Erasure Tolerant Coding for Cognitive Radios

Harikeshwar Kushwaha, Yiping Xing,
R. Chandramouli, and K. P. Subbalakshmi

Media Security, Networking and Communications Lab
Department of ECE
Stevens Institute of Technology

1 Introduction

In recent years, the proliferation of spectrum-based services and devices for uses such as cellular communication, public safety, wireless LAN, and TV broadcast have forced the human society to become highly dependent on radio spectrum. This dependency and explosive growth in demand for radio resources is propelled by a host of factors: the economy has moved towards the communication-intensive service sector, the workforce is increasingly mobile, and consumers have been quick to embrace the convenience and increased efficiency of the multitude of wireless devices available today.

In the United States, the Federal Communication Commission (FCC) regulates access to spectrum. These regulations have lead to reservation of spectrum chunks for specific purposes; for example, 824-849 MHz, 1.85-1.99 GHz frequency bands are reserved for licensed cellular and personal communications services (PCS) and require a valid FCC license, whereas 902-928 MHz, 2.40-2.50 GHz, 5.15-5.35 GHz, and 5.725-5.825 GHz frequency ranges are reserved as free-for-all unlicensed bands [1]. This strict long-term spectrum allocation as shown in Fig. 1 is space and time invariant and any changes to it happen under strict FCC control.

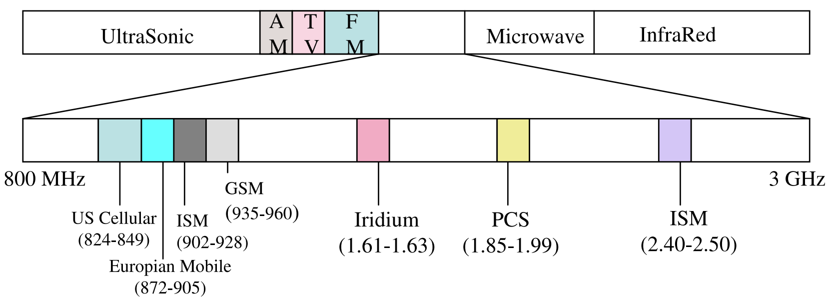

Figure 1: FCC's Static Spectrum Allocation

Considering the increase in demand for freely available, i.e., unlicensed radio spectrum, it is clear that the necessary radio spectrum will not be available in the future, due to the limited nature of radio resources in the current unlicensed frequency bands. Since many services require protection against interference, most of the radio spectrum is allocated to traditional licensed radio service. Such licensing of spectrum has led to several extreme problems uncovered by recent spectrum utilization measurements [2] and also shown in Fig 2.

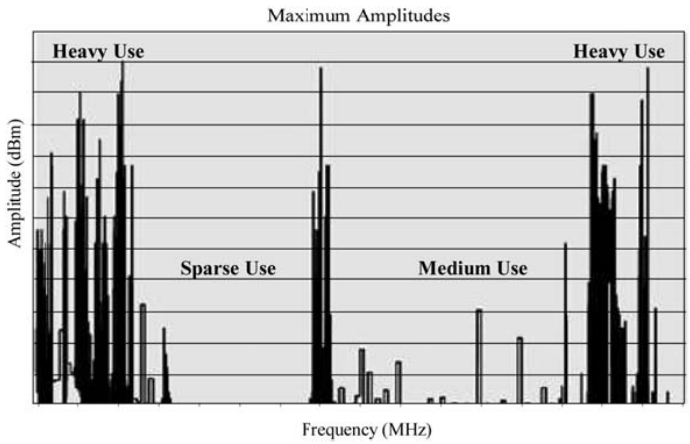

Figure 2: Spectrum Utilization Snapshot [3]

It is clear from Fig. 2 that a large portion of the allocated spectrum is highly underutilized. According to FCC [4], temporal and geographical variations in the utilization of the assigned spectrum range from 15% to 85%. Although the fixed spectrum assignment policy generally served well in the past, there is a dramatic increase in the access to the limited spectrum for mobile services in the recent years. This increase is straining the effectiveness of the traditional spectrum policies.

The limited available spectrum and the inefficiency in the spectrum usage necessitate a new communication paradigm to exploit the existing wireless spectrum opportunistically. Recent FCC Proceedings [5] propose the notion of secondary spectrum access to improve the spectrum utilization. This allows dynamic access to the unused parts of the spectrum owned by the primary license holder called primary user (PU) to become available temporarily for secondary (non-primary) user (SU). This dynamic access of spectrum by secondary users, which is facilitated by the use of cognitive radios, is one of the promising ideas that can mitigate spectrum scarcity, potentially without major changes to incumbents. Fortunately, the advances in software defined radio (SDR) [6] [7] has enabled the development of flexible and powerful radio interfaces for supporting spectral agility.

The cognitive radio devices communicate by using only the identified opportunities, without interfering with the operation of primary users. The main functions of the cognitive radios are summerized as follows:

- Detection of unused or available spectral bands
- Selection of the best available channels to meet the secondary user requirements
- Maintenance of the seamless communication requirements during transition to a different spectral band

The main objective of the cognitive radio is to obtain the best available spectrum from the temporarily unused spectrum refered to as spectrum hole. Since most of the spectrum is already licensed to primary users, the most important challenge is to share the licensed spectrum without interfering with the transmission of primary users. This sharing of spectrum is considered secondary usage of spectrum. Conceptually, primary users still owns the spectral resources and have primary access rights, however, the secondary users could use these spectral resources by guaranteeing the interference preservation of the primary users.

