# CR Based Video Communication Testbed with Robust Spectrum Sensing / Handoff

**Ji Qi, Fei Hu, Xin Li, A.M. Koushik, Lei Hu and Sunil Kumar**

**Abstract** As radio spectrum is becoming congested, wireless communications require more efficient spectrum usage. Recently, the cognitive radio (CR) techniques have become attractive as they can utilize any unused spectrum. In this paper, we build a CR video communication test-bed, which implements spectrum sensing and spectrum handoff functionalities in USRP boards. We have implemented compressive spectrum sensing, intelligent spectrum handoff, multi-video-flow transmission, under TDMA scheduling and Raptor codes for reliable video transmissions. By using compressive sensing in spectrum sensing method, the spectrum detection accuracy is improved without much algorithm complexity, and it is also robust to the noise uncertainty due to the use of cyclostationary features. To realize intelligent spectrum handoff, we have designed a real-time jamming detection scheme, as well as synchronized spectrum switching method. We have also implemented a multi-point TDMA-based communication system, which enables any node to send out multiple video flows to different neighbors with pipelined and scheduled data transmissions. We also proposed a special rateless codes called prioritized Raptor codes for more reliable video transmission and implemented in the GNU Radio applications. The proposed CR video transmission testbed can be used for new protocol testing purpose.

**Keywords** Cognitive radios · USRP · Compressive spectrum sensing · Intelligent spectrum handoff · TDMA · Raptor code

## 1 Introduction

Due to the fixed frequency allocation policy in today's wireless networks, the spectrum has become a scarce and precious resource in wireless communications.

J. Qi(✉) · F. Hu · X. Li · A.M. Koushik · L. Hu
Electrical and Computer Engineering, University of Alabama, Tuscaloosa, AL, USA
e-mail: jqi5@crimson.ua.edu

S. Kumar
Electrical and Computer Engineering, SDSU, San Diego, CA, USA





S. Latifi (ed.), *Information Technology New Generations*,
Advances in Intelligent Systems and Computing 448,
DOI: 10.1007/978-3-319-32467-8_6

However, a large portion of the assigned spectrum is not efficiently used. The utilization of assigned spectrum varies sporadically and geographically, and ranges from 15% to 85% in time [1]. Many Novel techniques have been investigated to solve this problem. Recently, the cognitive radio networks (CRN) [2] have become attractive as they can better utilize the existing wireless spectrum and provide high bandwidth through dynamic spectrum access (DSA).

According to the current fixed frequency allocation policy, the whole spectrum is divided into small bands with different ranges. Each band is exclusively used by a specific wireless system. In CRNs, there are two co-existing systems in the same frequency range, which are called primary system and secondary system [3]. Primary system is the licensed system with legacy spectrum. This system has the exclusive privilege to access the assigned spectrum. Secondary system represents the unlicensed cognitive system and can only opportunistically access the spectrum holes which are not used by the primary system. We call the user in the primary system as Primary User (PU) and the user in the secondary system as Secondary User (SU). By allowing the SUs to temporarily access the PU's under-utilized licensed spectrum, spectrum can be utilized effectively.

CRNs provide the capability of using or sharing the spectrum in an opportunistic manner. With dynamic spectrum access, a CRN can operate in the best available channel. More specifically, the CRN enables the users to (1) determine which portions of the spectrum are available and detect the presence of licensed users(PU) in licensed bands (spectrum sensing), (2) select the best available channel (spectrum management), (3) coordinate access to this channel with other SUs (spectrum sharing), and (4) vacate the channel when a PU is detected (spectrum mobility).

Different hardware test platforms have been built to study CRNs. The Universal Software Radio Peripheral (USRP) [4] is one of the most popular software defined radio (SDR) platforms: it implements front-end functionality and A/D and D/A conversion. It assumes that physical layer processing is done on the PC that hosts the device. The USRP connects to the PC through a high-speed interface, and the host-based software is used to control the USRP hardware to transmit/receive data. The USRP device has a motherboard that has the following subsystems: clock generation and synchronization, FPGA, ADCs, DACs, host processor interface, and power regulation. These are the basic components required for baseband processing of signals. A modular front-end, called a daughterboard, is used for analog operations such as up/down-conversion, filtering, and other signal conditioning. In stock configuration, the FPGA performs several DSP operations, which ultimately provide translation from real signals in the analog domain to lower-rate, complex, baseband signals in the digital domain. In most cases, these complex samples are transferred to/from applications running on a host processor that performs DSP operations. The code for FPGA is open-source and can be modified to allow high-speed, low-latency operations.

