Codes and Games for Dynamic Spectrum Access

Yiping Xing, Harikeshwar Kushwaha, K.P. Subbalakshmi, and R. Chandramouli
Media Security, Networking and Communications Lab
Department of ECE
Stevens Institute of Technology

0.1 Introduction

Cognitive radio is an emerging wireless communications paradigm in which either the network or the wireless node itself intelligently adapts particular transmission or reception parameters by sensing the environment. The goals of adaptation include maximizing spectral efficiency, minimizing interference to other cognitive radios, co-existence of licensed and unlicensed band communications, battery energy efficiency etc. The environmental parameters that are continually sensed for adaptation include occupied radio frequency bands, user traffic, network state etc. One promising technology that enables the implementation of a cognitive radio network is software-defined radio. The underlying theoretical principles for cognition are broadly based on signal-processing and machine-learning. The cognitive cycle [1] consists of three major components: (a) sensing of radio frequency (RF), (b) cognition/management and (c) control action. In particular,

1. RF sensing:
   - estimation of the total interference in the radio environment;
   - detection of spectrum holes (or unused bands);
   - estimation of channel state information (e.g. signal to noise ratio (SNR));
   - prediction of channel capacity for use by the transmitter.
2. Cognition/management:
   - spectrum management which controls the opportunistic spectrum access;
   - traffic shaping;
   - routing;
   - quality of service (QoS) provisioning.
3. Control actions:
   - transmit-power control;
   - adaptive modulation and coding;
   - transmission rate control.

Task (1) deals with spectrum sensing and prediction, hence, places emphasis on physical layer issues. In this chapter we will focus on tasks (2) and (3). We discuss these tasks for dynamic spectrum access from game and coding theoretic perspectives. Stochastic learning based solution to some of the issues

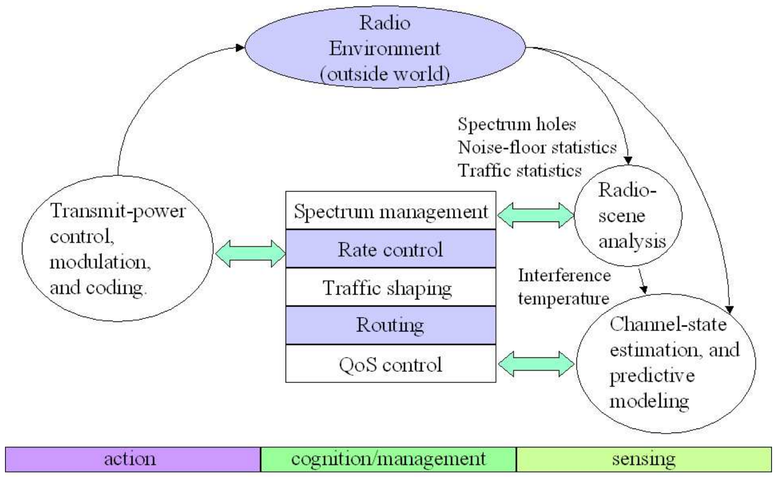

Fig. 0.1. Basic cognitive cycle.

in these tasks are also presented. The three tasks in a cognitive cycle are shown in Fig. 0.1

Dynamic spectrum access using cognitive radios is an emerging research topic. This research motivated by the fact that enhancing spectrum efficiency is an important task of regulatory authorities worldwide. Radio spectrum is considered a scarce resource with the growing demand for spectrum-based services because a major portion of the spectrum has been allocated for licensed wireless applications. Therefore new applications are competing for the very little spectrum that is left unlicensed. On the other hand, actual measurements taken on the 0-6 GHz band (Fig. 0.2) in downtown Berkeley show that the actual utilization of the spectrum is quite low. This re-iterates the argument that FCC's spectrum regulations diminish spectrum from being utilized efficiently. Other empirical studies (e.g., [8]) also seem to support the observation that spectrum is used inefficiently both in space and time.

Low utilization and increased demand for the radio resource suggest the notion of *secondary use*. This allows dynamic access to unused parts of spectrum owned by the primary license holder to become available temporarily for secondary (non-primary) users (SU). This dynamic access of spectrum by secondary users is one of the promising ideas that can mitigate spectrum scarcity, potentially without major changes to incumbents.

The first step in dynamic spectrum access is the detection of unused spectral bands. Therefore, a cognitive radio device measures the RF energy in a channel or monitors the received signal strength indicator to determine whether the channel is idle or not. But this approach has a problem in that wireless devices can only sense the presence of a primary user (PU) if and only if the energy detected is above a certain threshold. It is true that one can not arbitrarily lower the threshold as this would result in non-detection because of the presence of noise. In the feature detection approach, which has been used in the military to detect the presence of weak signals [9], the wireless device uses cyclostationary signal processing to detect the presence of primaries. If a signal exhibits strong cyclostationary properties, it can be detected at very low signal-to-noise ratios (SNR) [11]. Then, the question is how to share the available spectrum efficiently and fairly.

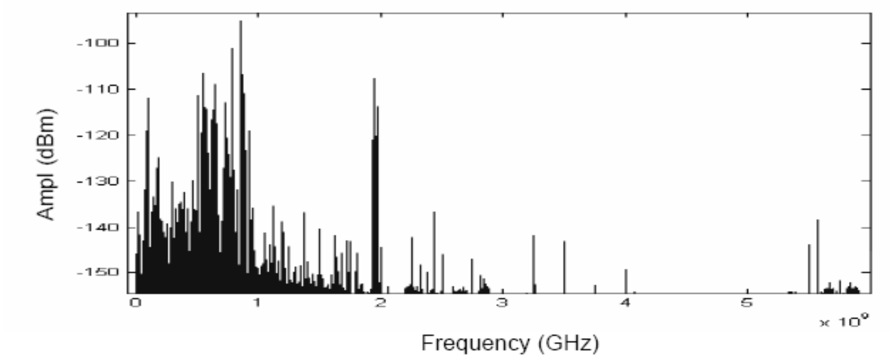

Fig. 0.2. Spectrum utilization snapshot at Berkeley.

The FCC spectrum Policy Task Force [12] has recommended a paradigm shift in interference assessment from the largely fixed operations. This facilitates real-time interactions between a transmitter and a receiver in an adaptive manner. The recommendation is based on a new metric called the *interference temperature*, which is intended to quantify and manage the sources of interference in a radio environment. The interference temperature is defined to be the RF power measured at a receiving antenna per unit bandwidth. The key idea for this new metric is that, firstly, the interference temperature at a receiving antenna provides an accurate measure for the acceptable level of RF interference in the frequency band of interest; any transmission in that band is considered to be "harmful" if it would increase the noise floor above the interference temperature threshold as shown in Fig. 0.3. Secondly, given a particular frequency band in which the interference temperature is not exceeded, that band could be made available to secondary users. Hence, a secondary device might attempt to coexist with the primary, such that the presence of secondary devices goes unnoticed.

Game theory plays an important role in the modelling and optimization of wireless communications among users with competing performance goals and resource constraints. For example, secondary users compete to access the

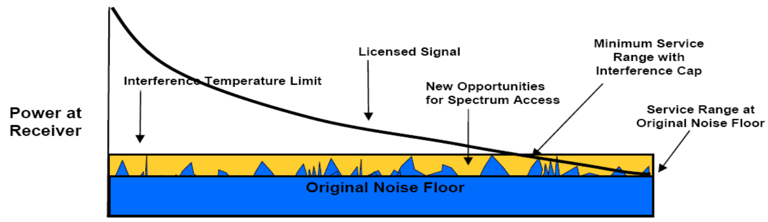

Fig. 0.3. This figure [14] shows the power that is received from the licensed transmitter as a function of distance. The spectrum agile radio has opportunities represented by the righthand side box.

available spectrum. Typically, these users do not co-operate with each other and selfishly try to maximize their own utility (e.g., throughput). This can be viewed as a non-cooperative game. Is there an equilibrium transmission strategy set for this game? If such a strategy set exists, how can it be computed by the user in a distributed fashion using only local information? These are all important questions that we address later in this chapter. We provide some game theoretic models for dynamic spectrum access and study the existence of equilibrium solution. Stochastic learning based techniques to discover the equilibrium solution is also discussed.

