Fountain Codes for Frequency Occupancy Information Dissemination

Xiaoying Shao, Harm Cronie, Fokke Hoeksema and Kees Slump
University of Twente, Faculty of EEMCS,
Signal and System Group,
P.O. box 217 - 7500 AE Enschede - The Netherlands
Phone: +31 53 489 4570 Fax: +31 53 489 1060
E-mail: {x.shao, h.s.cronie, f.w.hoeksema, c.h.slump}@ewi.utwente.nl

***Abstract*— Cognitive radio (CR) is defined as an intelligent wireless communication system based on secondary utilization of an already licensed frequency band. In order to communicate without interfering the legal users (primary users), cognitive radio nodes should have the same overview of the spectrum occupancy information. In other words, each cognitive radio node should disseminate frequency occupancy information (FOI) to other CR nodes reliably after sensing the environment.**

Normally, people employ retransmission protocols to communicate reliably. This leads to redundancy in reliable dissemination, especially in a multicasting or broadcasting situation. However, the application of *fountain codes* could make reliable dissemination possible without too much redundancy.

A fountain-code based approach is more efficient than using retransmission protocols for disseminating *large*-sized file. In the paper we show this is also the case when applying fountain codes in disseminating a *small* FOI file (e.g. a binary vector originating from a 512 points FFT).

Fountain codes only work optimally in an *erasure channel*. In the paper we show the utilization of error-correcting codes to convert a noisy channel into an erasure channel.

Key Words: Cognitive networking, Frequency occupancy dissemination, fountain code, error-correcting code.

I. Introduction

The *Adaptive Adhoc Free band Wireless communications* (AAF) project [1] focuses on designing a robust emergency communication system by using the techniques of cognitive radio, ad hoc networking and adaptive OFDM modulation. The cognitive radio is defined as an intelligent wireless communication system, which can be aware of the environment and uses the unoccupied frequency band to communicate. In order to make this wireless communication system reliable, every cognitive radio node should have the same view of the spectrum usage, which means each AAF node should have the same knowledge of frequency occupancy.

How can all AAF nodes have the same overview of the spectrum? This is a predominant question for the AAF project, also for cognitive radio. In the AAF project, each node will scan the spectrum between $400 - 1000$ MHz in some fixed time interval, then disseminate its frequency occupancy information (FOI) to other nodes reliably. After every node has received the FOI packages from other nodes, it will combine all the FOI packages and get the overview of the spectrum occupancy.

Normally, standard file-transfer protocols segment a file into many small packets, then repeatedly transmit each packet until it is received successfully. So, a feedback channel is needed for the transmitter to know which packets should be retransmitted. However, these traditional protocols do not work efficiently for reliable broadcasting, especially in the unreliable wireless channel. For example, if a file is broadcasted over an erasure channel, and every packet that is missed by one or more receivers has to be retransmitted according to the standard retransmission protocols, those retransmissions will be terribly redundant. Every receiver may receive retransmitted packets that it has already received.

So, we like to find a solution to broadcast reliably without too much redundancy in the wireless channel. Fountain codes satisfy the above requirements if we use error-correcting codes to convert the noisy wireless channel into an erasure channel [3]. The transmitter sprays the encoded packets as many as needed, and the receivers can decode the encoded packets as soon as they receive enough different encoded packets. The probability of generating a same encoded packet as one of previous encoded packets is very small. Therefore, Whether the receivers can recover the source data is related to the *number* of received packets, but has little to do with particular individual packets.

In this paper, we discuss how to apply fountain codes in cognitive networking to broadcast the FOI information.

II. Fountain Codes

A. The Random Linear Fountain

Consider a file of size K packets $s_1, s_2, ..., s_{\mathrm{K}}$ to be encoded by random linear fountain codes [3]. A 'packet' here is considered as an elementary unit that is either transmitted intact or erased by the erasure channel. We suppose that a packet is composed of a number of bits and each packet is chosen by the encoder with the probability of 0.5.

The idea of this encoder is: at each time instance, labeled by n, the encoder chooses randomly several packets, and computes the bitwise sum of these source packets to generate the corresponding transmitted packet. The number of the

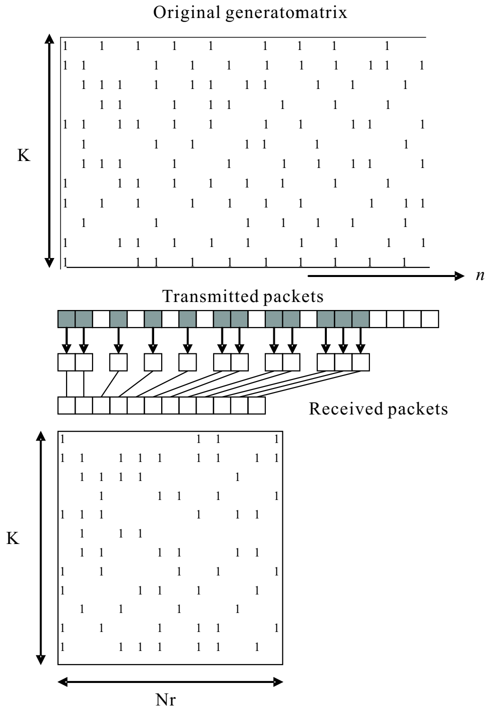

Fig. 1. The generator matrix of a random linear code (top). When the packets are transmitted ones, some are received, shown by the grey shading of the packets and the corresponding columns in the matrix. We can realign the columns to define the generator matrix, from the point of view of the receiver (bottom) [3].