As proposed in [8], there are two fundamental issues regarding the secondary usage of spectrum as mentioned below:

1. Detection of PU usage by secondary users
2. Maintenance of the SU communication in case a PU appeared

The detection of PUs can be accomplished in either of the two ways namely *Negotiated Spectrum Sharing* or *Opportunistic Spectrum Sharing*.

In negotiated spectrum sharing, whenever a PU decides to make use of its spectral band, it explicitly announces its arrival and SU gets a chance to relocate to a different spectral band before PU arrives. It ensures a completely interference free communication owing to a priori declaration of all spectral claims. However, this would require the change of legacy systems in order to enable secondary usage.

On the other hand, in opportunistic spectrum sharing, the PU usage is automatically monitered by SUs with the help of cognitive radios. This monitering could be done by SUs themselves or by some centralized device which broadcasts the PU activity to SUs. The main adavantage of this approach is that no changes have to be made to legecy systems as PU is unaware of the secondary usage of its spectrum.

In this scenario, SUs have to transmit at a relatively low power level in order to protect PU transmission from interference when they arrive. As a result, the PU arrival will most likely corrupt the payload data of SU transmission on the interfered spectrum as PUs might reclaim currently used spectrum of SUs. Although PUs might capture their spectrum randomly, link maintenance becomes an issue for SUs.

The PU arrival on SU link forces them to restructure their communication link and reduces the SU system performance. In order to overcome the problems caused by PU arrival on SU link, the following techniques have been proposed in [8].

- SU link should be structured in such a way that the probability of PU appearance in the currently used spectral resources by the SU is minimized.
- SU's payload data should contain some error-correcting mechanism so as to compensate for the loss incurred due to PU arrival on part of SU link.

The first technique is implemented by dividing the bandwidth into a large number of sub-channels as shown in Fig. 3. A SU selects a set of sub-channels in such a way that they lie in different PU's bands. This minimizes the loss of sub-channels upon arrival of a particular PU as no PU can cause the complete breakdown of the SU link. This is also called Spectrum Pooling Concept. Since the arrival of a PU acts like an *erasure* for payload data on the SU link, it causes the SU to loose all the packets that are being transmitted over the sub-channel which was under that particular PU's band.

In counteracting the effect of PU interference over the SU link, the second technique of error-correction plays an important role in the link maintenance mechanism. There are two approaches to it: channel coding approach and source coding approach. The channel coding is used to compensate for the loss due to PU appearance and source coding is used to recover the content upto a certain quality depending upon the number of packets received.

The chapter is organized as follows. Section 2 discusses the spectrum pooling concept. An overview of erasure channels is presented in Section 3. Section 4 deals with traditional erasure codes and Section 5 describes digital fountain codes. Multiple description codes are discussed in Section 6. We provide some applications of these codes in Section 7. Finally, Section 8 concludes the chapter.

2 Spectrum Pooling Concept

The notion of spectrum pooling was first introduced in [10]. In this resource sharing strategy called spectrum pooling the primary user would get the highest priority. Once a primary user appears in a frequency band all secondary users transmitting in this band would have to leave immediately giving priority to the primary user.

A cognitive radio based spectrum pooling concept has been developed in [9]. A **CO**gnitive **R**adio approach for **U**sage of **V**irtual **U**nlicensed **S**pectrum (CORVUS), a vision of a cognitive radio based approach that uses allocated spectrum in a opportunistic manner to create virtual unlicensed bands i.e. bands that are shared with primary users on a non-interfering basis, has been proposed in [11]. The principles of the CORVUS system are explained below.

2.1 CORVUS System

The basic assumptions of the CORVUS system are as follows:

- There is plenty of spectrum available for sharing by secondary users

- Secondary users are capable of using cognitive radio techniques to avoid interfering with primary users if present

In this system, the SUs have to keep monitering the presence of PUs at regular intervals and as soon as a PU is found using its spectrum band, the SU must vacate that particular band and try to relocate to some other band.

Figure 3: Spectrum Pooling Concept

The principle idea of spectrum pooling concept in CORVUS is depicted in Fig. 3. This system covers a certain bandwidth W. Within this spectral range, several PUs legally own different parts of the spectrum resulting in a theoretical occupancy of the whole spectrum. However, as different PUs do not always use all their spectrum at a certain time and location this temporarily unused spectrum is available for secondary usage. SUs within this model use these temporarily available spectral resources to meet their own communication needs. In order to accomplish this the whole bandwidth is divided into N sub-channels, each with a bandwidth $w = W/N$ to form a spectrum pool.

The shaded frequency bands indicate that the PU is currently active and consequently this frequency band can not be used by any secondary user. Out of the remaining PU bands, a set of sub-channels is selected to construct a Secondary User Link (SUL) in order to satisfy the communication needs of a secondary user. A SUL is a set of sub-channels, that varies depending on the PU activity on the used sub-channels. As soon as a corresponding PU wants to make use of its spectrum, all SUs have to immediately vacate the corresponding sub-channels giving precedence to the primary user.