Recent years much research on CRN and SDR have been done with USRP devices [5–7]. However, very few of them have built a comprehensive video communication network to evaluate the performance of spectrum sensing and spectrum handoff strategies. Our test-bed serves as a real-world platform to validate experimental ideas and verify protocols / algorithms in CRNs and real-time video transmission over wireless network. In our testbed, each laptop with its controlled USRP board, is called an USRP node .The USRP board is supported by an open-source software code repository, GNU Radio.

## 2 Compressive Spectrum Sensing

Compressive Sensing (CS) is developed in signal processing community, and is more efficient for the sparse signal sampling in the sensing step than the traditional Shannon-Nyquist sampling theorem. Other than using the Nyquist sampling which needs the sampling rate of more than double the highest frequency component in the signal, the CS uses much lower sampling rate by randomly collecting the samples from the entire sparse signal domain. Then the optimal method is used to iteratively reconstruct the original signal with little or no data loss. In our method, we directly sample the information from the cyclic domain without time-consuming signal reconstruction while reserving the important signature features for different types of modulated signals.

### *2.1 Cyclostationary Feature*

Modulated signals are in general coupled with sine wave carriers, pulse trains, hoping sequences, or cyclic prefixes, and thus have built-in periodicity. Even though the data is a wide-sense stationary random process, these modulated signals are characterized as cyclostationary [8]. Therefore, cyclostationary domain can be used to analyze the feature of the signals which are not stationary but with periodical appearance in specific frequencies. Cyclostationary feature detectors have been introduced as a complex 2-D signal processing technique for recognition of modulated signals in the presence of noise and interference [9].

Cyclostationary signal $x(t)$ has the property as

$$m_x(t) = m_x(t + kT) = E[x(t)] \quad (k = 1, 2, ..., N) \tag{1}$$

where $E$ is the expectation and estimation of the signal mean, $T$ is the cycle period. Thus the signal autocorrelation is

$$R_x(t, \tau) = R_x(t + kT, \tau) \quad (k = 1, 2, ..., N) \tag{2}$$

Taking FT w.r.t $\tau$, we get spectral correlation function (SCF) as

$$S_x^{\alpha}(f) = \lim_{T\to\infty} \lim_{\Delta t\to\infty} \int_{-\Delta t/2}^{\Delta t/2} \frac{1}{T} X_T\left(t, f + \frac{\alpha}{2}\right) X_T^*\left(t, f - \frac{\alpha}{2}\right) dt \tag{3}$$

The sufficient statistics used for the detection are obtained through non-linear squaring operation. Therefore, FFT-based methods are used in digital implementation of the cyclostationary detectors. Given N samples divided in blocks of $T_{FFT}$ samples, a simplified SCF is estimated as

$$\widetilde{S}_x^{\alpha}(f) = \frac{1}{NT}\sum_{n=0}^{N} X_{T_{FFT}}\left(n, f + \frac{\alpha}{2}\right) X_{T_{FFT}}^{*}\left(n, f - \frac{\alpha}{2}\right) \tag{4}$$

where $X_{T_{FFT}}$ is the N-point FFT around sample $n$.

## 2.2 Compressive Measurement and Compressed Signal Processing

In time-random, the received signal is usually sensed by CS to get a low dimensional vector $Y$ using a sensing matrix $\Phi$ as follows

$$Y = \Phi X \tag{5}$$

where $X$ is an $M \times 1$ vector representing the Nyquist samples of $x(t)$, $Y$ is the $N \times 1$ compressed measurement vector, and $\Phi$ is an $N \times M$ measurement matrix.

Therefore, a critical task in compressive sensing is to design the measurement matrix, which collects compressed signal measurements and fulfills the robust detection. Based on the matrix transformation theory we design the second-order measurement matrix. We not only detect the signals robustly under the colored noisy environment but also classify the different types of modulation signals in a wide spectrum.