selected packets is also random. In order to mark which packets are selected in the n-th time instance, the encoder generates K bits $\{G_{kn}\}$, in which '1' indicates the selected source packets, and '0' means these unselected source packets.

This idea can also be described in the mathematical way, which is shown in the following equation [3]:

$$t_n = \Sigma_{k=1}^{K} s_k G_{kn}. \tag{1}$$

in which t_n is the encoded packet at the nth time instance and G is the generator matrix. This sum can be done by exclusive-or-ing the packets together. For the set of K random bits at each time instance, we consider that set as a new column in an ever growing generator matrix, as shown at the top of figure 1 [3].

In order to recover the source data, we need to know the generator matrix, whose columns are corresponding to these received packets. We could assume the matrix G was generated by a deterministic random-number generator, and the receiver has an identical generator that is synchronized to the encoder's [3]. Alternatively, the sender could pick up a random key, k_n, which is used to generate the set of K random bits at each clock cycle. The random key is put in the head of the transmitted package, and the receivers could also generate the same set of K random bits according to this random key. After receiving N_r packets, these receivers could get the corresponding generator matrix G, which is used to decode. Decoding is possible when G is full rank.

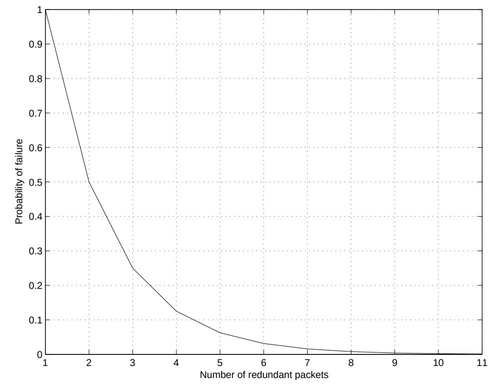

Fig. 2. Performance of the random linear fountain.

But, what is the minimum value of N_r? Of course, N_r should not be smaller than K, because the receiver do not have enough information to recover the source data. If $N_r = \mathrm{K}$, it's possible that the receiver can recover the file if the K-by-K matrix G is invertible (modulo 2). The source packages could be recovered by applying

$$s_k = \Sigma_{n=1}^{N_r} t_n \mathrm{G}_{nk}^{-1} \tag{2}$$

According to [3], the probability of the invertibility of a random K-by-K binary matrix is

$$(1-2^{-\mathrm{K}})(1-2^{-\mathrm{K}-1}) \times \cdots \times (1-\frac{1}{8})(1-\frac{1}{4})(1-\frac{1}{2}),$$

which is 0.289, for any K larger than 10. That's not great, because our expected value is close to 1.

What about if N_r is slightly larger than K? Let $N_r = \mathrm{K} + \mathrm{E}$, where E is the small number of redundant packets. The probability that the random K-by-N_r binary matrix G contains an invertible K-by-K matrix is $1-\delta$, where δ is the probability that the receiver will not be able to decode the file. For any K, the probability of failure is bounded above by

$$\delta(\mathrm{E}) \leq 2^{-\mathrm{E}}. \tag{3}$$

This bound is shown in figure 2. From this figure, we see that the failure probability of decoding random linear fountain codes is close to 1 if the number of redundant packets E is equal to or larger than 10 . The only disadvantage of random linear fountain codes is that their encoding and decoding costs are quadratic and cubic in the number of the source packets. [3]

B. LT Codes

LT (Luby Transform) codes invented by Michael Luby are called fountain codes, which retains the good performance of the random linear fountain codes, but drastically reducing the encoding and decoding complexities.

1) Encoder: First, we give the definition of degree we use in the following sections. Degree is defined as the amount of selected source packets to form an encoded packet. [6] shows that a file of size K packages $s_1 s_2 ... s_\mathrm{K}$ can be encoded by the LT codes according to the following three steps:

1) Design the degree distribution $\rho(d)$, which depends on the source file's size K, as we'll discuss in section II-B.3.
2) Randomly choose the degree d_n of the packets from $\rho(d)$.
3) Choose, uniformly at random, d_n distinct input packets, and set t_n equal to the bitwise sum, modulo 2 of those d_n packets. This sum can be done by successively exclusive-or-ing the packets together.

