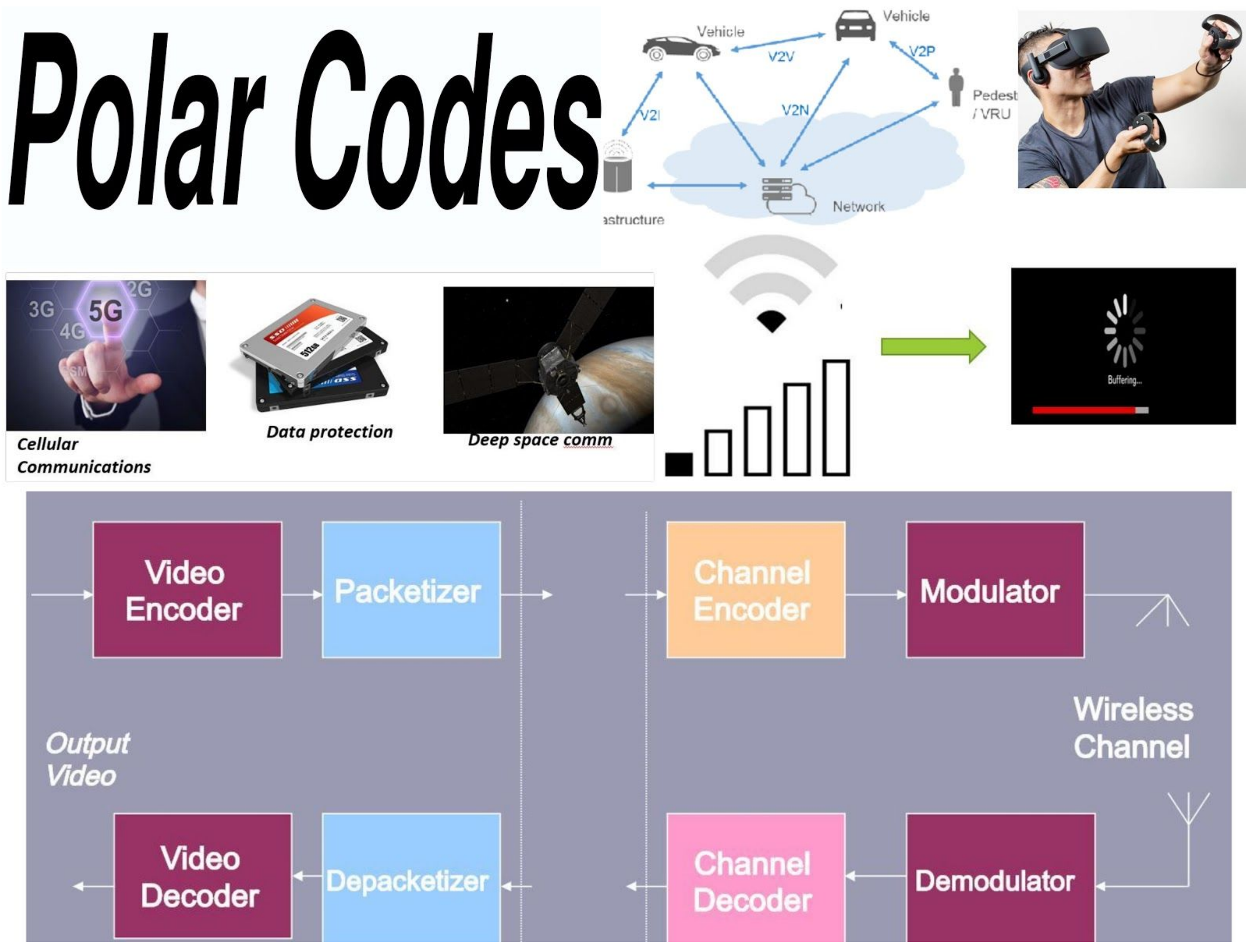
CSE 237D Final Report

Study of Polar Code Applications for Video Transmission in Extremely Lossy Wireless Environments



Arunkumar Ravichandran

Winter 2018

ABSTRACT

Low cellular signal strength affects streaming of high quality video. The packets sent will tend to have higher errors in them and the transmitter has to retransmit the packets often. Error correction is a technique where at the transmitter end we add parity information and at the receiver end we can correct those errors by mathematical computation. Thus retransmission of packet is avoided and throughput of the communication link is improved. In this course project, Polar codes, an error correction mechanism, is chosen for studying its performance for video streaming application. A comparison of various digital communication modulation schemes and their error correction performance, and how video transmission was done using Polar code in Linux platform is shown. The study was done using GNU-Radio Software defined platform with USRP hardware.