In the secondary usage scenario PUs are the license holders and secondary users SUs are allowed to use the band when PU's are inactive, but they are required to vacate the band as soon as PUs become active. The arrival of the PU causes the SU to loose that band and therefore link maintenance becomes a key issue. A solution to this problem has been proposed in [21] where the whole bandwidth is divided into a large number of subchannels as shown in Fig. 0.4. A SU selects a set of subchannels in such a way that they lie in different primary users' bands. This minimizes the loss of subchannels upon the arrival of a particular PU as no PU can cause the complete breakdown of the SU link. But the arrival of a PU acts like an erasure for a SU link and it causes the SU to loose all the packets that are being transmitted over the subchannel which was under that particular PU's band. In counteracting the effect of PU's interference over the SU's link, a class of erasure correction codes called *LT codes* or *Digital Fountain Codes* [22, 23] play an important role in the link maintenance mechanism. We consider the use of these codes for secondary spectrum access and provide some analysis and simulations.

The chapter is organized as follows. Preliminaries of stochastic automata games are discussed in Section 0.2. Co-existence issues in dissimilar secondary radio systems is presented in Section 0.3. Section 0.4 deals with dynamic spectrum access with QoS and interference temperature constraints. Digital fountain codes for link maintenance is presented in Section 0.5.

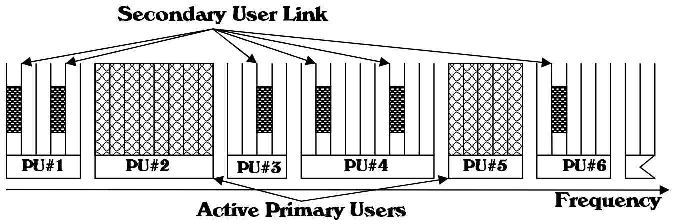

Fig. 0.4. Spectrum pooling.

0.2 Stochastic Learning Automata and Games

Abstractly, a learning automaton [2] can be considered to be an object that can choose from a finite number of actions. For every action that it chooses, the random environment in which it operates evaluates that action. A corresponding feedback is sent to the automaton based on which the next action is chosen. As this process progresses the automaton learns to choose the *optimal* action for that unknown environment asymptotically. The stochastic iterative algorithm used by the automaton to select its successive actions based on the environment's response defines the stochastic learning algorithm. An important property of the learning automaton is its ability to improve its performance with time while operation in an unknown environment. In this chapter, for the sake of consistency our notations follow or parallels that from standard books on game theory (e.g., [3]) and stochastic learning [4].

In multiple automata games, instead of one automaton (player) playing against the environment, N automata, say $A_1, A_2, ..., A_N$ take part in a game. Consider a typical automaton A_i described by a 4-tuple $\{S_i, r_i, T_i, \mathbf{p}_i\}$. Each player i has a finite set of actions or pure strategies, $S_i, 1 \leq i \leq N$. Let the cardinality of S_i be $m_i, 1 \leq i \leq N$. The result of each play is a random payoff to each player. Let r_i denote the random payoff to player $i, 1 \leq i \leq N$. It is assumed here that $r_i \in [0, 1]$. Define functions $d^i : \Pi_{j=1}^{N} S_j \rightarrow [0, 1], 1 \leq i \leq N, by$

$$\begin{aligned} d^i(a_1, ..., a_N) = \; & E[r_i | player\ j\ chose\ action\ a_j, \\ & a_j \in S_j, 1 \leq j \leq N]. \end{aligned} \tag{0.1}$$

The function d^i is called the *expected payoff* function or *utility* function of player $i, 1 \leq i \leq N$. The objective of each player is to maximize its expected payoff. Players choose their strategies based on a time-varying probability distribution. Let $\mathbf{p}_i(k) = [p_{i1}(k)...p_{im_i}(k)]^t$ denote the action choice probability distribution of the i^{th} automaton at time instance k. Then $p_{il}(k)$ denotes the probability with which i^{th} automaton player chooses the l^{th} pure strategy at instant k. Thus $\mathbf{p}_i(k)$ is the strategy probability vector employed by the i^{th} player at instant k. T_i denotes the stochastic learning algorithm according to which the elements of the set $\mathbf{p}_i$ are updated at each time k, i.e.,

$\mathbf{p}_i(k+1) = T_i(\mathbf{p}_i(k), a_i(k), r_i(k))$, where a_i and r_i are the actual actions selected by i and the payoff received by i respectively at $k, k = 0, 1, 2, ...$ The working of a learning automaton say A_i can be described as follows. Initially at $k = 1$ one of the actions is chosen by the player at random with an arbitrarily chosen initial probability. This action is then applied to the system and the response from the environment is observed. Based on the response r_i, the probabilities of actions $\mathbf{p}_i$ for the next period of time are updated according to the updating rule T_i. This process is repeated by all the N players until a stopping criterion is reached or the probability vector converges. Fig.0.5 depicts the basic automata game set-up.

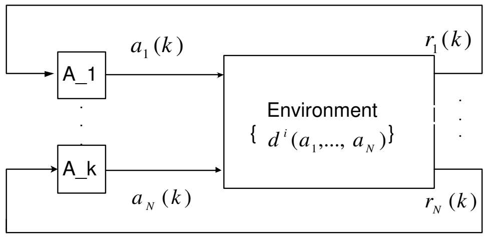

Fig. 0.5. The basic multiple automata game set-up.

We will frequently consider the situation of varying the strategy of a single player i while holding the strategies of his opponents fixed. Here, we let $s_{-i} \in S_{-i}$ denote a strategy selection for all players but i, and write (s'_i, s_{-i}) for the profile $(s_1, \cdots, s_{i-1}, s'_i, s_{i+1}, \cdots, s_N)$.

A strategy for player i is defined to be a probability vector $\mathbf{p}_i = [p_{i1}, ..., p_{im_i}]^t$, where player i chooses action j with probability p_{ij}. Then we can define the expected payoff for player i as g^i given by,

$$\begin{aligned} g^i(\mathbf{p}_1, ..., \mathbf{p}_N) &= E[u_i | j^{th} \textit{ player employs strategy} \\ \mathbf{p}_j, 1 \leq j \leq N] &= \sum_{j_1,...,j_N} d^i(j_1, ..., j_N) \prod_{s=1}^{N} p_{sj_s} \end{aligned} \quad (0.2)$$

Definition 1 *The N-tuple of strategies $(\boldsymbol{p}_1^o, ..., \boldsymbol{p}_N^o)$ is said to be a Nash equilibrium, if for each i, $1 \leq i \leq N$, we have*

$$\begin{aligned} & g^i(\boldsymbol{p}_1^o, ..., \boldsymbol{p}_{i-1}^o, \boldsymbol{p}_i^o, \boldsymbol{p}_{i+1}^o, ..., \boldsymbol{p}_N^o) \\ & \geq g^i(\boldsymbol{p}_1^o, ..., \boldsymbol{p}_{i-1}^o, \boldsymbol{p}_i, \boldsymbol{p}_{i+1}^o, ..., \boldsymbol{p}_N^o), \quad \forall \boldsymbol{p}_i \in [0,1]^{m_i}. \end{aligned} \quad (0.3)$$

A Nash equilibrium is said to be in *pure* strategies if $(\mathbf{p}_1^o, ..., \mathbf{p}_N^o)$ is a Nash equilibrium with each $\mathbf{p}_i^o$ being a unit probability vector. While, it is a non-degenerate mixed Nash equilibrium if the strategy assigns positive probabilities on more than one pure strategy. In general, each $\mathbf{p}_i^o$ above may be a mixed strategy and we refer to $(\mathbf{p}_1^o, ..., \mathbf{p}_N^o)$ satisfying (0.3) as a Nash equilibrium in *mixed* strategies. With this definition, when there is no pure equilibrium as maybe the case in the stochastic automata game, the users can search for a mixed Nash equilibrium. It is well known that *every finite strategic-form game has a mixed strategy equilibrium* [3].