The idea of the LT encoder is almost the same as the random linear fountain encoder. The only difference is that we should design the degree distribution in the LT encoder.

2) Decoder: In order to recover $\boldsymbol{s}$ from $\boldsymbol{t} = \mathrm{G}\boldsymbol{s}$, where G is associated with the tanner graph, the simple way to decode this sparse-graph code is by message-passing. The idea of this decoding process is as follows:

1) Find an encoded packet t_n that is only connected to one source packet s_k. If there is no such encoded packet, this decoding algorithm stops at this point, and fails to recover all the source packets.
   a) Set $s_k = t_n$.
   b) Add s_k to all encoded checks $t_{n'}$ that are connected to s_k:
   $$t_{n'} := t_{n'} + s_k \qquad \text{for all } n' \text{ such that } G_{n'k} = 1. \tag{4}$$
   c) Remove all the edges connected to the source package s_k.
2) Repeat step 1 until all source packets are determined.

3) Design the Degree Distribution: The probability distribution $\rho(d)$ of the degree d is an important part of the design. Good fountain codes should have the following characteristics:

- Some encoded packets must have high degree in order to ensure that every source packet is connected to at least one encoded packet.
- Many packets must have low degree, so that the decoding process could start, keep going, and so that the total number of addition operations involved in the encoding and decoding is kept small.

[6] shows that the robust soliton distribution satisfies the above requirements and works very well in practice.

The expected number of degree-one encoded packets is about
$$S \equiv c \ln(\mathrm{K}/\delta)\sqrt{\mathrm{K}}, \tag{5}$$
rather than 1, throughout the decoding process. The parameter δ is a bound on the probability that the decoding fails after N_r packets have been received. The parameter c can be viewed as a free parameter, with a value smaller than 1 giving good results (e.g. $c = 0.05$). The ideal soliton distribution is defined as follows:
$$\rho(d) = \begin{cases} 1/\mathrm{K} & d = 1 \\ \frac{1}{d(d-1)} & \text{for } d = 2, 3, \ldots, \mathrm{K} \end{cases} \tag{6}$$

Besides, a positive function is defined as:
$$\tau(d) = \begin{cases} \frac{S}{\mathrm{K}}\frac{1}{d} & \text{for } d = 1, 2, \ldots, (\mathrm{K}/S) - 1 \\ \frac{S}{\mathrm{K}} \ln(S/\delta) & d = \mathrm{K}/S \\ 0 & \text{for } d > \mathrm{K}/S \end{cases} \tag{7}$$

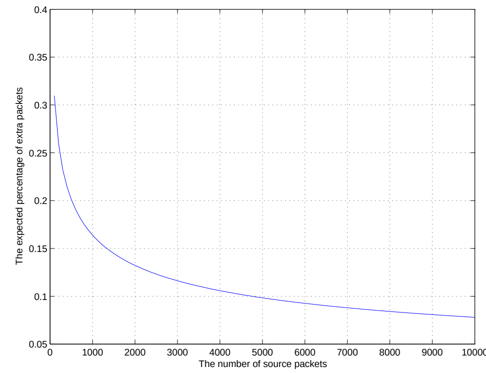

Fig. 3. Practical performance of LT codes. The parameters in this case are $c = 0.05$, $\delta = 0.05$.

Then, we could obtain the robust soliton distribution μ by adding the ideal soliton distribution ρ to τ and normalizing, which is
$$\mu(d) = \frac{\rho(d) + \tau(d)}{Z}, \tag{8}$$
where $Z = \sum_d [\rho(d) + \tau(d)]$. The expected number of the encoded packets that are required by the receivers to recover the source packets with probability at least $1 - \delta$, is given by [13]:
$$N_r = \mathrm{K}Z. \tag{9}$$

Figure 3 shows the performance of the LT codes according to (9). From this figure, we see that the expected percentage of extra encoded packets is inversely proportional to the number of source packets, which means LT codes are more efficient for large number of packets.

III. Channel Conversion

As we mentioned before, fountain codes are designed for the erasure channel. But, no real-world wireless channels are erasure channels, and normally, such that noisy channel could be modeled as an AWGN channel or a Rayleigh fading channel [9]. In order to convert a noisy channel to an erasure channel, we should use a special code, which is called error-correcting codes. LDPC codes can be employed to render the erasure channel from the noisy channel. In [11], the author mentions that the increased row weight increases the undetected error rate, which leads to an increase in the probability of failure by implementing FC to disseminate the FOI packets. CRC is used as an error detection code. In this paper, we combine the LDPC codes and Cyclic Redundancy Check (CRC) to create the erasure channel from the noisy channel.

Normally, regular LDPC codes are constructed at random subject to these constraints and can be largely computer generated. However, this kind of LDPC codes are good for long codes.