1. INTRODUCTION

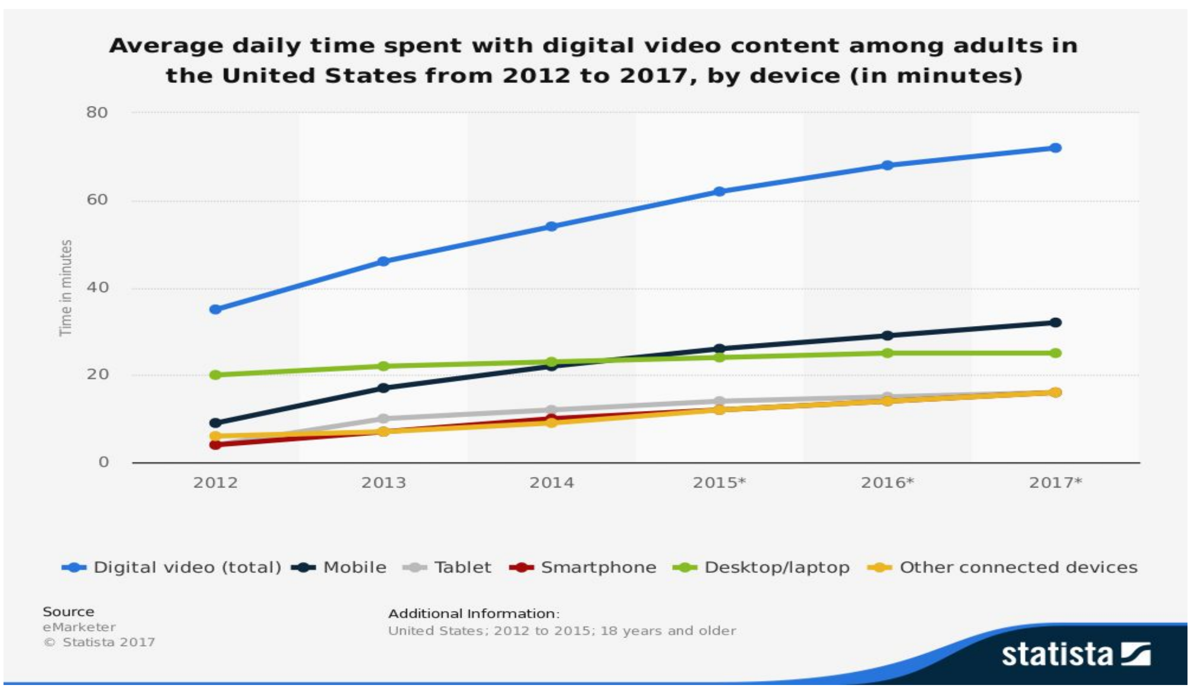


Fig 1: The graph shows that video consumption on mobile devices have increased compared to other connected devices over time and it's expected to increase further

As the amount of video-enabled mobile devices (such as smartphones) become more common, video consumption on mobile devices is increasing at a higher rate. However, restrictions in data transmission rate, battery life of the mobile device and wireless channel environment limits the performance of wireless video streaming on resource-constrained devices.

Providing high quality video over wireless networks is a challenging problem, due to both the erratic and time-varying nature of a wireless channel and the stringent delivery requirements of the media traffic. Applications such as VR gaming, autonomous driving by using V2X communication technology require higher bandwidth and low latency communication which could be enabled by having a reliable wireless transmission medium. Video and image data go through a lossy compression mechanism before being transmitted which are sensitive to errors in wireless medium. The quality of the transmitted data gets degraded due to the channel effects such as signal attenuation, interferences and collisions in the wireless medium. Forward error correction techniques are being employed to make the wireless channel reliable for transmission of data.

Channel coding is a technique to reduce the errors and make the communication channel more reliable. Polar Codes, proposed by Arikan [1] are a major breakthrough in the area of coding theory. These are the first error-correcting codes mathematically proven to achieve Shannon's Channel capacity. The reduced complexity of polar code encoding and decoding schemes have made them an attractive choice.

