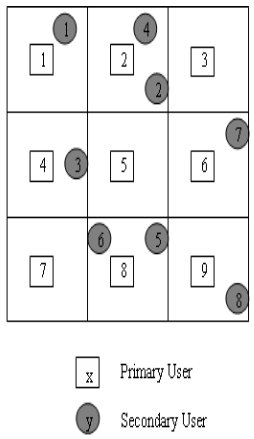

Figure 3.5: Example single primary user-multi cell network used in simulation

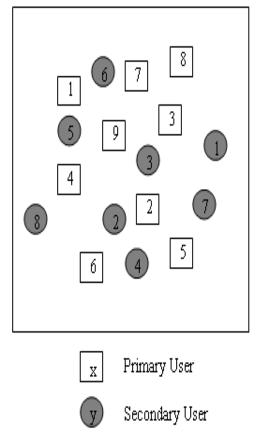

Figure 3.6: Example multi primary user-single cell network used in simulation

according to a Markovian random walk. In particular, at the end of every slot, a secondary user decides to stay in its current cell with probability $1-\beta$, else move to an adjacent cell with a probability $\beta/4(\beta = 0.25)$. If there is no feasible adjacent cell, the secondary user will stay in its current cell.

Since there is only one primary user in each cell, the maximum weight match is **same** as a greedy maximal match. Figure 3.7 plot the average total occupancy(summing all packets in the backlog queue of the secondary users) versus the input rateλ. Each data represents a simulation over 500,000 timeslots, and different curves correspond to values of $V \in \{1, 2, 5, 10, 100, \infty\}$. The network capacity for this network appears at $\lambda = 0.285$ packets/slot. Figure 3.8 illustrates the achieved throughput versus input rate. For small values of λ, the throughput is identical to the input rate, and the throughput gradually saturates depends on V. The throughput reaches the capacity level close to $\lambda = 0.285$ when V is large. Finally, the average collision rate is approximately to our target $\rho = 0.1$.