2.2 Reliable Communication among Secondary Users

In order to achieve reliable communication among secondary users against the loss of used sub-channels due to the PU reclaim, two means have been proposed in [8] to reduce the PU influence on SUL.

2.2.1 Sub-channel selection

In order to decrease the influence of the appearance of PUs on used spectral resources, an intelligent selection of sub-channels forming an SUL is required. Instead of selecting a contiguous set of sub-channels to form an SUL, sub-channels should be scattered over multiple PU frequency bands. Ideally, an SUL should consist of only one sub-channel per PU frequency band. This technique ensures a low effect of the PU appearance on an SUL. Since only one sub-channel is used from any PU frequency band, the payload data on only one sub-channel will be lost and only that sub-channel need to be vacated in case a PU appears.

2.2.2 New sub-channel acquisition

In order to maintain the required quality of service (QoS), the SUL needs to be updated to compensate for the loss of spectral resources due to a PU appearance. Whenever a sub-channel is lost, a reconfiguration of the SUL becomes necessary. The procedure for reconfiguration of an SUL is illustrated in Fig. 4.

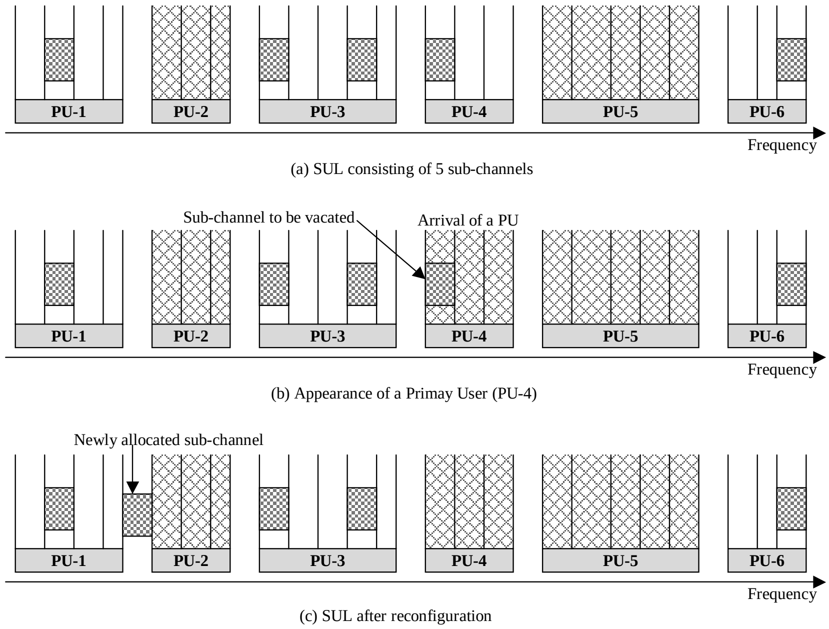

Figure 4: SUL Reconfiguration

3 Overview of Erasure Channels

The secondary user link can be modeled as an erasure channel. In erasure channels packet loss occur at the receiver due to various reasons, e.g. if the bit error correcting code fails, then the erroneous packet 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, only two possibilities are considered, i.e. a packet is either received correctly or is lost.

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

- In a secondary usage scenario, packets transmitted on a sub-channel of a SUL are lost upon reclaim of that sub-channel by a primary user.

- Packets may get discarded on the way to their destination for various reasons such as buffer overflows and congestion control at intermediate nodes.

- Packets may get corrupted due to noise and interference.

In this chapter, we mainly consider erasures due to the first reason i.e. sub-channel by a primary user. An M-ary erasure channel is shown in Fig. 5, where all input packets $\{1, 2, ..., m\}$ have a probability $1-p$ of being received correctly and a probability p of being lost (i.e. erased). If we consider the capacity of the perfect channel (i.e. $p = 0$) to be C then the capacity of the erasure channel is $(1-p)C$ irrespective of the erasure model.

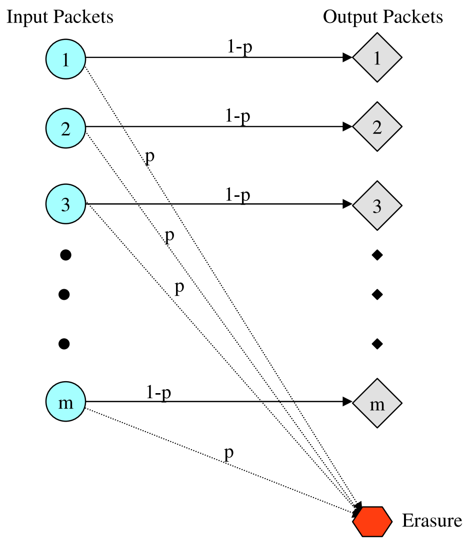

Figure 5: M-ary Erasure Channel Model

The traditional methods for communicating over such channels employ a feedback channel from the receiver to the sender that is used to control the retransmission of the erased packets. These retransmission protocols have the advantage that they work regardless of the erasure probability p but they perform poorly in the following circumstances:

- If the channel is very bad (i.e. if p is high), then retransmission protocols will introduce too much delay, thereby causing overall performance to suffer.