A strategy dominates another when, independent of the action taken by the other players, the first strategy leads to an outcome at least as favorable as the second. Consequently, a case can be made for never playing a strategy that is dominated. Although this is indeed true in a two-person zero-sum games and in identical payoff games, in general nonzero-sum games the use of a dominated strategy can lead to a better payoff for all players (e.g., the Prisoner's Dilemma).

In an N-person game, an outcome is said to be *Pareto-optimal* if there is no other outcome in which all players simultaneously do better. This is a modest requirement of any concept of group rationality. Unfortunately, it is not always consistent with concepts of individual rationality, such as dominance.

0.3 Coexistence of Dissimilar Secondary Radio Systems

Coexistence of dissimilar secondary radio systems in a spectral band raise two important issues: fairness and efficiency. To address these, we propose an access technique inspired by a human society model—*Homo Egualis.* Homo Egualis society consists of a group of agents who care not only about its own payoff, but also about how it compares with the payoff of others. In the human society, homo sapiens evolved in small hunter-gather groups. Such societies have no centralized structure of governance (state, judicial system, church). So the enforcement of norms depends on the voluntary participation of peers. A Homo Egualis society [5] can be modelled as follows, where the utility function of player i, u_i in an n-player game is:

$$u_i = x_i - \frac{\alpha_i}{n-1} \sum_{x_j > x_i} (x_j - x_i) - \frac{\beta_i}{n-1} \sum_{x_j < x_i} (x_i - x_j), \tag{0.4}$$

where $x = (x_i, \ldots, x_j)$ are the pay-offs for each player and $0 \leq \beta_i < \alpha_i \leq 1$. $\beta_i < \alpha_i$ reflects the fact that Homo Egualis exhibits a weak urge to increase inequality when doing better than the others and a strong urge to reduce inequality when doing worse than the others. In [5] it is shown that in this model the salient behaviors in ultimatum and public goods games, where fairness does matter, can be reproduced. We proposed a distributed dynamic spectrum access scheme for dissimilar cognitive radio networks in [7] based on the Homo Egualis society model.

The 5 GHz unlicensed frequency band is a candidate for a large set of radio services, and is one of the unlicensed frequency bands that may be efficiently used only with an established spectrum etiquette. We use the same abstract model of an unlicensed frequency band as in [10], as illustrated in Fig. 0.6. Here, three different types of radio systems are assumed to operate in the band, each operating with different frequencies and channel bandwidths. The radio systems of type A operate on three frequency channels (center frequencies f_2, f_5, f_8), the radio systems of type B operate on nine frequency channels (center frequencies $f_1 \dots f_9$), and radio system of type C operates on one frequency channel (center frequency f_5). The frequency channels overlap with each other, as indicated in the figure. The number and bandwidth of the frequency channels in Fig. 0.6 do not represent any existing unlicensed band, this usage model serves without loss of generality only as an example model. Here radio system A can be compared to wireless LANs operating in the 5 GHz band (using OFDM). Radio system B represents narrow-band radio systems supporting for example a limited number of voice calls or blue tooth systems, while radio system C can be a broadband CDMA system. In our scenario instead of modelling the detailed protocols, a simplified listen before talk (LBT) is used for all radio systems. A type A radio system requires the respective three frequency channels to be idle before allocating radio resources. Collisions of allocation attempts occur when more than one radio system detects the channel as idle at the same time. In simulations, when collisions happen, one of the radio systems is randomly selected to allocate the radio resource.

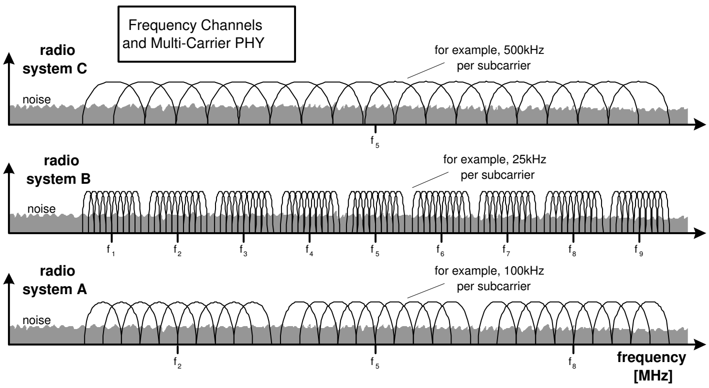

Fig. 0.6. Frequency channels and multi-carrier PHY.

Two of the most representative etiquette rules [10] are as follows:

- rule ♯4: a radio system of type A, type B or type C should apply LBT when operating.
- rule ♯6: in order to protect other radio systems most efficiently, a radio system that follows rule ♯4 should synchronize its LBT process in time across neighboring frequency channels that overlap with the same channels.

One of the most important metrics in the unlicensed band access is the average airtime per radio system. Airtime refers to the ratio of allocation time per radio system type to the reference time (say one hour) [10]:

$$airtime_{type=A,B} = \frac{1}{N_{type}} \sum_{i=1}^{N_{type}} \frac{allocation\ time(i)}{reference\ time}, \tag{0.5}$$

where N_{type} is the number of type i radio systems.

Efficiency and fairness are obviously the main goals of a spectrum etiquette. If every radio system accesses the unlicensed band in a greedy manner, then the radio system requiring broader band to operate will suffer from an unacceptable low airtime share. So one way to provision fairness to the etiquette rules would be to require each radio system to work in a cooperative manner. One option would be that each radio system i tries to contend for the spectrum with probability p_i. After the radio system has decided to contend for the spectrum it accesses the spectrum compliant to etiquette rule ♯4 described before. If perfect fairness is achieved and each radio system is maximizing its airtime share, then it can be shown that [7] there exists an access probability vector $(p_{a_{opt}}, p_{b_{opt}}, p_{c_{opt}})$ that corresponds to the strategy that no radio system can do better in terms of airtime share without harming the other coexisting radio systems. So in this sense, both the efficiency and fairness are obtained by using this optimal probability pair. Obviously, without considering any fairness issues, the most efficient access in the sense of pure spectrum utilization would be that all the users compete for the spectrum greedily. But, using this scheme some type of networks may be totally blocked out of the spectrum access. And further, if different types of radio systems have different traffic load a simple equalization of the *airtime* may cause low spectrum utilization. Also, exact fairness does not always apply as far as an operator is concerned, they are likely to be more concerned with revenue. So users who pay more will get more access, which means different networks may have different priorities. To properly address these problems, we need to define the fairness and efficiency carefully.

Definition 2 *Weighted fairness is achieved if the following equation holds,*

$$\frac{airtime_{type_i}}{L_i} = \frac{airtime_{type_j}}{L_j} = K \quad \forall i, j. \tag{0.6}$$

where K is a constant, and the weight $L_i = \theta_i \lambda_i$, θ_i is the priority parameter and λ_i is the traffic load for type i radio system.

Definition 3 *Given the weighted fairness of different types of radio system is satisfied, efficiency is achieved if each of the radio system's airtime is maximized.*

Note that this definition of efficiency actually corresponds to the Pareto efficiency solution in game theory. A strategy profile is Pareto optimal if some players must be hurt in order to improve the payoff of other players [6].