For cyclostationary signal detection and classification, we use compressed signal processing (CSP) to reserve the signal geometry structure in the compressive domain. The classification we use here is defined as the CSP signal detector based on the hypothesis test, in order to distinguish different modulation signals between $\Phi(H_0)$ and $\Phi(H_1)$:

$$t_i = \min \| P_{\Phi}(t)X - P_{\Phi}(t)S_i \| \tag{6}$$

where the $P_{\Phi}(t)$ is the CSP detector and $s_i$ is the signal of the $ith$ user.

As shown in Fig. 1, our CSP detector is divided into two main parts. First, we add the cyclic feature into the CS random measurement and build a sampling matrix as described above. The sensing matrix is implemented via the low-rate sampling on the cyclic features, which are calculated via the filter banks.

Thus, with USRP hardware, we can incorporate autocorrelation function to visualize the cyclic features of various signals in real-time. By sending the RF signals from the USRP receiver to Matlab and then calculate the Cycle Domain Profile (CDP) via an efficient method to recognize the different modulated signals.

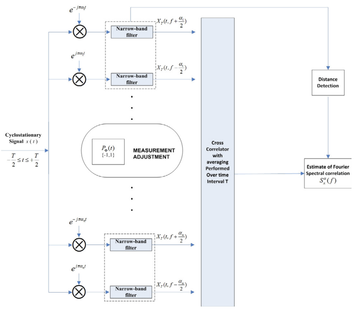



Fig. 1 Cyclostationary Compressed Detector.

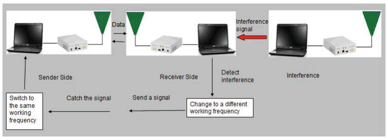



**Fig. 2** Spectrum Handoff Framework.

## 3 Intelligent Spectrum Handoff

With the development of Cognitive radio techniques, spectrum handoff has become one of the hot topics recently. In CRN, when a primary user (PU) reappears on a channel, existing secondary users (SUs) must return the channel control to the PU and find other unused spectrum band(s) to switch. Spectrum handoff techniques can help the interrupted SU to vacate the occupied licensed channel and find a suitable target channel to resume its unfinished data transmission.

### *3.1 Framework Design*

To realize spectrum handoff, we change the way that the conventional wireless system used to work. The third USRP node is used as an jamming node that will send jamming signals during the transmission process. We build a jamming event warning mechanism and make it possible for both sides to switch to the opposite mode (more details will be discussed in section B). When the receiver detects the jamming signal, it will send a warning signal to the sender side and changes to a different frequency. Then the sender will catch this signal and also switches to the same frequency. Fig. 2 shows our spectrum handoff framework.

During wireless video transmission, the quality of video can decrease sharply when the jamming signal comes. The reason is that the jamming signal will be in collision with the original video signal, which causes serious packet loss. We use the packet error rate (PER) to detect the invasion of the jamming signal.

In USRP implementation, we put a checksum into the head of each video packet. Thus the receiver can detect whether the current packet is correctly received or not. With this information, we can obtain the real-time PER of our system. However, the wireless link has fast fading. Even there is no jamming signal, the packets can still be discarded during the transmission. If every occurrence of a non-zero PER is judged as an invasion of the jamming signal, we will get many false alarms. Based on the above analysis, we can determine the invasion of the jamming signals by checking:

(1) PER is higher than a threshold $\delta$, and (2) PER remains high for a sufficient time duration $T$. The whole system operates as follows:

- Both sender and receiver are set to a specific spectrum frequency $f_1$ and it starts video transmission.
- During the transmission, after every time interval $\Delta T$, the sender side will switch to receiving mode with a different frequency $f_2$ for a short time $\Delta t$, and then change to sending mode again.
- The jamming node, starts to send jamming signals at frequency $f_2$.
- The receiver detects the jamming signal and switches to sending mode with the working frequency $f_2$.
- Once the sender side catches the signal, it will change to another working frequency $f_3$, and the receiver side will also switch to the same frequency $f_3$.

## 3.2 *Multi-thread Scheduling*

In GNU Radio programming, each action of the USRP hardware will be controlled by one thread. In the thread, users can build a *radio chain* by defining a flowgraph using the *connect* function. The connect function specifies how the output stream of a processing block connects to the input stream of downstream blocks. The flowgraph mechanism then automatically builds the flowgraph. However, as all of the inputs and outputs are predefined before running, it is impossible for the USRP hardware to perform two different actions (i.e., sending and receiving) in one thread. Therefore, to realize the mode switch in our spectrum handoff, we need to introduce multi-thread scheduling.