- In multicast or broadcast scenario, if all receivers request for retransmission then the total number of retransmissions at the sender will be very high and will decrease the overall efficiency.

- If the sender and receiver are geographically too far then the effect of propagation delay due to retransmissions may be significant.

In order to deal with the above problems of feedback-based protocols, new transmission solutions using erasure-correcting codes, that require no feedback, have been proposed in [12, 13]. So if some bits in a packet are lost or if bit 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.

4 Traditional Erasure Codes

Traditional erasure codes are typically block codes with a fixed rate, i.e. K input packets are used to generate N output packets with $N-K$ redundant packets and code rate K/N. For example, an (N, K) Reed Solomon Code [14, 15] has the ideal property that if any K of the N transmitted packets are received then the original K packets can be recovered perfectly. But this code has the following limitations:

1. In a high loss channel, for a fixed value of N, the receiver may not receive K out of N packets. In order to recover original K packets, a retransmission is necessary causing duplicate receiption of the previously received packets and thus decreasing the channel efficiency.

2. A more serious limitation is w.r.t. its encoding and decoding complexity. The reason is that, standard algorithms for encoding and decoding Reed Solomon codes require quadratic time, which are too slow for even moderate values of K. One way to deal with this problem is to encode over small blocks of data. This approach reduces the total computation, but can significantly increase the number of packets that must be obtained before complete decoding.

Another code is a class of LDPC codes called Tornado Codes. Tornado codes are a class of erasure-correcting codes that have linear encoding and decoding times in N. Software-based implementations of Tornado codes are about 100 times faster on small lengths and about 10,000 times faster on larger lengths than software-based Reed-Solomon erasure codes while having only slightly worse overhead i.e. slightly more than K packets need to be received to successfully recover original K packets. In Reed Solomon code, every redundant packet (constraint) depends on every input packet and this increases encoding and decoding complexity. On the other hand, in Tornado codes, each constraint depends only on a few input packets, on an average, only a constant number of operations are required to generate each redundant packet (or constraint). This makes Tornado codes linear and have fast encoding and decoding times.

Even if Tornado codes have fast encoding and decoding complexity, they still suffer from a major drawback. Since it is a fixed rate code, the value of N

must be fixed for a given value of K ahead of time. And if the channel erasure probability p is large then the receiver may receive more than K packets which is required in order to recover the original K packets. Unfortunately, there is no way to extend this code on the fly to generate more output packets as the demand arise.

Subsequent works [15, 16]in this field has led to codes that overcome the above problems by developing *rateless codes*, also called *Digital Fountain Codes.* These codes are discussed in the next section.

5 Digital Fountain Codes

In this section, we discuss a new class of erasure-correcting codes, called digital fountain codes [13, 17], that can be used to provide protection against erasures caused by a primary user appearance over a secondary user link. These codes are capable of providing protection from the effects of packet loss irrespective of the loss model of the secondary user link. By recovering lost data packets without requesting retransmission from the sender, these codes provide reliability in various network applications [14] such as multicast, parallel downloading, video streaming etc. 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.

In the following sections, we discuss Luby Transform Codes (*LT Codes*) and *Raptor Codes.*

5.1 LT Codes

LT codes [15, 12, 18] are the first practical realization of a rateless code. These codes were introduced by Luby in 1998 for the purpose of scalable and fault tolerant distribution of data over computer networks. It works on the following fundamental principle: The encoder can be thought of as a digital fountain that produces a continuous supply of water drops called encoded packets or *output packets.* Let's say the original file or message data consists of K source packets. The length of each input packet 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 slightly 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 packets that can be generated from the original data is potentially limitless and their number can be determined on the fly. The general idea behind the rateless codes is the following: given the data block (file) divided into K equal length packets of length l bits, generate infinite number of encoded packets of the same length l, such that any set containing a little more than K randomly selected encoded packets is sufficient for file reconstruction. This means that the receiver is needed just to collect slightly more than K packets, irrespective of its ordering, to successfully recover the original file.

LT codes also have very small encoding and decoding complexities. If the original data consists of K input packets then each output packets can be generated, independently of all other output packets, on an average with $O(ln(K/\delta))$ operations, and the K original input packets can be recovered from any N output packets with probability $1-\delta$, where N is $O(\sqrt{K}ln^2(K/\delta))$ more than K, for $\delta > 0$. 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.