To obtain $(p_{a_{opt}}, p_{b_{opt}}, p_{c_{opt}})$ we need the information of all the arrival rates λ's and service rates μ's of the the radio systems which is impractical in a real access scenario. A more realistic scheme would be to allow the radio systems to learn the values p_a, p_b and p_c themselves with only local information or measurement. We present such a Homo Egualis (HE) based technique next. The inequality aversion property of the Homo Egualis agents can be utilized to achieve fairness in the spectrum access problem. In this scheme each radio system learns the access probability p_i by itself. Here we define $Onlinetime_i$ as the averaged cumulative "on" spectrum time per radio system of type i. Then we define $x_i = \frac{Onlinetime_i}{L_i}$, where $L_i = \theta_i \lambda_i$ is the same as used before in (0.6). The cumulative $Onlinetime_i$ is normalized by the radio system's traffic load and priority, which makes this spectrum access scheme able to adapt to different traffic loads and priority, hence achieve more efficiency and maintain our defined weighted fairness. With initial $p_i = 1$, each time the probability p_i is updated as follows,

$$\begin{aligned} p_i =& max(0, min(1, p_i + \frac{\alpha_i}{n-1} \sum_{x_j \geq x_i} (\frac{x_j - x_i}{x_j}) \\ & - \frac{\beta_i}{n-1} \sum_{x_j < x_i} (\frac{x_i - x_j}{x_i}))) \quad for\ all\ j \neq i, \end{aligned} \tag{0.7}$$

where, n is the number of different radio system types, $0 < \beta_i < \alpha_i$ reflects the fact that radio system exhibits a weak urge to when doing better than others and a strong urge to reduce inequality when doing worse than the others. This forces each radio system to make an effort to efficiently use the idle spectrum while taking fairness into consideration. Here the only local information needed is the radio system's own history of the *Onlinetime* and the *Onlinetime* of the other radio systems whose spectrum is within the same spectrum block. This can be obtained by keeping a record of the busy time of the required spectrum, which can be obtained by periodical spectrum scanning. When there are more than two different radio systems trying to coexist in the same spectrum, different radio systems can be identified by some smart technologies (e.g. we can detect the different transmitting power levels to distinguish from different radio systems). So each radio system can access the spectrum based only on its own recorded history and the local measurements performed by itself. λ_i can be estimated by historical usage record of radio system type i. The priority parameter θ_i needs to be announced by the radio system or be broadcast by some operator. If each radio system can only gather an imperfect information about the spectrum usage time of other radio systems, then there may be some degradation in the achieved weighted fairness.

We model the arrival traffic as a Poisson process. It can be seen from Fig. 0.7 and Fig. 0.8 that all the different radio systems have almost the same airtime share and blocking probability by using the proposed HE access scheme, which is what is desired for fairness.

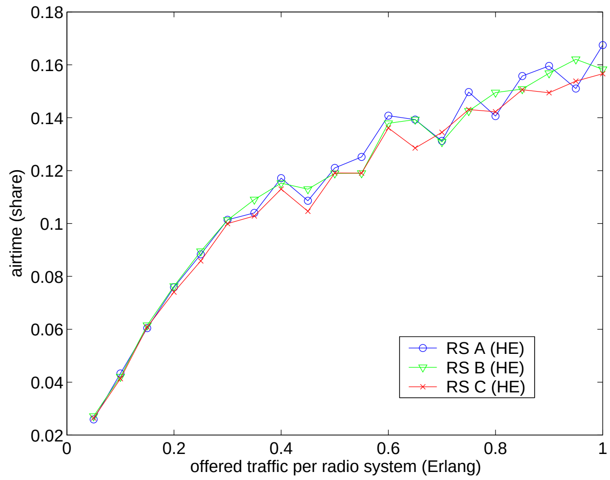

Fig. 0.7. *airtime* v.s. offered load for HE access scheme. Three different types of spectral agile radio systems with no queuing.

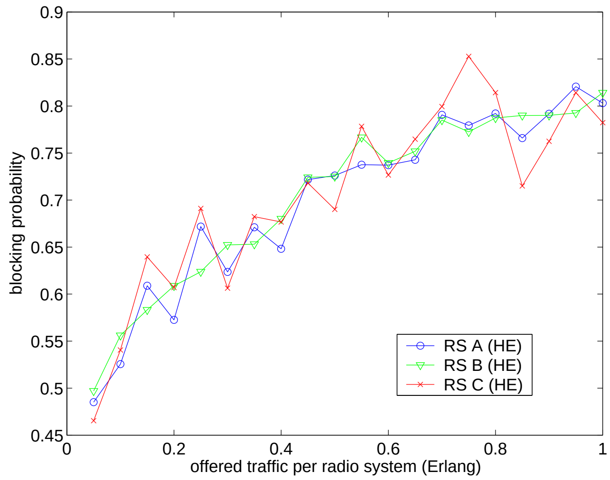

Fig. 0.8. Blocking probability v.s. offered load for HE access scheme. Three different types of spectral agile radio systems with no queuing.

0.4 Impact of QoS and Interference Temperature Constraints

Previous work on secondary use of radio spectrum include [13, 15, 16, 17, 18, 7]. Let us consider a scenario similar to [18], where secondary users wish to use a local, relatively short-term data service, and all users adopt a spread spectrum signaling format, in which the transmitted power is evenly spread across the entire available band controlled by the manager. A practical realization of this model would be when secondary users (with spread spectrum signaling) and primary direct-sequence CDMA (DS-CDMA) system coexist in the up-link spectrum band of the primary DS-CDMA system. In the uplink, the interference temperature can be measured at the base station which is the receiver of the primary system. Hence, the number of measuring points can be significantly reduced. But of course secondary access is definitely not limited to this scenario. The spectrum can be available in a TV broadcast band or other emergency band.

0.4.1 Secondary Spectrum Sharing Model

Spectrum with bandwidth W is to be shared among M spread spectrum users, where a user refers to a transmitter and an intended receiver pair. For each i, the received signal to interference ratio (SIR) is given by

$$\gamma_i = \frac{y_i h_{ii}}{\frac{1}{L}(\sum_{j \neq i} y_j h_{ji}) + \sigma^2}, \tag{0.8}$$

where L is the normalized spreading sequence length, y_i is user $i's$ transmission power, h_{ij} is the channel gain from user $i's$ transmitter to user $j's$ receiver, and σ^2 is the background noise power that is assumed to be the same for all users. In order to satisfy an interference temperature constraint, the total received power at a specified measurement point must satisfy:

$$\sum_{i=1}^{M} y_i h_{i0} \leq B, \tag{0.9}$$

where h_{i0} is the channel gain from user i's transmitter to the measurement point, and $B > 0$ is a pre-defined threshold. We assume that all the secondary users adopt a spread spectrum signaling format, in which the transmitted power is evenly spread across the entire available band. This allows efficient multiplexing of data streams from different sources corresponding to different applications, and reduces the combined power-bandwidth allocation problem to a received power allocation problem. Hence, the interference temperature constraint is translated to a total received power threshold B at the measuring point.

0.4.2 Spectrum Assignment with Priority Classes

If different users have different contracts defining different priorities or throughput, then it is acceptable for a spectrum operator to provide different throughputs. Let us say that secondary links, depending on their willingness to pay belong to L priority classes. Let a_i be the priority parameter for link i. The operator problem is then to maximize the network revenue:

$$\begin{array}{lll} \max \sum_{i\in(i:x_i=1)} a_i. & & \\ \text{subject to} & & \\ SIR_i & \geq \gamma_i^t & \forall i \in (i : x_i = 1) \\ \sum_i h_{i0} y_i & \leq B & \forall i \in (i : x_i = 1) \\ y_i & > 0 & \forall i \in (i : x_i = 1) \\ y_i & \leq y_i^{max} & \forall i \in (i : x_i = 1). \end{array} \tag{0.10}$$

where x_i is a collection of binary variables, $x_i = 1$ means the i^{th} link transmits otherwise $x_i = 0$. By maximizing this revenue, secondary users who pay more will get accessing priority over those who pay less. We model the relation between the price p_i a user paid and the priority parameter a_i as follows,

$$a_i = p_i^{\alpha} \tag{0.11}$$

where $0 \leq \alpha \leq 1$ is an operator designable parameter. Small α corresponds to putting more emphasis on system capacity (number of active secondary links), while large α corresponds to putting more emphasis on guaranteeing service to the user paying higher price. Specifically, $\alpha = 0$ corresponds to the problem of maximizing the number of active secondary links (capacity).