We assume the FOI packet to be rather short [1], and we could not use the above way to encode them. In [7], a geometric

[1] e.g. a result of a 512 FFT, following a hand decision, resulting in a 512-bit FOI packet

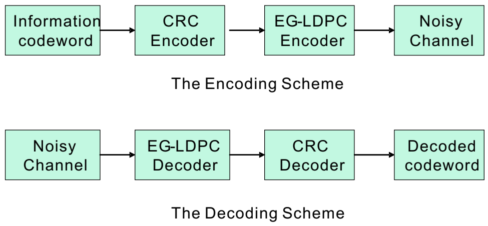

Fig. 4. The encoding and decoding scheme of the error-correcting codes.

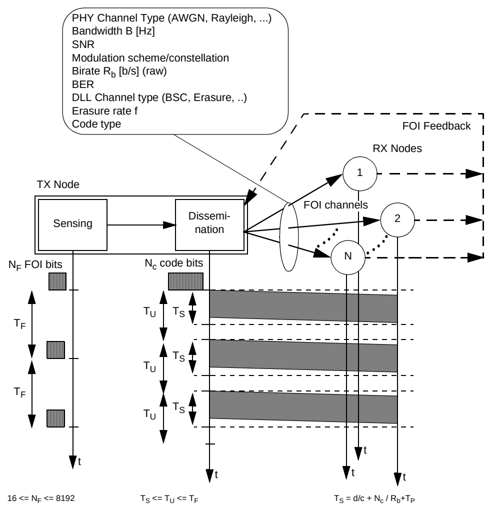

Fig. 5. FOI Dissemination System Model

.

approach is presented to the construction of short LDPC codes, which could be used to encode the FOI packets. The construction of this kind of LDPC codes is based on the lines and points of the Euclidan Geometry (EG), so they are known as the EG-LDPC codes.

The basic idea behind CRC is to treat the message string as a single binary word M, and divide it by a key word k that is known to both the transmitter and the receiver. The remainder r left after the division constitutes the 'check word' for the given message. The transmitter sends both the message string M and the check word r, and the receiver can then check the data by repeating the calculation, dividing M by the key word k, and verifying that the remainder is r. These calculations are done in the form of the code polynomials. The key word k could be in the form of a 'generator polynomial' $g(x) \in \mathrm{GF}(2^x)$ whose coefficients are the binary bits of the number k. In general, a polynomial with k bits leads to a '$k-1$-bit CRC' [12].

After combining these two codes, the encoding and the decoding schemes are shown in figure 4.

IV. Simulation Results

A. System Model

The Frequency Occupancy Information (FOI) Dissemination system model is shown in figure 5. In this figure, we suppose that one cognitive radio node is the transmitting node (the TX node) that gathers FOI information by using a spectrum sensing device (a scanner) and disseminates it to N FOI receiving nodes (the AAF RX nodes). We also assume that an (ideal) feedback channel between the TX node and the RX nodes exists. In the case of using Fountain codes we do not use this feedback channel.

Let T_F be the FOI-interval, N_F be the number of FOI bits. Assume that every T_F, a block of N_F FOI bits is gathered by the sensing function. These N_F FOI bits represent the frequency occupancy information of a certain band, which is a small part of the whole band between $400-1000$ MHz. In the cognitive radio system, cognitive radio nodes use control channels for the exchange of control and sensing information [4]. Hence, the FOI bits we mean here include not only the frequency occupancy information, but also the control information. These are the bits that have to be disseminated to the RX nodes reliably. T_F depends on the usage pattern of the primary users (e.g. whether the secondary user aims at re-using a data channel or a TV channel) and on the variability of the communication channel (e.g. as specified by its coherence time). Suppose that the N_F FOI bits consists of an occupancy vector of bits representing usage or non-usage by the primary user and a vector of bits corresponding to the control information. The occupancy vector may be assumed to be created using an FFT of length between 16 and 2048. Due to the control information, we assume that $16 \leq N_F \leq 8192$.

In order to disseminate these N_F FOI bits reliably, these bits will be encoded by the error-correcting code and the fountain codes. Let N_c code bits be the number of the encoded FOI bits, T_U be the update-interval of N_c code bits, which is the time interval between the end of the previous scanning-information-dissemination and the beginning of the present scanning-information-dissemination. These N_c code bits are received by all N TX nodes in the dissemination time T_S. We want $T_S \leq T_U$ and the update-interval T_U should be less or equal to the FOI-interval: $T_U \leq T_F$. So

$$T_S \leq T_U \leq T_F. \tag{10}$$

Suppose that the dissemination time T_S is composed by the propagation delay given by distance d divided by the speed of light c (to be neglected in our application), a transmission delay given by the amount of bits N_c divided by the bit rate R_b of the dissemination channel and a processing delay T_P (also to be neglected or to be accommodated for by increase of processing power), so

$$T_S = \frac{d}{c} + \frac{N_c}{R_b} + T_P. \tag{11}$$

In our application, the propagation delay $\frac{d}{c}$ and the processing delay T_P will be neglected, so (11) can be rewritten as:

$$T_S = \frac{N_c}{R_b} \tag{12}$$

As mentioned before, the fountain codes only work in an erasure channel, so the error-correcting codes will be used to convert the noisy wireless channel to an erasure channel.