5.1.1 LT Encoder

The task of the encoder is to generate an infinite stream of encoded packets given the file to be transmitted. We assume that the original file is segmented into K packets of length l bits each. Let the source packets be $(x_1, x_2, ..., x_K)$ then the encoded packets $(y_1, y_2, ...)$ can be generated in the following manner:

1. A random degree d is selected according to a degree distribution $\rho(d)$ on $\{1, 2, ..., K\}$.
2. d out of K packets are uniformly selected at random from the original source packets $(x_1, x_2, ..., x_K)$.
3. The d selected packets are added using bitwise xor operation to produce an encoded packet
4. Steps 1-3 are repeated to produce new encoded packets

This process is represented with a bipartite graph as shown in Fig. 6, connecting the source packets to the encoded packets. When transmitting packets using a traditional erasure code both the sender and receiver are aware of the exact description of the encoding methods used. This is not the case with rateless codes, since the code is being generated concurrently with the transmission. Therefore, in order to be able to recover the source data from the encoded packets, it is necessary to transmit a description of the code structure together with the encoded packets. In other words, the receiver needs to know the exact graph

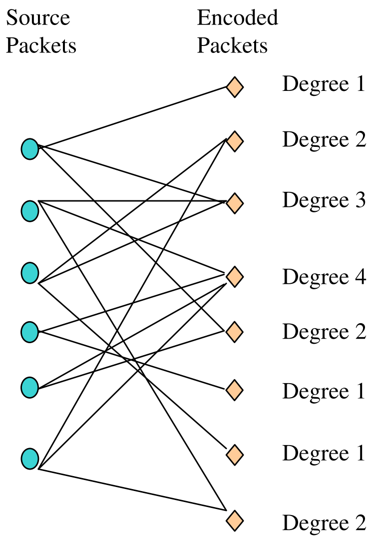

Figure 6: A bipartite graph representing LT encoding

i.e. which k source packets (but not their values) have been used to generate a particular output packet. it could be accomplished in any of the folowing manner [15]:

- Sender and receiver can use identical pseudo-random number generators, seeded by a clock if it is synchronized.
- By a pseudo-random process which can determine the degree and the connections given a key, say κ. Then the sender generates κ 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.

5.1.2 LT Decoder

After having received a slightly larger than K encoded packets and the bipartite graph, the decoder tries to recover the source packets by message passing algorithm. The LT decoding process can be described as follows:

1. From the set of received encoded packets, search for a packet y_i that is connected to only one source packet x_j. If there is no such encoded packet then this decoding algorithm can not proceed further and fails to recover all the source packets.
2. Copy the value associated with the selected degree-one node to its only neighboring source packet, i.e. set $x_j = y_i$. Remove this degree-one output packet (y_i).

3. Add (modulo 2) x_j to all the encoded packets that are connected to this source node (x_j).
4. Remove all edges connected to the source packet (x_j).
5. Repeat steps 1-4 until all x_i's have been determined.

This is a simple decoder, but it turns out that the degree distribution function, $\rho(d)$, is a critical part of the LT encoder and decoder design. The decoding process will fail at step-1 if there is no output packet of degree one. Therefore, for good decoding performance, an appropriate selection of degree distribution is required.

5.1.3 Degree Distributions

There are two fundamental tradeoffs in the design of degree distribution:

- At the end of each iteration of the decoding process, the set of output packets that have degree one called *ripple* should be small to avoid redundancy in encoded packets. On the other hand, if the size of the ripple is too small and drops to zero at any iteration then decoding process halts at that point.
- The degree of encoded packets should be small to reduce the decoding complexity as it depends on the number of edges in the bipartite graph. On the other hand, if most of the encoded packets have small degree then there is a high probability that some of the source packet may not be connected to any of the output packets, thereby they can't be recovered.

In the following sections we discuss *Ideal Soliton Distribution* and *Robust Soliton Distribution* as proposed by Luby in [15].

5.1.4 Ideal Soliton Distribution

A good degree distribution should have the property that the ripple size never gets too small or too large. In order to satisfy the above criteria an ideal soliton distribution has been proposed as follows:

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

This distribution displays ideal behavior in terms of the expected number of encoded packets required to recover the data. In this distribution we need at least $K\,ln(K/\delta)$ edges in our bipartite graph. The decoding operation uses all the edges at least once. The overall running time for the decoding is $O(K\,ln(K)$. For this distribution the expected degree is $ln(K)$.

This distribution behaves well in expection. However it turns out that this is not a good distribution to use in reality. In practice a slight deviation from its expected behavior may create a situation where there is no output packet with degree one, and this will cause the decoding process to halt. In order to avoid this problem, the robust soliton distribution has been discussed.

5.1.5 Robust Soliton Distribution

While the ideal soliton distribution is optimal in some ways, it performs rather poorlyin practice. However, it can be modified slightly to yield the robust soliton distribution denoted by $\omega(d)$ and defined in the following manner. It has two parameters, σ and δ. It is designed to ensure that the expected number of degree-one output packets is about

$$S = \sigma ln(K/\delta)\sqrt{K} \tag{2}$$

rather than one throughout the decoding process. The parameter δ is a bound on the probability that the decoding fails to run to completion. The parameter σ adjusts the size of the ripple (S). Now define

$$\tau(k) = \begin{cases} (S/K)\frac{1}{d} & \text{for} \quad d = 1, 2, \ldots, (K/S) - 1 \\ (S/K)ln(\frac{S}{\delta}) & \text{for} \quad d = K/R \\ 0 & \text{for} \quad d > K/R \end{cases} \tag{3}$$

Add the ideal soliton distribution $\rho(d)$ to $\tau(d)$ and normalize to obtain $\omega(d)$ as follows:

$$\omega(d) = \frac{\rho(d) + \tau(d)}{\beta} \tag{4}$$

where $\beta = \sum_{d=1}^{K}[\rho(d) + \tau(d)]$ is the normalization constant chosen to ensure that $\omega(d)$ is a probability density function. This constant β also determines the number of encoded packets, N, required to recover the original K source packets with probability $1 - \delta$ where $N = \beta K$. These distributions have been shown in Fig. 7.