0.4.3 Secondary Spectrum Sharing Potential Game

The nature of secondary spectrum sharing is temporary and distributed. Therefore a practical secondary spectrum sharing scheme must be distributed. In this section, we discuss such a distributed algorithm to solve the operator problem discussed before. This distributed process is composed of two phases. The coordination phase controls the optimal set of active secondary links which can access the spectrum, and the power control phase is to allocate transmit power to support the minimum target link SIR γ_i^t given the set of active links.

Potential Games

Suppose there are M transmitter and receiver link pairs competing for the secondary spectrum access opportunities. Let k be a time (iteration) counter and $N(\mathbf{x}(k))$ be the aggregate received power at the measuring point at time k given by,

$$N(\mathbf{x}(k)) = \sum_{i=1}^{M} y_i(k) h_{i0} x_i(k), \tag{0.12}$$

In order to maximize the network revenue while keeping the aggregated received power at the measuring point under the interference threshold, we define the utility function $u_i(\mathbf{x})$ shown in Fig. 0.9 for each link pair as follows,

$$u_i(x) = \begin{cases} \frac{\sum_{j\in(j:x_j=1)} a_j}{3\sum_j a_j} + \frac{2}{3}, & N(\mathbf{x}(k)) < B, \min_{j\in(j:x_j=1)} \gamma_j > \gamma^t \\ \frac{B}{3N(\mathbf{x}(k))} + \frac{1}{3}, & N(\mathbf{x}(k)) \geq B, \min_{j\in(j:x_j=1)} \gamma_j > \gamma^t \\ \frac{\min_{j\in(j:x_j=1)} \gamma_j}{3\gamma^t}, & \min_{j\in(j:x_j=1)} \gamma_j < \gamma^t \end{cases} \tag{0.13}$$

In the secondary spectrum sharing game each user maximizes its utility function $u_i(\mathbf{x}(k))$ by its choice of being active or not. By maximizing this utility function, the system will reach an operating point where the network revenue is maximized while satisfying QoS and interference temperature constraints. To emphasize that the i^{th} user has control only over its own choice, we use an alternative notation $u_i(x_i, \mathbf{x}_{-i})$, where $\mathbf{x}_{-i}$ denotes that vector consisting of elements of $\mathbf{x}$ other than the i^{th} element. And after each changing of the active link set, the distributed power control algorithm DCPC [20] will be activated to allocate the transmitting power. Results about the existence of pure strategy Nash equilibrium for the potential game etc. can be shown to hold. A practical method to compute the Nash equilibrium would be to use the sequential play where each player maximizes its own utility function sequentially while other players' strategies are fixed. The sequential play will never converge to a solution where the total received power at the measuring point exceeds the interference temperature bound. Also, the sequential play will always converge to a solution where all the active links are supported with their target SIR. But, sequential play does not allow asynchronous updates by individual users. This may cause signaling and other overhead. To overcome this issue we consider a stochastic learning based distributed solution that is described next.

Stochastic learning techniques has been successfully used in wireless packet networks for on-line prediction, tracking and discrete power control [19] [20]. It is shown to be computationally simple and efficient. Learning algorithm determines probabilistic strategies for players by considering the history of play. The probability updating algorithm used by each user is as given below:

1. Set the initial probability $p_i(0)$.
2. At every time step k, the i^{th} user chooses $x_i(k) = 1$ or 0 (to transmit or not) according to its action probability p_i.
3. After distributed power control (DCPC [20]) phase, each player obtains a feedback $u_i(\mathbf{x}(k))$ based on the set of all actions $\mathbf{x}$.
4. Each player (i) updates its action probability according to the rule:

$$\begin{aligned} p_i(k+1) &= p_i(k) + b u_i(k)(1 - p_i(k)) \quad x(k) = 1 \\ p_i(k+1) &= p_i(k) - b u_i(k) p_i(k) \qquad\quad x(k) = 0, \end{aligned} \tag{0.14}$$

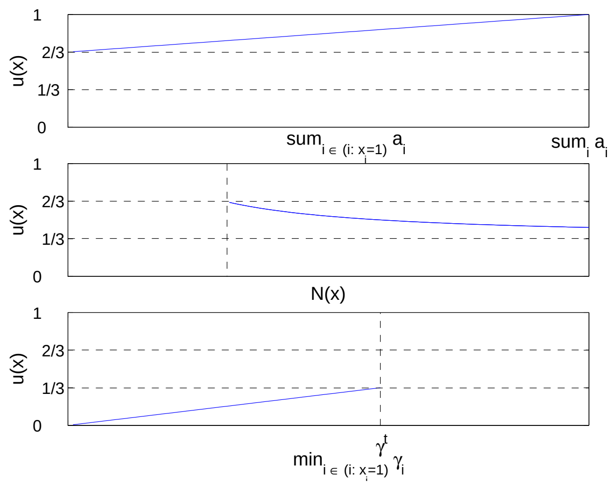

Fig. 0.9. Utility function $u_i(N(\mathbf{x}))$.

where $0 < b < 1$ is the step size, and $u_i(k)$ is utility function which lies in the interval $(0, 1)$.

5. If p_i converges, stop. Otherwise, go to step 2).

The probabilistic update used in (0.14) is a stochastic learning automata updating known as linear reward-inaction (L_{R-I}) [2].

It can be proved that the proposed learning algorithm will converge to one of the Nash equilibria of the game. It will never converge to a point where the total received power at the measuring point exceeds the interference temperature bound. The algorithm will always converge to a point where all the active links are supported with their target SIR.

The learning automata algorithm needs less information and control signalling to operate than the sequential play. The sequential play requires that each player updates its strategy one by one. And at the time of decision, in order to compute the utility function, the information required includes all active users' SIRs, priorities, and the current interference temperature. Significant control signalling is also required to accomplish this process distributively. An alternative choice would be to run this sequential play at a central controller. For the learning automata, asynchronous updating is permitted. The only information needed is a feedback of the current utility function value. To compute this utility function, each active user should report its SIR and priority parameter to the measurement point which acting also as a central controller. But the tradeoff is that learning automata converges much slower than the sequential play. The information needed by these algorithms can be distributed through a common control channel which has been assumed in many similar literatures in this area.

One assumption in this chapter is that the interference temperature remains constant during the secondary use of the spectrum. With mobile users, the interference temperature may vary. One solution to account this variance would be that the secondary access algorithm is triggered periodically. A short period would ensure that when the interference temperature is violated, the system would recover quickly.

Here, we first present some numerical examples for a simple secondary sharing system with only four transmitting and receiving pairs. The target SIR is selected to be $\gamma^t = 12.5$, and the noise power is $\sigma^2 = 5 \times 10^{-13}$, which approximately corresponds to the thermal noise power for a bandwidth of 1 MHz. We consider low rate data users, using a spreading gain of $L = 128$. Path gains are obtained using the simple path loss model $h_j = K/d_j^4$ where $K = 0.097$. We set the interference temperature bound and noise ratio to be $B/\sigma^2 = 60$ (We use these ratios only to illustrate clearly how the system works. Of course we can use lower ratio by reducing the target SIR). Under equal priority case $\mathbf{a} = [1, 1, 1, 1]$. When the learning automata algorithm is used, the evolution of the total received power $N(\mathbf{x})$ is shown in Fig. 0.10. As discussed, we see that the total received power converges to a value below the interference temperature threshold B. The evolution of the choice probability p is shown in Fig. 0.11. After convergence, only links 2 and 3 are active with equal probability initialization. It can be observed in Fig. 0.10 that during the settling phase, $N(x)$ exceeds B. So, depending on the application, there should be some predefined extra margin to accommodate the fluctuation during the settling phase. But this may reduce the capacity of the secondary system.