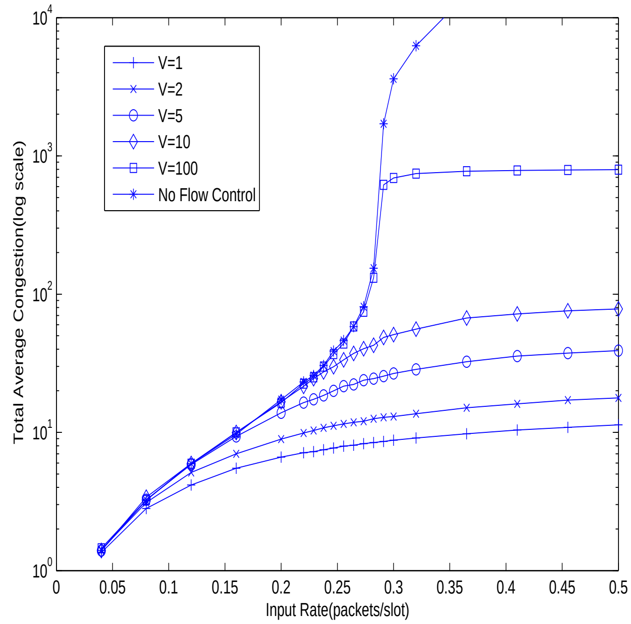

Figure 3.7: Total congestion vs. input rate for Single Primary User-Per Cell

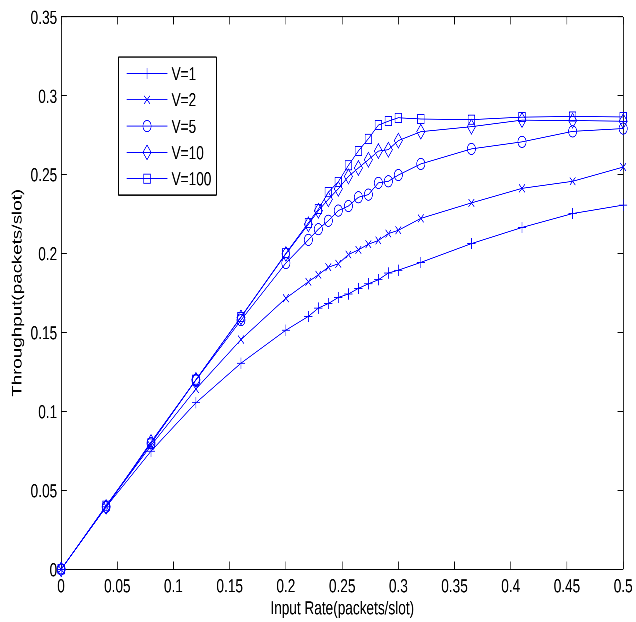

Figure 3.8: Throughput vs. input rate for Single Primary User-Per Cell

2. Multi Primary Users-Single Cell

   Previous example is a rather simplified case as there is only one primary user in one cell, hence the idea in the algorithm to cooperate with different channel can not be illustrated. We present another example consisting 9 primary users in one single cell with different number of secondary users. The set of the number of the secondary users is $\{4, 8, 10, 15, 20, 30\}$. The channel state process is same as in the previous example. The state that whether secondary users stay or leave the cell is governed by an ON/OFF Markov chain with symmetric transition probabilities given by 0.05.

   The Greedy Maximal Match used in this example can not achieve the optimal allocation, thus the network capacity remains to be improved. In the implementation of GMM, the computing overhead is much larger than previous example. Here we use the same parameter except that the total timesolts is reduced to 50,000.

   As can be seen in Figure 3.9, the achieved throughput remains almost the same when the number of secondary users is smaller than that of primary users. However, when secondary users increase in size, the throughput decreases rapidly since less channels are available to them. The flow control parameter we use here is $V = 100$.

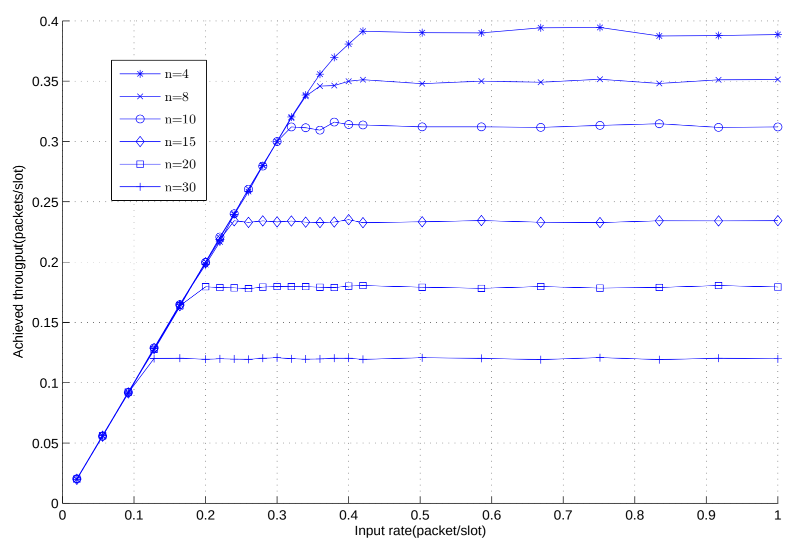

Figure 3.9: Throughput vs. input rate for Multi Primary Users-Single Cell with different number of secondary users

3.4.3 Discussion of the CNC model

The CNC model converts the problem of allocation channel between primary users and secondary users to the problem of maximal weight match. The weight of the match is used to guarantee the collision in the channels, and greedy maximal match is used to shorten computing overhead. This model can only be applied to the centralized-control network architecture. Moreover, the fairness of the secondary users to access the channel is not taken into consideration. Finally, since base station should be utilized to centralize resource allocation in CNC, another different algorithm is needed for ad-hoc network at which channel allocation can be done distributedly.