Luby's analysis [15] explains how the small-d end of τ has the role of ensuring that teh decoding process gets started, the spike in τ at $d = K/S$ ensures that every input packet is likely to be connected to an output packet at least once. Luby's key result is that receiving $K + 2ln(S/\delta)$ output packets ensures that all input packets can be recovered with probability at least $1 - \delta$. The only disadvantage is that the decoding complexity grows as $O(Kln(K))$, but it turns out that such a growth in complexity is necessary to achieve capacity. However, slightly sub-optimal codes with decoding complexity $O(K)$ called raptor codes have been developed by Shokrollahi in [16].

5.2 Raptor Codes

Raptor codes [16] extend the idea of LT codes one important step further. LT codes suffer in that an averge degree of $O(ln(K))$ is needed to cover every source packet with high probability. As a result, the decoding complexity also increases as K becomes large. Raptor codes are designed to achieve linear time encoding and decoding complexity. These codes accomplish this task by first pre-coding the source packets by an appropriate fixed erasure code, say some LDPC code,

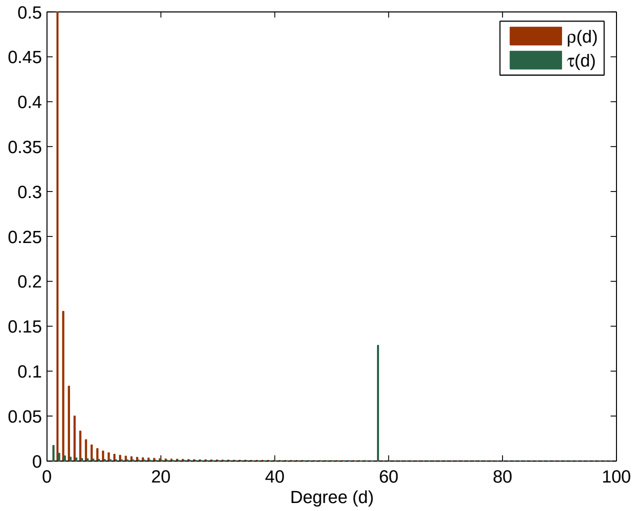

Figure 7: The distributions $\rho(d)$ and $\tau(d)$ for $K = 10000$, $\sigma = 0.15$ and $\delta = 0.1$

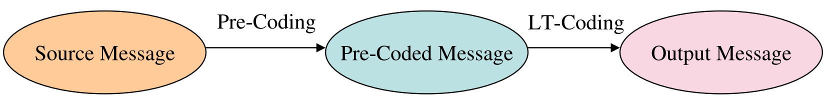

Figure 8: Raptor Codes

before passing it to LT encoder as shown in Fig. 8.

Let the original message be M, and the pre-encoded version of the message be M' then this M' becomes the input to the LT encoder. Now, since we don't need to recover every packet of M' in order to recover M, but just a constant fraction $1 - \epsilon$ of the number of packets in M'. The main idea of the 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 packet is $ln(K)$. Since raptor codes use a pre-code, all input packets of the LT encoder need not be covered during encoding with LT encoder. So Raptor codes use LT codes with very small average degree which is generally constant and that makes the encoding and decoding linear in the number of packets.

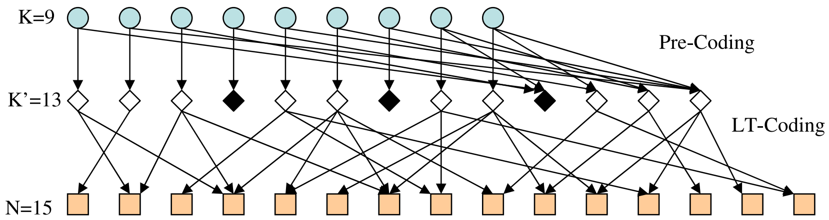

Figure 9: Raptor Code Schematic showing the uncovered input packets by a low degree LT code

For example, let the original message contains K packets, and after pre-coding message M' has K' packets. And suppose a fraction ϵ of K' remains uncovered during encoding with LT codes. Then if $K' \geq K/(1-\epsilon)$ then once a slightly more than K' of the output packets have been received, we can recover the original message M as shown in Fig. 9. In this figure, $K = 9$ source packets are precoded to generate $K' = 13$ input packets for LT code, which then produces output packets. Once $N = 15$ output packets have been received, we can see that 3 of the input packets have not been covered so far by this low degree LT encoder but still 10 of the input packets can be recovered and then the original message can be obtained from pre-code decoder.

Raptor codes currently offer the best approximation to a digital fountain code. Some analysis of LT codes and raptor codes are presented in [19], [20].

6 Multiple Description Codes

Multiple description coding (MDC) is a recent source coding approach that is gaining popularity as a viable coding mechanism for delivery of multimedia data over networks: especially because networks typically have multiple paths between the transmitter and the receiver. In such a scheme, several representations of the source, called descriptions, are generated such that the quality of the received signal is proportional to the *number* of descriptions received and not which descriptions were actually received [24], [25].