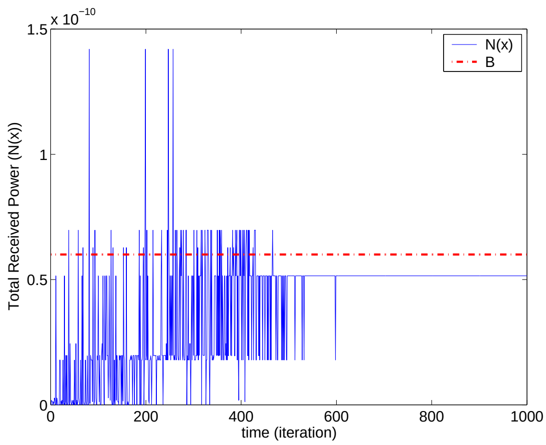

Fig. 0.10. Evolution of the total received power $(N(x))$ over time for the stochastic learning algorithm.

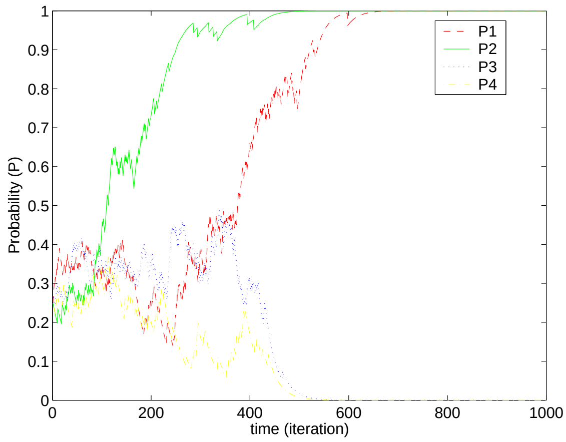

Fig. 0.11. Choice probability p of the activation strategy over time for the stochastic learning algorithm.

Despite the sub-optimal nature of the sequential play algorithm, the convergence speed is dramatically reduced as shown in Fig. 0.12. It can be seen that the convergence speed is linear with respect to the number of secondary link pairs.

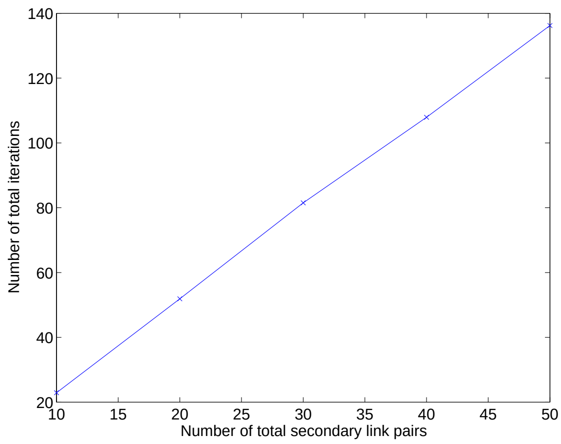

Fig. 0.12. Convergence of the sequential play

Note that bit errors can occur even after optimal access, power control etc. If an error correcting code fails to correct these errors then this will lead to packet losses–erasures at higher layers. Therefore, it is important to consider erasure channels and techniques to mitigate packet losses. We discuss a strategy that addresses this in the next section.

0.5 Fountain Codes for Dynamic Spectrum Access

In this section, we discuss a new class of *erasure correction codes* for packet-based channels with erasures, called *digital fountain codes* [24, 25, 22, 23], which can be used to provide packet-level protection at the transport layer or higher, augmenting the bit-level protection that may be provided by the medium access control (MAC) and PHY layers. These codes are capable of correcting *missing* or *lost* data. By recovering lost data packets without requiring retransmission from the sender, these codes efficiently and effectively provide reliability in data networks. And, like a water fountain producing an endless supply of water drops, any of which can be used to fill a glass, these fountain codes can generate an unlimited number of encoded output packets, any of which can be used to recover the original input packets. They have the following characteristics:

- The ability to generate a potentially limitless amount of encoded data from any original set of source data and providing reliable message delivery over extremes of low to high network losses.
- The ability to recover the original data from a subset of successfully received encoded data regardless of which specific encoded data has been received.
- Exceptionally fast encoding and decoding algorithms–operating at nearly symmetric speeds that grow only linearly with the amount of source data to be processed and independently of the actual amount of network loss.

0.5.1 Erasure Channels

In erasure channels packet losses occur at the receiver due to a variety of reasons. For example, if the bit error correcting code fails, then the erroneous packets may not be passed on to the higher layers in the protocol stack. Therefore, the application layer sees this as an erasure. In erasure channels we can assume that a packet is either received in-tact or is lost.

In a wireless network, packet loss generally occurs due to following reasons:

- Packets get discarded en route to their destination for various reasons such as buffer overflows and congestion control at intermediate routers.
- Packets get corrupted due to noise and interference.
- In a secondary user environment, when a primary user occupies a channel, secondary users loose their packets on that channel.

A M-ary erasure channel is shown in Fig. 0.13, where all input points $\{0, 1, 2, \ldots, m-1\}$ have a probability $1-p$ of being received correctly and a probability p of being lost or discarded (i.e. erased). If the perfect channel (i.e. $p = 0$) capacity is C then the capacity of the erasure channel is $(1 - p)C$.

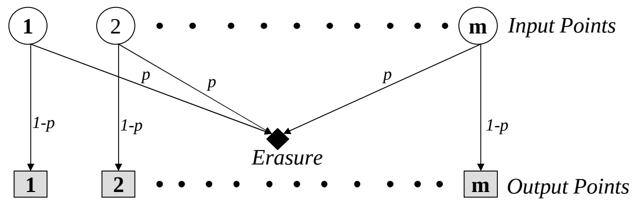

Fig. 0.13. M-ary erasure channel model.

The traditional method for having reliable communication over such type of channels uses feedback to request for retransmission in order to make up for erased packets, but they perform poorly in the following circumstances:

- One-to-many transmission: In broadcast, if all receivers request for retransmission then the number of retransmissions at the sender is going to be very high which decreases the overall efficiency.
- Transmission over bad channels (i.e. if p is high), here again the number of retransmissions will be high causing the overall performance to suffer.
- If the sender-receiver distance is large then the effect of propagation delay due to retransmissions may be significant.

To deal with the above problems of feedback-based protocols new transmission solutions using erasure correcting codes have been proposed. So if some bits in a packet are lost or errors could not be corrected using a forward error correcting (FEC) code then it may be recovered using erasure correcting codes. It is required that these codes be capable of correcting as many erasures as possible and at the same time have fast encoding and decoding algorithms.

Since the channel capacity is $(1-p)C$ independent of the feedback channel, erasure correcting codes try to achieve this capacity without using feedback. Traditional erasure codes are $[N, K]$ block codes such as Reed-Solomon (RS) and Tornado codes [22, 23]. In these codes, any K or slightly more than K out of N outputs are sufficient to recover the original K inputs. But they are practical only for small values of K and N. Furthermore, both the codes are fixed rate codes i.e., if the channel conditions deteriorate, additional redundant packets cannot be generated on the fly. Therefore to construct a reliable and robust transmission over erasure channels, a new class of rateless codes called digital fountain codes have been developed. In the following sections we will discuss Luby Transform (*LT Codes*) and *Raptor Codes*.

0.5.2 LT Codes

LT codes [24] that we describe here, are a new class of codes introduced by Luby in 1998 for the purpose of scalable and fault tolerant distribution of data over computer networks. The basic idea is as follows. The encoder can be thought of as a digital fountain that produces a continuous supply of water drops (i.e., *output packets*) called *output points*. Let's say the original file or message data consists of K packets or *input points*. The length of each input point can be arbitrary, from one-bit to $l > 1$-bits. Now, in order to completely decode the received stream, one needs to hold a bucket under the digital fountain and collects drops until the number of drops in the bucket is a little larger than K. This means that decoding does not depend on which packets were received but only on the number of received packets.

LT Codes are rateless in the sense that the number of output points that can be generated from the original data is potentially limitless and their number can be determined on the fly. Depending on the erasure probability we can send as many output points as are needed in order for the decoder to recover the original message. The original message can be recovered from any set of N output points, for N slightly larger than K, for a $[N, K]$ code.