Both sender and receiver need to switch between sending mode and receiving mode. In software implementation, we adopt two-thread scheduling. Each thread runs independently according to the preset timetable. If the system notices that there is a jamming signal, one thread will send a warning message to the other one to change its RF frequency. The thread schedule on the sender and receiver is shown respectively in Figs. 3 and 4.

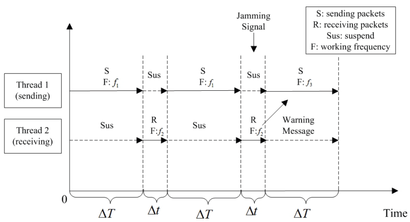



**Fig. 3** Multi-thread Scheduling on Sender Side.

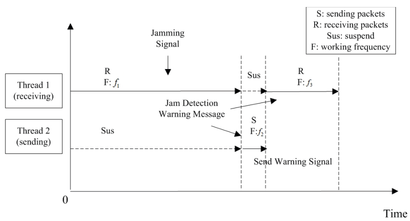



**Fig. 4** Multi-thread Scheduling on receiver Side.

# 4 Network Protocols and Applications

## *4.1 Multi-video Transmission via TDMA Framework*

Time division multiple access (TDMA) is a channel access method for a shared wireless medium. It is typically used in MAC layer. But scheduled communication can occur in any layer. It allows several users to share the same frequency channel by dividing the communication duration into different time slots. The users transmit in rapid succession, one after the other, each using its own time slot. This allows multiple stations to share the same transmission medium (e.g. radio frequency channel) while using only part of its channel capacity.

In our test-bed, we first use two USRP nodes to build a basic TDMA transmission network. We divide the whole transmission process into time slots and then send out two video streams alternately. According to the predefined time slot for each stream, the receiver can correctly decode and display two real-time video streams (we use ffmpeg, x264 and ffplay to encode, decode and display the video). The whole system is shown in Fig. 5. The whole system will work as the following steps:

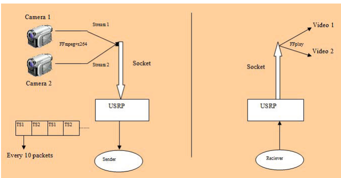



**Fig. 5** TDMA Dataflow and Framework

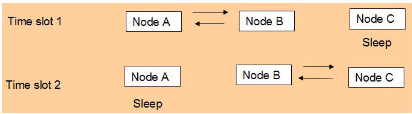



**Fig. 6** Scheduled Data Transmission Framework.

– Set sender and receiver to a specific working frequency $f$ and start transmission.
– The sender switches between two video streams every 10 packets.
– The receiver sorts two data-streams based on the predefined time slots and displays two real-time video streams.

Then we increase the number of USRP nodes to demonstrate how our test-bed performs the scheduled data transmission in a network. Each node can switch between sending, receiving and sleeping modes quickly and randomly. Taking 3-nodes case as an example (see Fig. 6), here we define the time slot schedule such that only two nodes can talk with each other at the same time while the third one goes to sleep.The whole system works as follows:

– All three nodes are set to a specific frequency $f$ and start video transmission.
– Node A first begins to send packets and wakes up node B. Then node B starts to receive these packets, while node C keeps sleeping.
– After a time interval of $\Delta t_1$, node A changes to sleeping mode and node B starts to send packets and wakes up node C, at the same time node C will begin to receive.

– Again after a time interval $\Delta t_2$, node B will switch to sleeping mode and node C begins to send packets and wakes up node A, at the same time node A will change to receiving mode.
– After a time interval $\Delta t_3$, the system goes back to the second step.

### *4.2 Raptor Codes for Reliable Video Transmission*

Fading and shadowing in wireless channels cause packet loss and deterioration of video quality. When packet loss occurs, the feedback from the receiver can be used for the request of the retransmission of the loss packets. This retransmission mechanism is bandwidth-costly. Raptor codes, which are a class of powerful rateless codes, can completely recover the source data with little overhead and linear encoding/decoding time.