Bibliography

[1] J. Mitola. Software radios: wireless architecture for the 21st century. John Wiley & Sons Inc, 2000.

[2] J. Mitola et al., Cognitive radio: Making software radios more personal, IEEE Pers. Commun., vol. 6, no. 4, pp. 13C18, Aug. 1999.

[3] J. Mitola III, Cognitive radio for flexible mobile multimedia communication, in: Proc. IEEE International Workshop on Mobile Multimedia Communications (MoMuC) 1999, November 1999, pp. 3C10.

[4] I. F. Akyildiz, W-Y Lee, M. C. Vuran and S. Mohanty. NeXt generation/ dynamic spectrum access/cognitive radio wireless networks: a survey. Computer Networks Journal (Elsevier), vol. 50, pp. 2127-2159, Sept. 2006.

[5] Q. Zhao and B.M. Sadler. A survey of dynamic spectrum access: signal processing, networking, and regulatory policy. IEEE Signal Processing Magazine, vol. 24, no. 3, pp. 79-89, May, 2007.

[6] FCC, ET Docket No 03-222 Notice of proposed rule making and order, December 2003.

[7] C. Peng, H. Zheng, and B.Y. Zhao. Utilization and fairness in spectrum assignment for opportunistic spectrum access. ACM/Springer MONET, vol. 11, issue 4, Aug. 2006.

[8] L. Cao, H. Zheng. Distributed spectrum allocation via local bargaining. Proc. IEEE SECON, Sept. 2005.

[9] Q. Zhao, L. Tong, A. Swami, and Y. Chen. Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework. IEEE Journal on Selected Areas in Communications (JSAC): Special Issue on Adaptive, Spectrum Agile and Cognitive Wireles Networks, vol. 25, no. 3, pp. 589-600, April, 2007.

[10] Y. Chen, Q. Zhao, and A. Swami. Joint design and separation principle for opportunistic spectrum access. under submission to IEEE Transactions on Information Theory

[11] M. J. Neely, E.Modiano, and C. E Rohrs. Dynamic power allocation and routing for time varying wireless networks. IEEE Journal on Selected Areas in Communications, vol. 23, no. 1, pp. 89-103, Jan 2005.

[12] M. J.Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. Proc. IEEE INFOCOM, March 2005.

[13] M. J. Neely. Energy optimal control for time varying wireless networks. IEEE Transactions on Information Theory, vol. 52, no. 7, July 2006.

[14] Urgaonkar, Michael J. Neely: Opportunistic Scheduling with Reliability Guarantees in Cognitive Radio Networks. PROC. IEEE INFOCOM, PHOENIX, AZ, APRIL 2008

[15] C. H. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity. Englewood Cli, New Jersey: Prentice-Hall, 1982.

[16] X. Lin and N. B. Shroff. The impact of imperfect scheduling on crosslayer rate control in wireless networks. Proc. IEEE INFOCOM, March 2005.

[17] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. Proc. of 43rd Allerton Conference, Sept. 2005.

[18] S. Haykin, Cognitive radio: brain-empowered wireless communications, IEEE Journal on Selected Areas in Communications 23 (2) (2005) 201C220.

[19] H. Zheng and C. Peng, Collaboration and fairness in opportunistic spectrum access, in Proc. IEEE Int. Conf. Communications (ICC), 2005, pp. 3132-3136.

[20] W. Wang and X. Liu, List-coloring based channel allocation for open-spectrum wireless networks, in Proc. IEEE VTC, 2005, pp. 690-694.

[21] C. Peng, H. Zheng, and B.Y. Zhao. Utilization and fairness in spectrum assignment for opportunistic spectrum access. ACM/Springer MONET, vol. 11, issue 4, Aug. 2006.

[22] M.Halldorsson, J. Halpern, L. Li, and V. Mirrokni, "On spectrum sharing games," in Proc.23rd Annual ACM Symp. Principles Distributed Computing, 2004, pp. 107-114.