LT codes also have very small encoding and decoding complexities. If the original data consists of K input points then each output point can be generated, independently of all other output points, on average with $O(ln(K/\epsilon))$ point operations, and the K original input points can be recovered from any N output points with probability $1 - \epsilon$, where N is $O(\sqrt{K} ln^2(K/\epsilon))$ more than K, for $\epsilon > 0$. Luby describes these codes as universal due to their being near optimal for every erasure channel and being very efficient as the data length grows.

Encoding Process

For a given set $(x_1, x_2, ..., x_K)$ of input points, the output points $(y_1, y_2, ...)$ can be generated in the following steps:

1. For each output point, randomly select a degree k from a degree distribution $\Psi(k)$ on $(1, 2, ..., K)$.
2. Select uniformly at random k distinct input points from $(x_1, x_2, ..., x_K)$.
3. Set the output point equal to the bitwise XOR-sum of these k input points.

This process can be represented by a bipartite graph, connecting the input points to the output points. Here, in order for the decoder to recover the input points from the output points, it needs to know the exact graph, i.e., which k input points have been used to generate an output point. This information has to be communicated to the receiver in some manner. It could be accomplished in any of the following ways:

- Sender and receiver can use identical pseudo-random number generators, seeded by the clock if it is synchronized.

- By a pseudo-random process which can determine the degree and the connections given a key, say ν. Then the sender generates ν to compute the output point and transmits the key in the header of the packet. If packet size is much larger than the key size (32 bits or so) then it adds only a small overhead.

Decoding Process

The decoder, having received N output points and the bipartite graph, can recover the input points in the following steps. We will call output points that have been reduced to degree one as *reduced points*.

1. Find an output point y_j from the set of reduced points that is connected to an input point, say, x_i. If Reduced Points Set (RPS) is empty, this process can not proceed further, and decoding fails.
2. Set $x_i = y_j$.
3. Add (bitwise XOR) x_i to all the output points that are connected to x_i.
4. Remove all the edges connected to the input point x_i.
5. Go to step(1) untill all $x_i's$ are determined.

The above decoding process looks simple but it turns out that the degree distribution $\Psi(k)$ plays a critical part for the success of the decoding process. The probability that RPS is not empty depends on $\Psi(k)$. Therefore, for good decoding performance we need to choose a good degree distribution.

The Degree Distribution

The degree distribution should be designed so as to meet the following design criteria:

- Most of the packets should have low degree so as to initiate the decoding process and keep the RPS from being empty.
- Some of the packets should also have high degree so as to make sure that all input points are connected to at least one of the output points.

The Ideal Soliton Distribution

In order to satisfy the above criteria an ideal soliton distribution has been proposed:

$$\Psi(k) = \begin{cases} 1/K & \text{if } k = 1 \\ 1/k(k-1) & \text{for all } k = 2, 3, \ldots, K \end{cases}$$

The motivation for the choice of the ideal soliton distribution is that at each iteration, the rate at which the input points are covered (decoded) is the same as the rate of output points joining the RPS. This prevents the decoding process from failure by making sure that at each step there are reduced points which are ready to be decoded. And in order to avoid redundancy, this distribution causes the graph to reduce the degree of output points in a way that exactly one reduced point appears at the end of each iteration. For the ideal soliton distribution the expected degree is $ln(K)$.

In practice this distribution performs poorly because a slight deviation from its expected behavior may create a situation where there is no output point with degree one, and that will cause the decoding process to stop. In order to avoid this problem the *robust soliton distribution* has been investigated.

The Robust Soliton Distribution

The robust soliton distribution denoted by $\Omega(k)$ is defined in the following manner. It is defined by two parameters, ϵ and σ. The parameter ϵ sets a bound on the probability of decoding failure and the parameter σ adjusts the RPS. The expected size of the RPS is given by

$$R = \sigma ln(K/\epsilon)\sqrt{K},$$

instead of one as in the case of ideal soliton distribution. Now we define another function

$$\Phi(k) = \begin{cases} (R/K)\frac{1}{k} & \text{for } k = 1, 2, \ldots, (K/R) - 1 \\ (R/K)ln(\frac{R}{\epsilon}) & \text{for } k = K/R \\ 0 & \text{for } k > K/R \end{cases}$$

which is added to the ideal soliton distribution $\Psi(k)$ and normalized to obtain the robust soliton distribution $\Omega(k)$ as follows:

$$\Omega(k) = \frac{\Psi(k) + \Phi(k)}{Z},$$

where $Z = \sum_{k=1}^{K}(\Psi(k) + \Phi(k))$, is the normalization factor and it determines the number of output points, N, required to recover the K input points with probability $1 - \epsilon$ where $N = ZK$. Fig. 0.14 shows the robust soliton distribution.

In [24], Luby's analysis explains how $\Phi(k)$ lets the decoding process to begin with a moderate size of RPS, but a spike at $k = K/R$ ensures that every input point has at least one connection with the set of output points. Luby's key result is that $N = K(1 + 2\Phi(K/R))$ output points are sufficient to recover original K input points with probability at least $1 - \epsilon$. The only disadvantage is that the decoding complexity grows as $O(Kln(K))$. In order to reduce the decoding complexity to $O(K)$, Raptor codes have been designed by Shokrollahi [25].

0.5.3 Raptor Codes

Raptor codes [25], first developed by Shokrollahi, are designed to achieve linear time encoding and decoding complexity. This is accomplished by first

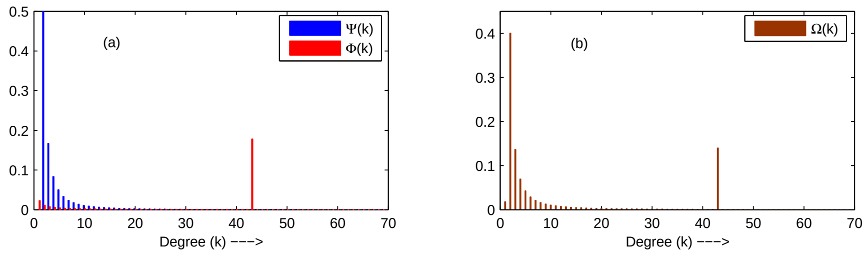

Fig. 0.14. For $K = 10000$, $\sigma = 0.2$ and $\epsilon = 0.1$, LHS figure shows the ideal soliton distribution $\Psi(k)$, and RHS shows the robust soliton distribution $\Omega(k)$

.

pre-coding the source data by an appropriate outer code to generate the input points for the LT code as shown in Fig. 0.15.

Fig. 0.15. Raptor Codes.

The main idea of pre-coding can be explained as follows. LT Codes have a complexity of the order of $ln(K)$ per packet, because the average degree per output point is $ln(K)$. Raptor codes use LT codes with average degree $\bar{k} \approx 3$. The advantage of using a lower degree is that the decoder will work almost surely and will not get stuck but there is a possibility that a fraction of the input points may not be connected to any of the output points and hence they cannot be recovered by the decoding process. Let the fraction that remains unrecovered be δ, and let the original message contain M packets. Then if we use a pre-code that generates $K = M/(1-\delta)$ packets as input points for a LT code then once slightly more than K of the output points have been received, we can recover $(1-\delta)K$ of the input points, i.e., pre-coded packets, then we can use the pre-code decoder to recover the original message as shown in Fig. 0.16. Here $M = 9$ source packets are pre-coded to generate $K = 13$

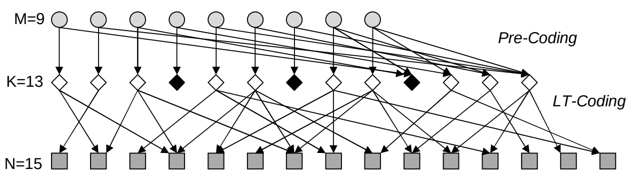

Fig. 0.16. Raptor code schematic diagram.

input points for the LT Code to generate $N = 15$ output points. Here output points have low degree and 3 of the input points have no connection with any of the output points, hence they cannot be recovered by LT decoding, so LT decoding will recover only 10 of the input points, which are used by the pre-code decoder to deduce original 9 source packets.