Such a coding scheme is well suited to packet networks and fading wireless channels. Unlike layered coding however, the quality of the received video does not depend on *which* descriptions are actually received only on how many of them are received and hence one can consider MDC as a generalization of the layered coding.

MDC is a form of data partitioning, thus comparable to layered coding as it is used in MPEG-2 and MPEG-4. Yet, in contrast to MDC, layered coding mechanisms generate a base layer and n enhancement layers. The base layer is necessary for the media stream to be decoded, enhancement layers are applied to improve stream quality. However, the first enhancement layer depends on the base layer and each enhancement layer $n+1$ depends on its subordinate layer n, thus can only be applied if n was already applied. Hence, media streams using the layered approach are interrupted whenever the base layer is missing and, as a consequence, the data of the respective enhancement layers is rendered useless. The same applies for missing enhancement layers. In general, this implies that in lossy networks the quality of a media stream is not proportional to the amount of correct received data. A general example of a MDC system with two descriptions is given in Fig. 10.

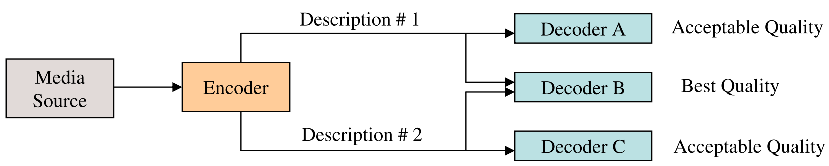

Figure 10: An example of MDC system with two descriptions

In a secondary usage scenario, this type of coding could be useful in recovering the original multimedia to a certain quality. If a large number of sub-channels are captured by PUs then the channel coding schemes discussed in the previous sections may not be able recover enough number of packets or it may take unacceptably longer time to receiver sufficient number of packets for LT decoding process to start. But if the original media is converted into descriptions with MDC then we can still construct the image upto a certain quality depending upon the number of descriptions received.

Some MDC video coding algorithms have been discussed in [25], [26].

7 Applications

In this section, we discuss the secondary spectrum access with LT codes based on [21]. In the secondary usage scenario, a SU selects a set of sub-channels from the PU bands as discussed in section 2. The SU is required to vacate the sub-channel as soon as the corresponding PU becomes active on that sub-channel. This forces the secondary user to loose packets on that sub-channel. To compensate for the loss caused by the PU appearance, the source packets are encoded with LT codes. Let the secondary user has a message M of K packets to transmit. And let the LT decoder needs at least N packets in order to recover origial K packets with probability $1-\delta'$. Then in order to compensate for the loss due to PU appearance, we add some more redundancy, say X, which depends on PU arrival probability p. If PU arrival is quite frequent then we need to use high value of X, if PU arrives occasionally then even a small value of X is sufficient.

As discussed in [8], let one packet is transmitted per subchannel, then the total number of sub-channels used by SU is $Q = N + X$, and this transmission will be successful only if at most X of the sub-channels are lost due to PU appearance. Therefore, the probability of successful transmission for the secondary users is given by

$$P_{success} = \sum_{i=0}^{X} \left(\begin{array}{c} N+X \\ i \end{array} \right) p^i (1-p)^{N+X-i} \tag{5}$$

Let T_{frame} be the time required for transmission of each packet and W be the bandwidth per subchannel. Then the spectral efficiency of the secondary user link (η) can be computed by

$$\eta = \frac{(1-\delta) N P_{success}}{S \times W \times T_{frame}} \tag{6}$$

The LT decoding error probability δ', evaluated via simulation for robust soliton distribution with parameters $\sigma = 0.1$ and $\delta = 0.5$ is shown is Fig. 11 for different values of K and redundancy factor $\chi = X/K$.

As seen in Fig. 11, LT decoding error is very high for small values of redundancy, and after some value of χ, it becomes very small and almost constant.

So if a large number of sub-channels are captured by PUs, then the total received overhead χ will be small and will result in a high LT decoding error probability. But if source packets are converted into descriptions with multiple description coding then even with a small overhead, the original data can be recovered upto a certain quality.

Various applications of fountain codes in video streaming, broadcast, multicast, and parallel downloading are presented in [14, 22, 23].

8 Conclusion

This chapter gives an overview of some coding aspects of dynamic spectrum access with cognitive radios. Spectrum pooling concept is presented for secondary usage of spectrum. The loss on a secondary user link is modeled as an erasure and erasure tolerant coding has been proposed to compensate for it.