The Raptor codes consist of a precode (usually a LDPC code) as the outer code and a weakened LT code as the inner code. They can be parameterized by $(K, C, \Omega(x))$, where $K$ is the number of source symbols, C is a pre-code with block-length $L$ and dimension $K$, and $\Omega(x)$ is a degree distribution of LT codes. Each encoded symbol is associated with an ID (ESI). The pre-code and weakened LT code can ensure a high decoding probability with a small coding overhead.

In our test-bed, we use the systematic Raptor codes [10]. If there are $K$ source symbols $S[i]$ in one block, $i = 0, ..., K - 1$, the first $K$ encoded symbols are constructed such that $E[0] = S[0], E[1] = S[1], ..., E[K - 1] = S[K - 1]$. The systematic Raptor codes can therefore correctly decode some source symbols even if the number of received encoded symbols $N_r$ is less than the number of source symbols $K$.

As far as we know, no rateless codes (such as Raptor codes, LT codes) are implemented in GNU Radio and USRP platform. Only very simple FEC scheme, such as RCPC codes, is implemented in GNU Radio. Therefore, we built all Raptor codes programs from the scratch. In GNU Radio, applications are primarily written in Python programming language, while some performance-critical components are implemented in C++. Raptor codes, which include Gaussian Elimination and Belief Propagation, could consume many computing resources. Therefore we implemented the Raptor codes in C++. And then we used the SWIG software to transfer some Raptor codes C++ API into Python API to be called by GNU Radio applications. The implementation of Raptor codes for video transmission over USRP is shown in Fig. 7.

## 5 Experimental Results

This section presents the numerical and simulation results that verify the effectiveness of our proposed video CRN test-bed.

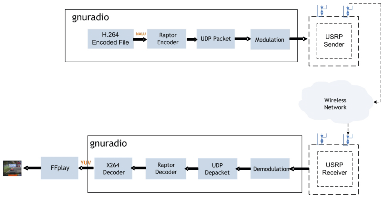



**Fig. 7** Video Transmission Network with Raptor codes

## *5.1 Spectrum Sensing*

Our proposed CSP detector is tested under different compressed rates and SNR levels, with the purpose of analyzing (1) the effects introduced by the sensing rate, (2) the detection accuracy, and (3) its robustness to the noise. As shown in Fig. 8, the classification errors quickly approach to zero under high SNR levels. The compression ratio is 50% for the worst case with a low error rate. It also means that half of the sensing energy can be saved.

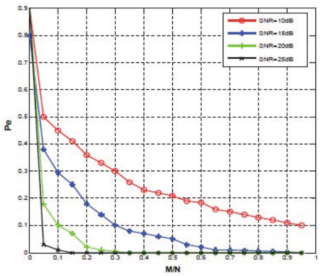



**Fig. 8** CSP under different SNR levels

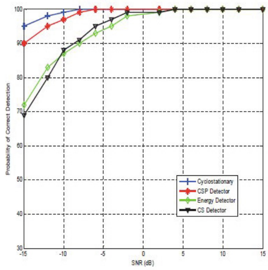


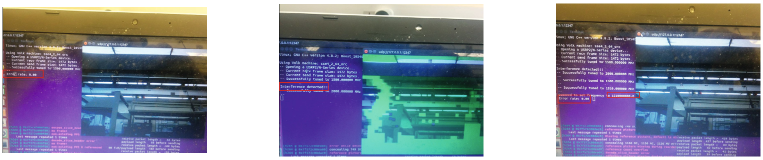

**Fig. 9** Detection Probability for different detectors.

We also compare the receiver operating characteristic (ROC) of our proposed compressive cyclic feature detector with the ROC of the energy detector, traditional CS and cyclostationary detectors. ROC is typically used to evaluate the detectors sensitivity and accuracy, and a larger area under the curve means a better performance. As shown in Fig. 9, under a SNR of 0 dB and with a 50% compression ratio ($M/N$) for the CSP measurements, the performance of our system is much better than the energy detection and CS detection methods.

## *5.2 Spectrum Handoff*

In this test, we use a real-time video transmission framework to test the channel switching accuracy and the response time of our intelligent spectrum handoff scheme.

In physical test the USRP sender node captures the real-time video with a built-in camera in the host computer, and then sends the video data out. And the jamming node sends out jamming signals. The performance of our spectrum handoff scheme in the receiver side is shown in Fig. 10.