Code	Combination	K
(15,5)	(15,7) with 2-bit CRC	1637
(63,29)	(63,37) with 8-bit CRC	283
(255,143)	(255,175) with 32-bit CRC	58

TABLE I
THE NUMBER OF FOI PACKETS FOR THE DIFFERENT CONCATENATED CODES.

Assume that the error correcting code rate is $R_c = \frac{k}{n}$, and N_F bits are separated into $K = \frac{N_F}{k}$ packets. For every packet, k/R_c bits need to be transmitted. In order to recover the source information, a fountain code needs $K(1+\varepsilon)$ packets in order to decode correctly, so:

$$N_c = \frac{N_F}{R_c}(1+\varepsilon)/(1-f) \quad (13)$$

code bits need to be transmitted in the case of erasure rate f of the erasure channel. According to (12), we could get the dissemination time T_S and are able to check whether it satisfies (10).

B. Relation Between T_S and $\frac{E_b}{N_0}$

In this section, we find how T_S varies with the bit signal-to-noise ratio $\frac{E_b}{N_0}$ for a noisy channel model, e.g, Rayleigh fading channel. The idea is to establish the corresponding erasure rate of the noisy channel from a certain $\frac{E_b}{N_0}$ subsequently. We find how T_S changes according to the variation of $\frac{E_b}{N_0}$.

Here, we will use three error-correcting codes to convert the noisy channel into an erasure channel, as described in section III. The number of FOI packets in these codes is listed in table I. The implementation of this simulation is based on software called IT++ [14]. The way to get the erasure rate corresponding to $\frac{E_b}{N_0}$ is to generate K FOI packets until we receive K′ correct FOI file. Therefore, the erasure rate is given by:

$$f = \frac{\mathrm{K} - \mathrm{K}'}{\mathrm{K}} \quad (14)$$

where K′ is the number of received FOI packet and $\mathrm{K}' = 5000$ in our case. In this paper, we assume that the channel is a Rayleigh fading channel and the signal is QPSK-modulated. The performance of these three error-correcting codes is shown in figure 6. From this figure, we draw some conclusions: the shorter code has lower code rate than the longer one, and the erasure rate of the shorter error-correcting code is lower than the longer one under the condition of the same Eb/N0.

Suppose the bandwidth (BW) is equal to 1 MHz, and the transmission rate R_b is given by:

$$\frac{R_b}{\mathrm{BW}} = \log_2 \mathrm{M}, \quad (15)$$

where M is the amount of points in the constellation diagram. Because the symbol is QPSK-modulated, R_b=2 Mbits/s.

Figure 6 shows the relationship between the erasure rate and $\frac{E_b}{N_0}$. From (13), we could derive the relation between T_S and $\frac{E_b}{N_0}$ as:

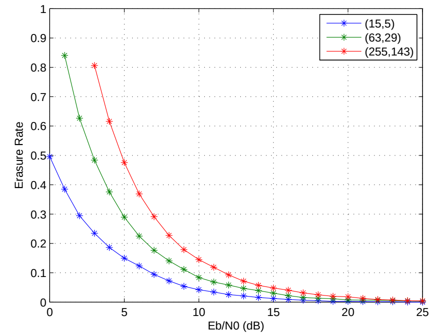

Fig. 6. Performance of the different error-correcting codes in the Rayleigh channel with QPSK.

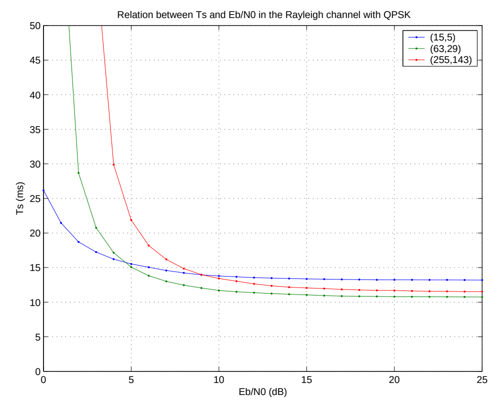

Fig. 7. The relation between T_S and $\frac{E_b}{N_0}$ in the Rayleigh channel with QPSK

$$\begin{aligned} T_S &= \frac{N_c}{R_b} \\ &= \frac{N_F}{R_c} \cdot (1+\varepsilon) \cdot \frac{1}{1-f} \cdot \frac{1}{R_b} \\ &= \frac{N_F}{R_c} \cdot (1+\varepsilon) \cdot \frac{1}{1-f(\frac{E_b}{N_0})} \cdot \frac{1}{\log_2 M \cdot \mathrm{BW}} \end{aligned} \quad (16)$$

The comparison of the relation between T_S and $\frac{E_b}{N_0}$ for these three error-correcting codes in an Rayleigh fading channel and with QPSK modulation scheme is shown in figure 7.