The polar codes have been primarily tested for a random binary input with various channel conditions such as AWGN noise, fading channels etc. In this study, we use data such as images and video streaming to test the efficiency of polar code to be used as a forward error-correction technique in the wireless medium. We also use different wireless modulation schemes to see which is better for Polar code transmission.

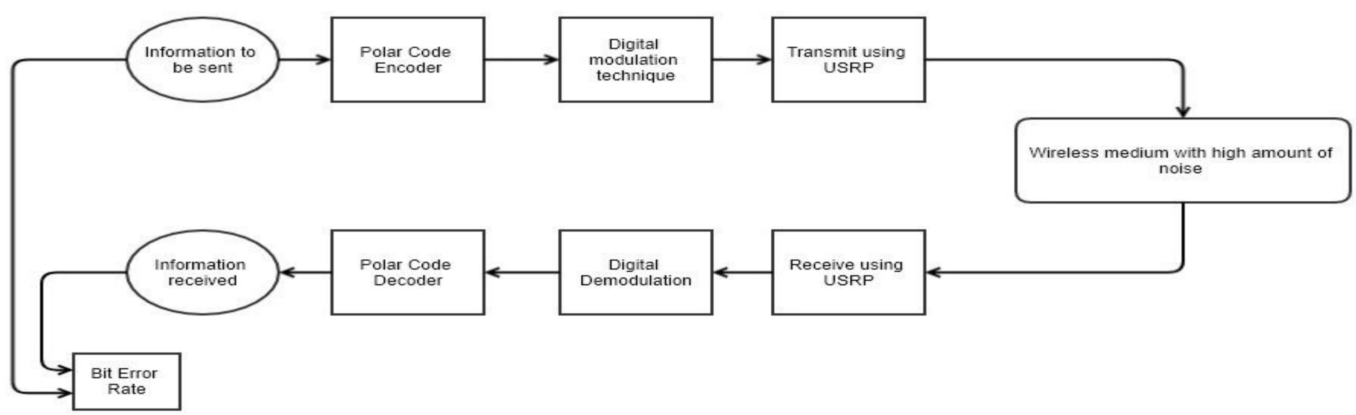


Fig 2: Overall Tx/Rx architecture implemented in USRP is shown.

The objectives of this project are:

1. Implement the Polar Code encoding and decoding blocks on the GNU Radio platform.
2. Implement wireless transmitter and receiver system on the GNU Radio /MATLAB platform to enable transmission of data over wireless using USRPs.
3. Identify the wireless digital modulation technique which works better for the Polar Code based error-correction mechanism.

4. Transmit image and stream video over wireless medium to calculate the Bit-error rate performance and packet loss performance respectively to study their performance.

BACKGROUND about POLAR CODES

Encoding:

To construct an (N,k) polar code, the k most reliable bits are used to carry the information bits. The rest of the bits are set as frozen bits(frozen bits are almost equivalent to parity bits in traditional error correction). The frozen bits are usually assigned a value of 0. However it can be assigned with other values too. The frozen bit locations are selected based on what kind of channel it is. Usually frozen bit locations are chosen for those bits which have higher probability of getting corrupted and data bit locations are chosen for those bits which have higher probability of transmission without any error. This reliability calculation is the integral part to achieve higher transmission capacity and reduce the errors. There are several mathematical tools proposed by various researchers to identify the reliability of a channel. The frozen bits are assumed to be known both at the transmitter and at the receiver. As shown in figure 3a, N= 8 K=4 Rate = ½, initial 4 bits are assigned the frozen bit values 0 and rest 4 bits are used to carry data.