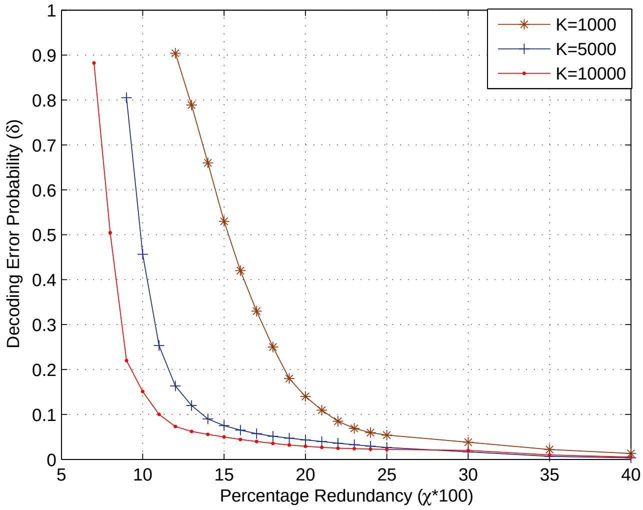

Figure 11: LT Decoding Error Probability

Traditional erasure codes such as Reed Solomon codes and Tornado codes have been discussed as fixed erasure codes. Then digital fountain codes such as LT codes and raptor codes have been presented as rateless codes. Multiple description codes have been discussed as a source coding approach for robust communication in secondary usage unreliable scenario. Their applications in secondary spectrum access with cognitive radios have been discussed. Some simulation results on LT codes decoding error probability are also presented.

References

[1] M.M. Buddhikot, P. Kolody, S. Miller, K. Ryan, J. Evans, "DIMSUMNet: new directions in wireless networking using coordinated dynamic spectrum access," *In Proc. IEEE WoWMoM 2005*, June 2005, pp. 7885.

[2] M. McHenry, "Dupont Circle Spectrum Utilization during Peak Hours," *The Shared Spectrum Company*, 2002.

[3] I. F. Akyildiz, W. Lee, M. C. Vuran, S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks Journal (Elsevier)*, Vol. 50, pp. 2127-2159, September 2006.

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

[5] Federal Communications Commission (FCC), "Spectrum Policy Task Force," *ET Docket no. 02-135*, November 15, 2002.

[6] J. Mitola, "The software radio architecture." *IEEE Communications*, Vol.33, No.5, 1995, pp.26-38.

[7] "White paper on regulatory aspects of software defined radio", *SDR forum document number SDRF-00-R-0050-v0.0.*

[8] 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.

[9] T. Weiss and F. Jondral, "Spectrum pooling: An innovative strategy for the enhancement of spectrum efficiency," *IEEE Commun. Mag.*, vol. 42, pp. S8S14, march 2004.

[10] J. Motola, "Cognitive Radio for Flexible Mobile Multimedia Communications," *IEEE International Workshop on Mobile Multimedia Communications*, November 1999.

[11] D. Cabric, S. M. Mishra, D. Willkomm, R. W. Broderson, and A. Wolisz, "A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum," *In Proc. of 14th IST Mobile Wireless Communications Summit 2005*, Dresden, Germany, June 2005.

[12] D.J.C. Mackay, "Information theory, inference, and learning algorithms," *Cambridge, UK: Cambridge University Press.*

[13] D.J.C. MacKay, "Fountain codes," *Communications, IEE Proceedings*, Vol. 152, Issue 6, December 2005, pages 1062-1068.

[14] M. Mitzeumacher, "Digital Fountains: A Survey and Look Forward," *IEEE Information Theory Workshop*, San Antonio, Texas, October 2004

[15] M. Luby, "LT codes," *In Proc. The 43rd Annual IEEE Symposium on Foundations of Computer Science*, November 16-19, 2002 , pp. 271-282.

[16] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, No. 6, June 2006.

[17] "DF Raptor™ Technology: Advanced FEC Technology for Streaming Media and Data Distribution Applications," *A white paper from www.DigitalFountain.com.*

[18] D. Vukobratovic, M. Despotovic, "On the Packet Lengths of Rateless Codes," *In EUROCON 2005 - The International Conference on Computer as a tool*, Belgrade, Serbia & Montenegro, November 2005.

[19] R. Karp, M. Luby, and A. Shokrollahi, "Finite Length Analysis of LT Codes," *in Proceedings of ISIT 2004*, p. 37.

[20] A. Shokrollahi, "Raptor Codes," *in Proceedings of ISIT 2004*, p. 36.

[21] H. Kushwaha, and R. Chandramouli, “Secondary Spectrum Access with LT Codes for Delay-Constrained Applications,” *IEEE Consumer Communications and Networking Conference*, January 2007.

[22] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, W. Xu, “Raptor Codes for Reliable Download Delivery in Wireless Broadcast Systems,” *In Proc. Consumer and Communications Networking Conference (CCNC 2006)*, Las Vegas, NV.

[23] J. Wagner, J. Chakareski and P. Frossard, “Streaming of Scalable Video from Multiple Servers using Rateless Codes,” *In Proceedings of IEEE International Conference on Multimedia and Expo 2006*, July 2006

[24] V. K. Goyal, “Multiple Description Coding: Compression Meets the Network,” *IEEE Signal Processing Magazine*, vol. 18, no. 5, pp. 74 – 94, Sept. 2001.

[25] R. Puri and K. Ramchandran, “Multiple description source coding using forward error correction codes,” *IEEE Proceedings Asilomar Conference on Signals, Systems, and Computers*, Asilomar, CA, October 1999.

[26] S. Somasundaram and K. P. Subbalakshmi, “3-D Multiple Description Video Coding for Packet Switched Networks,” *IEEE International Conference on Multimedia and Expo*, Baltimore, Maryland, July 2003.