The conclusions from figure 7 are: T_S for the (63,29) code is the smallest in case $\frac{E_b}{N_0}$ is larger than 5 dB. When $\frac{E_b}{N_0}$ is less than 5 dB, the (15,5) code has a smaller dissemination time than the other two cases. As $\frac{E_b}{N_0}$ is larger than 9 dB, the (255,143) code works better than the (15,5) code.

C. Relation Between T_S and BW

In this section, we establish how the dissemination time T_S changes with the bandwidth BW in the good channel and bad channel, respectively. The good channel we defined here has high $\frac{E_b}{N_0}$ and high corresponding erasure rate, whereas the bad channel is defined in the opposite way.

Suppose $\frac{E_b}{N_0}$ for the good Rayleigh channel is 15 dB, and $\frac{E_b}{N_0}$ for the bad channel is 5 dB. The erasure rates are determined

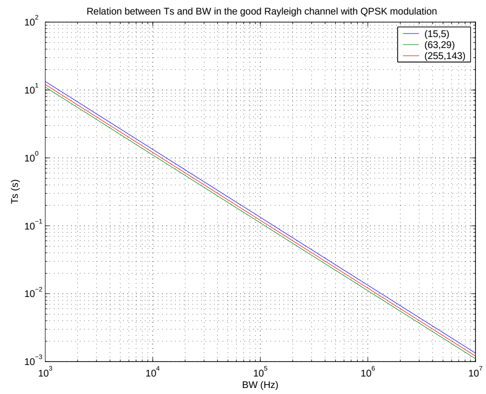

Fig. 8. The relation between T_S and BW in the good Rayleigh channel with QPSK

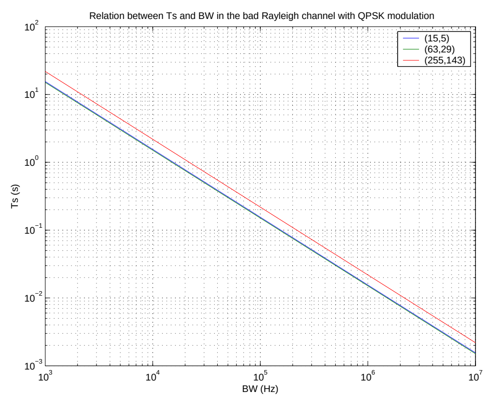

Fig. 9. The relation between T_S and BW in the bad Rayleigh channel with QPSK

by the same way as described in section IV-B. The erasure rates for the good channel and bad channel with QPSK are shown in table II. From (13) and (15), we determined the variation of T_S corresponding to BW, as shown in figures 8 to 9. From these two figures, we draw the conclusion that:

- The dissemination time T_S for the case of the (63,29) code is the smallest for each channel.
- The dissemination times T_S for these error-correcting codes are quite close to one another.
- In the bad Rayleigh channel, the line of T_S for the (15,5) code and that for the (63,29) code almost overlap.
- There is a restriction between T_S and BW. Smaller T_S requires larger bandwidth.

Code	Channel State	$\frac{E_b}{N_0}$ (dB)	Erasure Rate (%)
(15,5)	Good	15	1.24
(63,29)	Good	15	3.08
(255,143)	Good	15	4.83
(15,5)	Bad	5	14.97
(63,29)	Bad	5	28.97
(255,143)	Bad	5	47.55

TABLE II
ERASURE RATE FOR EACH CONCATENATED CODE IN EACH CHANNEL STATE WITH QPSK

D. Scenario Comparison

In order to find out whether using a particular approach to disseminate FOI packets is more efficient under the criterion of the minimum dissemination time T_S, we are going to compare three scenarios:

- Scenario 1: using a fountain code.
- Scenario 2: using a retransmission protocol.
- Scenario 3: increasing the transmitted power without using a fountain code or a retransmission protocol.

The common assumption of these three scenarios is that the noisy channel is converted to the erasure channel by using the error-correcting codes as described in section III. We also assume that there are N AAF RX nodes in this comparison and N $= 10 \sim 100$. The Rayleigh channel is modeled as the noisy wireless channel in the simulation. QPSK will be employed as the digital modulation scheme in the simulation. For each $\frac{E_b}{N_0}$, there is a corresponding erasure probability. Suppose $\frac{E_b}{N_0}$ for a good channel is 15 dB and $\frac{E_b}{N_0}$ for a bad channel is 5 dB.