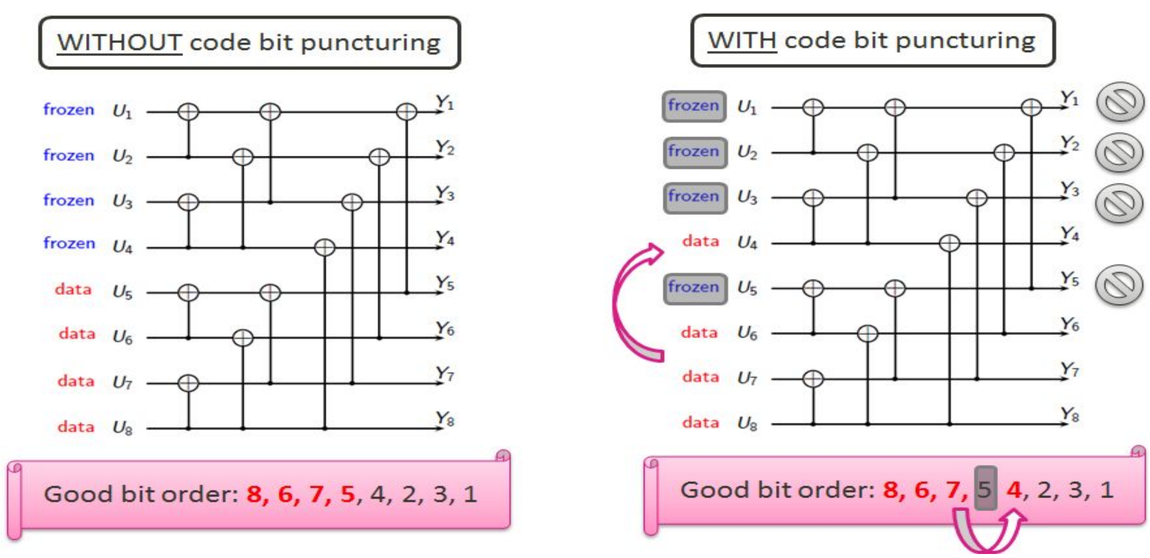


Fig 3a: Polar code construction with different Frozen bit locations are shown. In the left image, the frozen bit locations are assigned in the initial 4 bit locations arbitrarily. In the right image, the frozen bit locations are inserted in-between actual data information and this might increase the decoding performance at the receiver as suggested in [3].

Decoding:

The received bits can be realized in the form of a graph with depth N as shown in Figure 3b. There are several decoding schemes proposed and we chose to go with the

Successive-Cancellation list decoding. In this SC decoding, the correct information, also referred to as Code-word, which was sent at the transmitter is found by traversing the tree. While traversing, decisions are made to evaluate if the chosen path is correct or not based on the log-likelihood ratios(LLR). This Log-Likelihood Ratios (LLR) computation also helps in deciding whether a bit is a frozen bit or reliable bit. In a typical SC decoding, a single path will be evaluated to find the code-word. But in a List Decoding, L paths are decoded simultaneously so the process of finding the correct code word is accelerated.