(a) Before jamming (b) After jamming (c) Spectrum handoff

**Fig. 10** Spectrum Handoff Hardware Test (Receiver Side)

Through our tests, we find out that the the jamming signal detection can be done in 2 seconds but the whole frequency switching process can last for 15 to 20 seconds. There are two reasons for this time consumption. Firstly, it takes a while for USRP hardware to suspend current working thread and change to another thread. Secondly, the sender side needs to switch to receiving mode for a short time $\Delta t$ after every time interval $\Delta T$. Thus if the sender side is working on sending mode, even the receiver side tries to tell the sender side that the jamming signal is detected, the sender side may not be able to hear this warning message within a short time.

## 5.3 *TDMA Framework*

To test and verify the performance of our TDMA scheme, we check both the two-node case and the three-node case. For multi-video transmission between two USRP nodes, we use a built-in camera in the host computer as well as a USB camera to capture two video streams simultaneously. The performance of our multi-video transmission scheme in receiver is shown in Fig. 11.

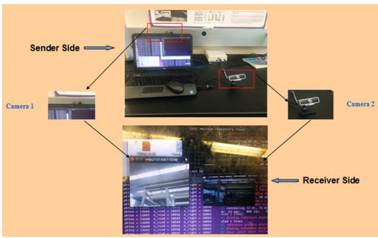



**Fig. 11** Multi-video Transmission Hardware Test (Receiver Side)

**Fig. 12** 10-th frame of the received video. Left: video transmission without Raptor codes; Right: video transmission with Raptor codes

### *5.4 Raptor Codes*

In our experiment, we evaluate the performance of video transmission over USRP with and without Raptor codes scheme. The raw video source is first encoded by H.264/AVC encoder and then encoded by Raptor codes at the Application layer. After that the encoded packets are encapsulated into UDP packets, which are modulated and transmitted over wireless networks using USRP. For comparison purpose, the raw video source is only encoded by H.264/AVC (without Raptor codes) and then gets transmitted by UDP protocol in USRP hardware.

From Fig. 12, we can see that the video quality with Raptor codes protection is much better than the one without Raptor codes protection.

## 6 Conclusions

In this work, we have built a CRN video communication test-bed based on USRP hardware. In our test-bed, we have incorporated some CRN techniques including spectrum sensing and spectrum handoff. To realize multi-video transmissions with high video quality, we have also adopted TDMA framework and Raptor codes. Experimental results showed that our test-bed provides a comprehensive CRN platform for video transmissions. In the future, we will continue to improve our test-bed framework, especially in CRN MAC layer and routing layer design.

## References


1. FCC, E.: Docket no 03-222 notice of proposed rule making and order (2003)
2. Thomas, R.W., Friend, D.H., DaSilva, L.A., MacKenzie, A.B.: Cognitive networks. Springer (2007)
3. Akyildiz, I.F., Lee, W.-Y., Vuran, M.C., Mohanty, S.: A survey on spectrum management in cognitive radio networks. IEEE on Communications Magazine **46**(4), 40–48 (2008)

4. Ettus, M.: Usrp users and developers guide. Ettus Research LLC (2005)
5. Dhar, R., George, G., Malani, A., Steenkiste, P.: Supporting integrated MAC and PHY software development for the USRP SDR. In: 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks, SDR 2006, pp. 68–77. IEEE (2006)
6. Sarijari, M.A., Marwanto, A., Fisal, N., Yusof, S.K.S., Rashid, R., Satria, M.H., et al.: Energy detection sensing based on GNU radio and USRP: an analysis study. In: 2009 IEEE 9th Malaysia International Conference on Communications (MICC), pp. 338–342. IEEE (2009)
7. Marwanto, A., Sarijari, M.A., Fisal, N., Yusof, S.K.S., Rashid, R., et al.: Experimental study of OFDM implementation utilizing GNU radio and USRP-SDR. In: 2009 IEEE 9th Malaysia International Conference on Communications (MICC), pp. 132–135. IEEE (2009)
8. Gardner, W., et al.: Exploitation of spectral redundancy in cyclostationary signals. IEEE on Signal Processing Magazine **8**(2), 14–36 (1991)
9. Gardner, W.A., Napolitano, A., Paura, L.: Cyclostationarity: Half a century of research. Signal Processing **86**(4), 639–697 (2006)
10. Shokrollahi, A., Luby, M.: Raptor codes. Now Publishers Inc. (2011)