These three scenarios will be compared in different simulation conditions as described above. According to our criteria, the most efficient scenario has the minimum dissemination time T_S. In other words, under the condition of the same transmission rate, we are going to find the scenario transmitting the minimum amount of code bits N_c.

Table II shows the erasure rate f corresponding to the specific channel condition and the error-correcting code. Now, suppose f is the erasure rate, K is the number of the source FOI packets, n is the block length of the error-correcting code, and ε is the percent of extra packets that need to be received in order to recover the source FOI file. Therefore, for the first scenario using fountain codes, the number of the transmitted packets is given in (13), which with $R_c = \frac{k}{n}$ is

$$\mathrm{N}_{Sc1} = \mathrm{K} \times n \times (1+\varepsilon) \times \frac{1}{1-f}. \tag{17}$$

These values are listed in table III.

Model Nr.	Code	Channel State	$\frac{E_b}{N_0}$ (dB)	f (%)	N_{Sc1}
1	(15,5)	Good	15	1.24	26731
2	(63,29)	Good	15	3.08	22101
3	(255,143)	Good	15	4.83	24115
4	(15,5)	Bad	5	14.97	31048
5	(63,29)	Bad	5	28.97	30156
6	(255,143)	Bad	5	47.55	43756

TABLE III
THE N_{Sc1} LIST OF THE FIRST SCENARIO (K=1637, R=0.33)

In [11], it is observed that for the second scenario using a retransmission protocol, the expected number of times a single packet has to be transmitted to be all received by all receivers is given by:

$$E[T] = \sum_{k=1}^{\infty} k((1-f^k)^N - (1-f^{k-1})^N) \tag{18}$$

where T is the number of times that a single packet has to be transmitted to be received by all receivers. So, the required

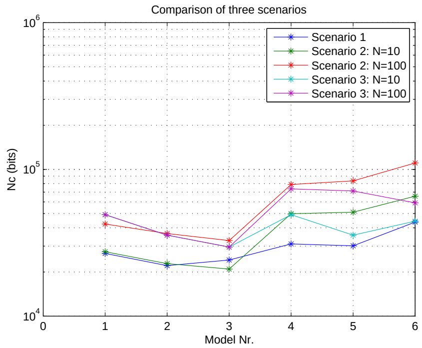

Fig. 10. Comparison of three scenarios

number of the transmitted packets is given by:

$$\mathrm{N}_{Sc2} = K \times n \times E[T], \quad (19)$$

whose value is listed in table IV.

Model Nr.	Code	Channel State	$\frac{E_b}{N_0}$ (dB)	f (%)	$E[T]$		N_{Sc2}	
					N=10	N=100	N=10	N=100
1	(15,5)	Good	15	1.24	1.1189	1.7283	27475	42438
2	(63,29)	Good	15	3.08	1.2784	2.0498	22793	36546
3	(255,143)	Good	15	4.83	1.4147	2.213	20923	32730
4	(15,5)	Bad	5	14.97	2.0302	3.2156	49852	78959
5	(63,29)	Bad	5	28.97	2.8661	4.6875	51100	83573
6	(255,143)	Bad	5	47.55	4.44	7.478	65668	110600

TABLE IV

THE $E[T]$ AND N_{Sc2} LIST FOR THE SECOND SCENARIO (K=283, R=0.46)

Without using the fountain codes and the retransmission protocol, the third scenario consists just increasing the transmitted power P_t, which means its $\frac{E_b}{N_0}$ increases and its corresponding erasure rate decreases. In this scenario, the AAF TX node just keeps transmitting all FOI packets again and again until all cognitive radio RX nodes receives the whole FOI file. We assume the transmitted power P_t in the previous two scenarios is 100 mW, and P_t' in the third scenario is 200 mW. Hence, $\frac{E_b}{N_0}'$ is increased by 3 dB comparing to $\frac{E_b}{N_0}$ in the first two scenarios. We suppose the new erasure rate is f' after increasing the transmitted power, and the corresponding $E[T']$ could be got from (18). So, the required retransmission times in this scenario is $\lceil E'[T] \rceil$, and the required number of the transmitted FOI packets is given by:

$$\mathrm{N}_{Sc3} = \mathrm{K} \times \lceil E[T'] \rceil \quad (20)$$

The value of f' and N_{Sc3} is listed in table V.