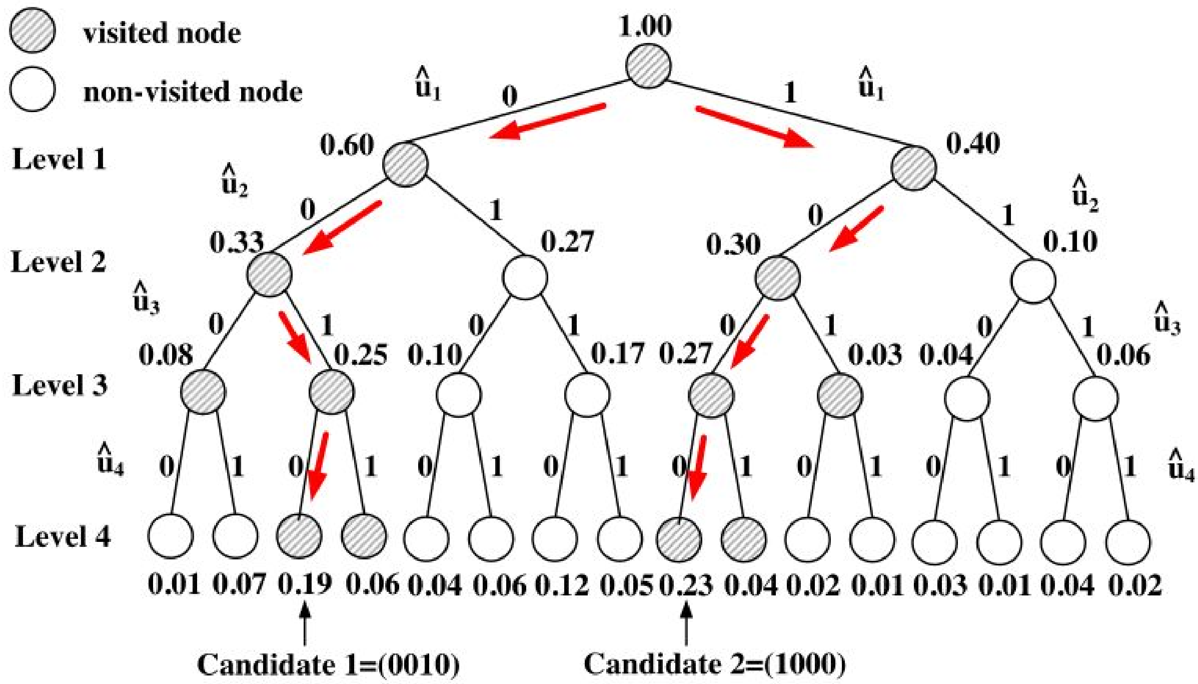


Fig 3b: Polar code decoding tree is shown. In this, based on the computed LLRs the path is chosen to generate the actual codeword which was sent. Here N=4 and L=2. Since L is 2, there are two paths(these 2 paths are computed in parallel; thus the output is 2 possible codewords and one of those codeword is chosen based on the Log-likelihood ratio) which is shown by the Red arrows.

2. TECHNICAL MATERIAL

The overall objective of the project is to build a wireless transceiver system using USRPs with Polar Codes as Forward error-correction mechanism.

2.1 Implementation of Polar Code blocks on GNU Radio:

This part deals with the implementation of the Polar code encoder and Successive cancellation list decoder on the GNU Radio platform. The GNU Radio platform provides a set of signal processing blocks that can be easily implemented using Python/C++ or the GUI available with the GNU Radio Companion. The GNU Radio platform had a preliminary implementation of Polar codes which are present in the function **polar_decoder_sc_list**. Below are the changes made to the Encoder and Decoder blocks to help in our study.

2.1.1 Removal of CRC check at the decoder:

In Tal et al[2], CRC check was added after the SCL decoding to improve the bit error rate performance. However this extra computation of CRC block at the transmitter and CRC checking at the receiver causes higher delay and reduces the throughput required for video transmission. Hence the appending of CRC at the end was removed and at the decoder the CRC check was eliminated. The elimination of the CRC check at the decoder would reduce the time taken to retrieve the packet and thereby provides a lower latency. Once the underlying code for encoder and decoder were changed in the GNU Radio platform, the GNU Radio was built again as suggested in their Github repository[4].

2.1.2 Log-likelihood Ratio based graph search in the Successive list cancellation decoding:

In the SCL decoder, Log-Likelihood Ratios (LLR) are computed which are helpful in the decision evaluation to find out whether a bit is a frozen bit or reliable bit. Maximum Likelihood based graph search is a resource intensive process and hence it was avoided. Therefore the LLR computation was added to the GNU Radio code to replace the Maximum Likelihood computation.

2.1.3 Selection of Frozen Bits:

To construct an (N,k) polar code, the k most reliable bits are used to carry the information bits. The rest of the bits are set to a frozen bit 0. Initially the frozen bits 0 are appended at the start of the information and their construction is done. Later, as suggested in Liu et al[3], we punctured the frozen bits at a different location so that the bit-error rate is increased because it's mathematically proven to improve the reliability. The bit positions of the frozen bit were assigned randomly within 1024 keeping few frozen bits fixed. This was done using MATLAB.

Fig 4: Polar code encoder and decoder blocks in GNU Radio is shown. These are the blocks which were developed as discussed in section 2.1

2.2 What are the digital modulation schemes used to study?

This part deals with the implementation of 4 modulation schemes used. The reason for choosing GNU Radio platform to study the polar codes is that this GNU Radio platform had all the modulation scheme blocks implemented already. Thus using them the study was made. The overall architecture of the study is shown in the figure.

2.2.1 BPSK - Binary Phase shift keying: In Binary Phase Shift Keying (BPSK) the 0's and 1's are represented by two different phase states: $\theta = 0^{\circ}$ for binary 1 and $\theta = 180^{\circ}$ for binary 0.