0.5.4 Secondary Usage of Spectrum using LT Codes

In the secondary user scenario, a SU selects a set of subchannels from PU's bands (Fig. 0.4) to establish a Secondary User Link (SUL). A SUL is a set of subchannels that adapts itself in accordance with the PU's spectral activity on that band. The SU is required to vacate the subchannel as soon as the corresponding PU becomes active on that subchannel. This forces SU to loose packets on that subchannel. To compensate for this loss caused by PU

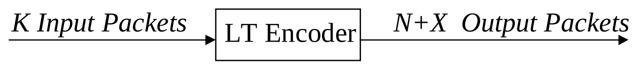

Fig. 0.17. LT Encoder.

Interference, LT Codes can be used before transmitting packets on these subchannels (Fig. 0.17). Let the SU have a message of K packets to transmit. It

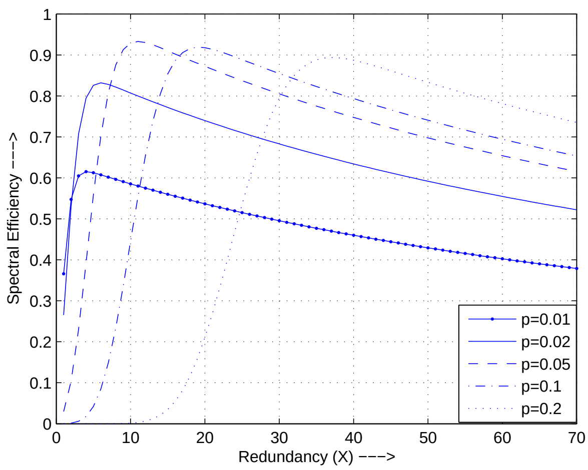

Fig. 0.18. Spectral efficiency vs redundancy with different PU arrival probability p.

uses LT codes to encode it to $N + X$ packets (as shown in Fig. 0.17), where N is the number of packets required to recover the original K packets with probability $1 - \epsilon$ and X is the redundancy required to compensate for the loss due to the interference by the PU. This in-turn depends on the PU arrival probability p. So the SUL behaves like an erasure channel with erasure probability equal to p.

Assuming that one packet is transmitted per subchannel, the total number of subchannels used by SU is $S = N + X$, and this transmission will be successful only if at most X of the subchannels are captured by the PUs. Therefore the probability of successful transmission for the SU is,

$$P_{success} = \sum_{i=0}^{X} \binom{N+X}{i} p^i (1-p)^{N+X-i} \tag{0.15}$$

Let T_p be the time required for transmission of each packet and W be the bandwidth per subchannel. Then the spectral efficiency (η) can be computed by

$$\eta = \frac{NP_{success}}{S \times W \times T_p} \tag{0.16}$$

For $W = 1/T_p$, the spectral efficiencies for $N = 100$ packets, with respect to subchannel redundancy X and PU arrival probability p are plotted in Fig. 0.18. It can be observed that for each p there is an optimal value of redundancy X which delivers the maximum spectral efficiency.

0.6 Summary

This chapter gives an overview of some game and coding theoretic approaches to dynamic spectrum access. Several game theoretic formulations of distributed dynamic spectrum access are presented. Existence of Nash and Pareto optimal solutions to these games are discussed. Stochastic learning automaton based techniques to solve these games using local information are also presented.

Secondary spectrum access is modelled as a communication over an erasure channel. Application of LT and Raptor codes for secondary user link maintenance over these channels are discussed. Some simulation results are presented to outline the trade-off between redundancy and spectral efficiency in the presence of primary users.

References

1. J. Mitola, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.,* vol. 6, no. 4, pp. 13-18, Aug. 1999.
2. K.S. Narendra and M.A.L. Thathachar, "Learning Automata: An Introduction," Englewood Cliffs: Prentice Hall, 1989
3. D. Fudenberg, J.Tirole, *Game Theory,* the MIT Press, Cambridge, 1992
4. P.S Sastry, V.V. Phansalkar, and M.A.L. Thathachar, "Decentralized Learning of Nash Equilibria in Multi-Person Stochastic Games With Incomplete Information," *IEEE Trans Systems, Man, and Cybernetics.,* vol. 24. pp. 769-777, May 1994
5. Gintis, H. "Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Behavior." Princeton University Press, 2000.
6. D. Fudenberg and J.Tirole, *Game Theory.* Cambridge, MA: MIT Press, 1991.
7. Yiping Xing, R. Chandramouli, S. Mangold and S. Sankar, "Dynamic spectrum access in open spectrum wireless networks," *IEEE Journal on Selected Areas in Communications* Special issue on 4G Wireless Systems, vol. 24, No. 3, pp. 626-637, 2006.
8. http://www.sharedspectrum.com/
9. Kolodziejski and J. Betz, "Detection of Weak Random Signals in IDD Non-Gaussian Noise," *IEEE Trans. On Comm.*, Vol 48, No. 2, February 2000.
10. Mangold, S. Challapali, K. "Coexistence of Wireless Networks in Unlicensed Frequency Bands." *Wireless World Research Forum* ♯9 Zurich Switzerland, July 2003.
11. Sai Shankar N, Carlos Cordeiro, and Kiran Challapali, "Spectrum Agile Radios: Utilization and Sensing Architectures," *IEEE DySPAN* 2005.
12. Federal Communication Commission, "Spectrum Policy Task Force," Rep. ET Docket no. 02-135, Nov. 2002.
13. Jaap C. Haartsen, Lasse Wieweg, Jorg Huschke, "Spectrum Management and Radio Resource Management Considering Cognitive Radio Systems," *XXVIIIth URSI General Assembly* Oct 2005.
14. P. Kolodzy, "Spectrum Policy Task Force: Findings and Recommendations," in *International Symposium on Advanced Radio Technologies (ISART)*, March 2003
15. Simon Haykin, "Cognitive Radio: Brain-Emprowered Wireless Communications." *IEEE JSAC.* vol. 23, No.2, Feb 2005.

16. Jon M. Peha and Sooksan Panichpapiboon, “Real-Time Secondary Markets for Spectrum.” *Telecommunications Policy*, Volume 28, Numbers 7-8, August 2004, pp. 603-618.
17. Arnon Tonmukayakul and Martin B. H. Weiss, “Secondary Use of Radio Spectrum: A Feasibility Analysis.” *TPRC*, Oct 2004.
18. J. Huang, R. Berry and M. L. Honig, “Auction-based Spectrum Sharing,” to appear in *ACM/Springer Mobile Networks and Applications Journal* (MONET)
19. S. Kiran and R. Chandramouli, “An adaptive energy efficient link layer protocol using stochastic learning control.” *Proc. IEEE Intl. Conf. on Communications (ICC).,* Alaska, 2003
20. Yiping Xing and R. Chandramouli, “Distributed discrete power control for bursty transmissions over wireless data networks,” *Proc. IEEE ICC*, pp. 139-143, vol. 1, 2004.
21. D. Willkomm, J. Gross, and A. Wolisz, “Reliable Link Maintenance in Cognitive Radio Systems,” *In Proc. of the IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN 2005)*, Baltimore (MD), November 2005.
22. D.J.C. Mackay, “Information theory, inference, and learning algorithms,” Cambridge, UK: Cambridge University Press.
23. D.J.C. MacKay, “Fountain codes,” *Communications, IEE Proceedings*, Vol. 152, Issue 6, December 2005, pages 1062-1068.
24. M. Luby, “LT codes,” *In Proc. The 43rd Annual IEEE Symposium on Foundations of Computer Science*, November 16-19, 2002 , pp. 271-282.
25. A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, No. 6, June 2006.