Model Nr.	Code	Channel State	$\frac{E_b}{N_0}$ (dB)	f (%)	f' (%)	$E[T']$		N_{Sc3}	
						N=10	N=100	N=10	N=100
1	(15,5)	Good	15	1.24	0.49	1.0482	1.3905	49110	49110
2	(63,29)	Good	15	3.08	1.35	1.1289	1.7614	35658	35658
3	(255,143)	Good	15	4.83	2.5	1.2301	1.9827	29580	29580
4	(15,5)	Bad	5	14.97	7.24	1.5836	2.4484	49110	73665
5	(63,29)	Bad	5	28.97	14.05	1.9927	3.1509	35628	71316
6	(255,143)	Bad	5	47.55	22.72	2.4808	3.9983	44370	59160

TABLE V

THE $E[T']$ AND N_{Sc3} LIST FOR THE SECOND SCENARIO(K=58, R=0.56)

Tables III $\sim$ V already showed the difference between the three scenarios. In order to see their difference clearly, we put all these N_{Sc} values as a function of the model number in figure 10, and draw the following conclusions:

1) For scenario 1 - using fountain codes, the value of N_{Sc} for each model has nothing to do with N, whereas for the case of the other two scenarios, they have different values of N_{Sc} for different numbers of the AAF RX nodes.
2) The value of N_{Sc} goes up when the channel is changed from the good state to the bad one for all these scenarios, except the case of scenario 3 with 10 AAF receivers. In this case, N_{Sc} almost has the same range in the bad and good channel.
3) In the good channel, scenario 1 and scenario 2 with 10 AAF receivers almost have the same performance, which is better than other scenarios. For the case of employing the (255,143) code, scenario 2 with 10 AAF receivers has smaller N_{Sc} than scenario 1, whereas for the other two codes, scenario 1 is slightly better performed than scenario 2 with 10 AAF receivers.
4) In the bad channel, the advantage of using fountain codes (scenario 1) is obvious, for the fact that its N_{Sc} is much smaller than that of other scenarios.
5) In general, scenario 1 works more efficient than the other two scenarios.

As we mentioned before, we want to find out the minimum dissemination time T_S which is related to N_c and R_b. In table III $\sim$ V, we found N_c, while the values of R_b are determined by the bandwidth BW. For the AAF project, BW is a limited resource, but BW will be the same for each scenario. In other words, the relation between T_S and BW in each model for each scenario will have the same variation trend. Therefore, it does not make any difference between comparing N_c and comparing T_S, for the reason that the purpose of this project is to find out which scenario transmits the minimum code bits. Through the comparison above, we draw this conclusion: in the scenario of using fountain codes, the AAF TX node transmits the minimum code bits comparing to the other two scenarios.

V. CONCLUSIONS

In this paper, we showed using fountain codes in FOI dissemination is more efficient than using the other schemes, e.g. retransmission protocol or increasing the transmitted power. Besides, we showed the tradeoff between the dissemination time T_S and bandwidth BW, as shown in figure 8 - 9. Short dissemination time requires large bandwidth; and small bandwidth makes the dissemination time long. For instance, T_S at 10^{-3} second needs 10 MHz bandwidth, and T_S at 10^{-1} second only asks 0.1 MHz bandwidth.

REFERENCES

[1] AAF. "The Adaptive Ad-hoc Free band communications (AAF) project Websites." http://www.freeband.nl/project.cfm?id=488, http://www.sas.el.utwente.nl/research/wireless_communication/AAF/ and http://www.wmc.ewi.tudelft.nl/research/projects/aaf.htm, 2004-2008.

[2] X. Shao. "Fountain Codes fore Frequency Occupancy Information Dissemination". Master's thesis, EECSM department, University of Twente, March 2006, EECSM-SAS10.06.
[3] D.J.C.Mackay: *Fountain codes*, http://www.inference.phy.cam.ac.uk/mackay/fountain.pdf.
[4] R.W.Brodersen, A.Wolisz, D.Cabric, S.M.Mishra, D.Willkomm: *CORUS: A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum*, http://bwrc.eecs.berkeley.edu/Research/MCMA/CR_White_paper_final1.pdf
[5] F. Hoeksema, M. Heskamp, R. Schiphorst, K. Slump: *A Node Architecture for Disaster Relief Networking*, Dynamic Spectrum Access Networks (DYSPAN), 2005.
[6] D. J. C. Mackay. *Information Theory, Interference, and Learning Algorithms*, Cambrige University Press 2003.
[7] Y. Kou, S. Lin, and M. P. C. Fossorier: *Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results*, IEEE Transcations on Information Theory, VOL.47, NO.7, November 2001.
[8] T. S. Rappaport: *Wireless Communications Principles* & *Practice*, Prentice Hall, 1998.
[9] J. G. Proakis, *Digital Communications*, Fourth Edition, McGraw-Hill, 2001.
[10] H. Futaki, T. Ohtsuki, *Low-Density Parity-Check (LDPC) Coded OFDM systems*,
[11] T. Lippmann: *Fountain codes on wireless channels*, Master's thesis, EECSM department, University of Twente, Feberary 2006, EECSM-SAS03.06.
[12] http://www.mathpages.com/home/kmath458.htm
[13] M. Luby, *LT Codes*, http://wol.ra.phy.cam.ac.uk/mackay/dfountain/LT.pdf.
[14] http://itpp.sourceforge.net/latest/index.html