2.2.2 QPSK - Quadrature Phase shift keying: QPSK is similar to BPSK where 2 bits of data is sent at a time. The amount of radio frequency spectrum required to transmit QPSK reliably is half that required for BPSK signals, which in turn makes room for more users on the channel.

2.2.3 16-QAM - Quadrature Amplitude Modulation: In this modulation scheme two carriers are shifted in phase by 90 degrees. Thus the resultant output consists of both amplitude and phase variations.

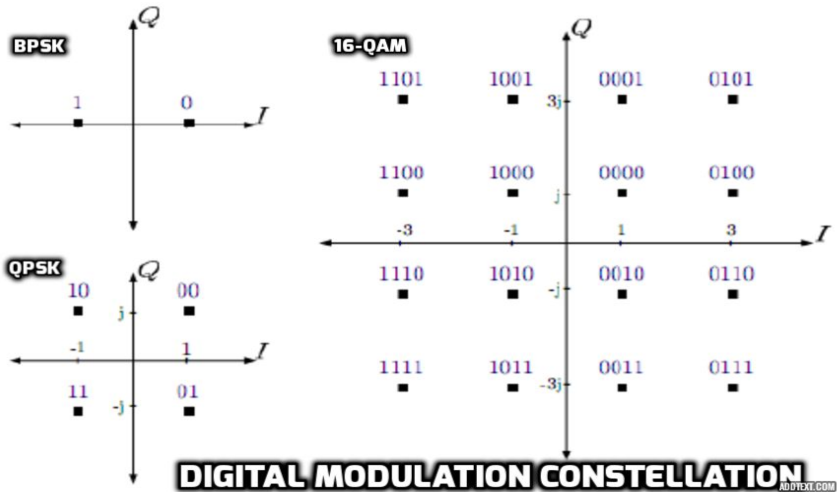


Fig 5: Constellation diagram for BPSK, QPSK, 16-QAM is shown. BPSK has the ability to send 2 distinct symbols, QPSK can send 4 distinct symbols and 16-QAM has the ability to send 16 distinct symbols. Thus more amount of information can be sent by using the 16-QAM modulation technique.

2.2.4 OFDM - Orthogonal Frequency Division Multiplexing: In OFDM the information to be transmitted is split into multiple smaller chunks and transmitted independently using different carriers. Cyclic Prefix is appended in OFDM to reduce the intersymbol interference.

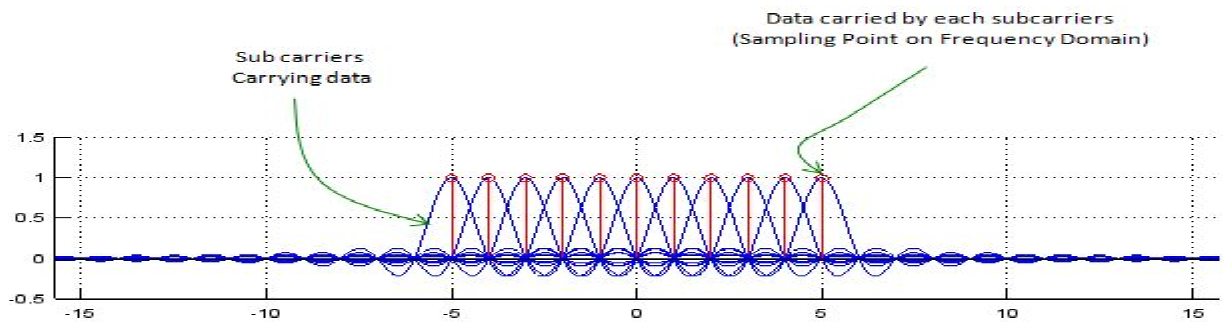


Fig 6: OFDM sub-carrier and data carriers is shown

2.3 Study setup:

In our study, we used 2 USRPs 2920 to transmit and receive the data. We used the GNU Radio platform and MATLAB to implement the communication blocks.

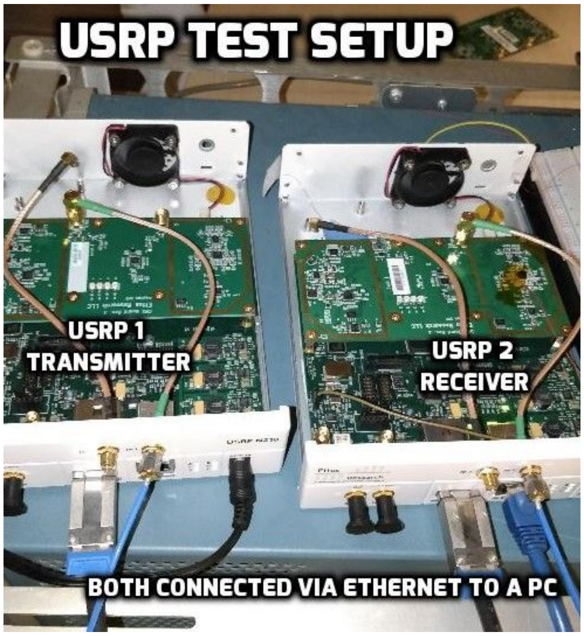


Fig 7: USRP test bed is shown. One USRP acts as Tx and the other USRP acts as Rx both connected to a system via Ethernet. The image on the right indicates the GNU radio blocks which are used to generate tx data and process the received data from USRP 2.

2.4 Observations:

2.4.1. CRC was not needed

Vardy et al[2] suggested that use of CRC at the decoder would improve the Bit-error rate performance. However based on the simulation results shown in the graph we can see that even with the CRC the bit-error rate were comparatively lower to that of a non-CRC based decoding system. This study was done using SCL decoder in MATLAB for different SNRs.

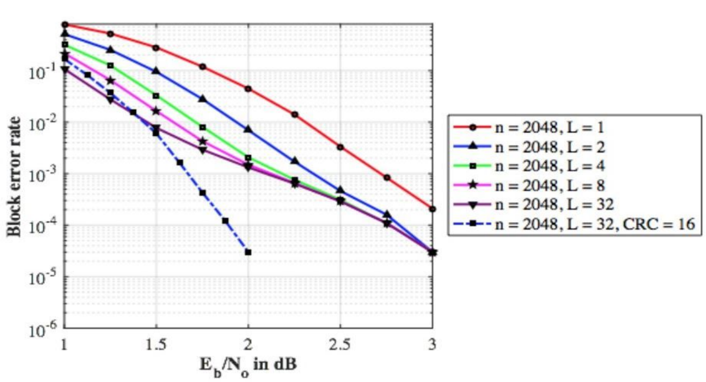


Fig 8: BER performance for different list size and with CRC is shown. SNR is shown in x-axis and Error rate is shown in y-axis. SCL with CRC has low error correction performance(dash dotted line) and hence CRC was not implemented in our SCL decoder. With higher list sizes BER performance increases but they have higher decoding latency as well. Thus we chose to have 1024 as block length(N) and list size(L) = 4.

2.4.2 Digital Modulation vs their performance

We implemented different modulation schemes on the GNU Radio platform to see which modulation scheme was better for Polar codes as forward error correction mechanism. As seen from the below graph, OFDM with 16-QAM was better in error correction performance. However OFDM with 16-QAM had the ability to carry higher data therefore higher block lengths were capable. This is in conjunction with the Polar codes results shown in Arikan et al [1]: As the block length increases channel reliability increases.

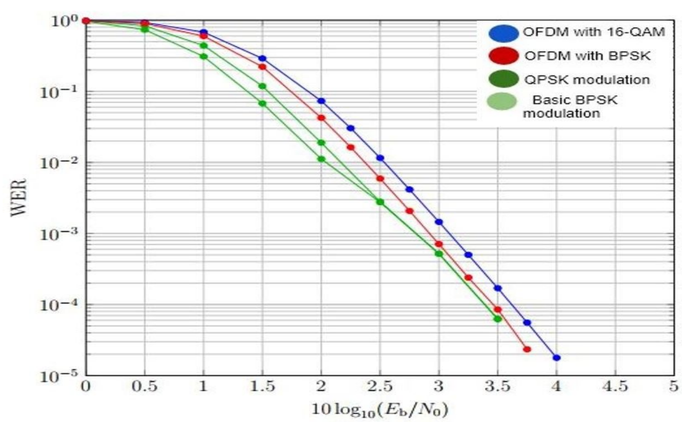


Fig 9: Error correction performance of different Modulation techniques are shown. SNR is shown on the x-axis and the error correction performance is shown on the y-axis. We can see that as the SNR increases error reduces. OFDM with 16-QAM was having higher error correction performance rate compared to other modulation schemes. This could be attributed to the ability of 16-QAM modulation technique to carry more amount of data and thus the reliable channels are higher in 16-QAM. The tests were conducted using both MATLAB and GNU Radio platform. OFDM with 16-QAM was performing better than other modulation techniques studied.

2.4.3. Image transmission

Image transfer was done using the GRC blocks present already. Image was transmitted as a file from one USRP to the other. The error rate was calculated which had similar results as shown above.

Fig 10: a and d indicate the transmitted image; b and e indicate the image obtained without any error correction mechanism; c and f indicates the image obtained after error correction. We can notice that b and e have higher noise in the image and appears distorted compared to image c and f.

2.4.4. Video streaming

Video streaming was done using the gstreamer tool available in Linux. The results of the video streaming tests are shown here. The packet loss results with Polar Code as FEC shows that the error is reduced compared to the packet loss without any FEC.

Trial	Packet loss without FEC	Packet Loss with FEC
1 (5Mb video)	45	34
2 (40Mb video)	126	56

Fig 11: Packet loss is low when Polar code FEC is used compared to when there was no FEC block present at the receiver.

2.4.5. Hardware implementation:

The time complexity of the SCL Decoder is very high O(LNlogN). As the Block length N = 1024 and list size L=4 which we are using is high, the decoding latency at the receiver is very high costing the video streaming to be choppy in nature. The packet processing delay at the receiver for a single packet was 1.34 seconds in the GNU radio platform. So to reduce the packet processing delay and to achieve higher quality video streaming, the GNU Radio blocks needs to be implemented in a FPGA which will improve the decoding performance.

However, the OFDM with 16-QAM as the demodulation technique could not be implemented in the FPGA because of very high resource usage. OFDM also requires packet detection and packet synchronization blocks which would increase the resource usage. Therefore a simple BPSK demodulation scheme was used to demodulate the packet and the output was given to the SCL decoder IP.

2.4.5.1. BPSK demodulator - Coherent

BPSK demodulator determines the phase of the input signal from the RF antenna to recover the information sent. We assume that there is a reference signal provided to the FPGA block which has the frequency and phase of the transmitter. A low-pass filter is added to remove the double-frequency term.

Table 1: BPSK Demodulator resource usage

BRAM	DSP	FF	LUTs
0	1	256	311

2.4.5.2 Successive Cancellation List decoder

The SCL decoder IP was implemented earlier during the CSE 237C course project. The resource usage of the SCL decoder IP is shown as obtained using the Vivado HLS synthesis tool.

Table SCL Decoder Resource usage and Throughput

Functional Element	BRAM	DSP	FF	LUT	Clock Period(ns)	Interval	Throughput(Hz)
list_scd_unoptimized	7	0	1916	6831	8.59	25237019	4.61
list_scd_optimized	3	0	2373843	921229	8.71	15409691	7.45

3. Milestones:

Initial milestones	Completion	Modified Milestones
Study of Polar Code with different block length and list size for implementation. Study is to be made using MATLAB.	Done	N/A
USRP implementation of Encoder and decoder.	Done	N/A
Sending information via SDRs(wireless interface) and prepare the BER evaluation testbench.	Done	N/A

Change different modulations on the encoder and decoder. Analyze the BER performance of these modulation schemes: BPSK, QAM, OFDM.	Done	N/A
Change the image compression scheme and figure out which is best for the Polar Code based on the BER performance.	Done. But expected results were not observed.	JPEG, PNG image formats were transferred; however there was no improvement in the BER performance. The channel coding treats the input data as random and hence there is no improvement. Thus only results of image transmission is shown instead of a comparison.
Stream video based on the MJPEG and evaluate the BER performance.	Done	MJPEG video encoding was not used. Instead H.264 MPEG-4 format video was streamed using the gstreamer tool present in Linux.
RFNoC implementation of receiver blocks	Partially completed	OFDM based receiver couldn't be fit in the RFNoC along with the SCL decoder IP. Hence BPSK demod and SCL decoder was implemented in Vivado HLS.

Assigned Grades for each milestone in mid quarter:

Grade	Milestone	Completion
B	Implement different modulation schemes and compute its BER performance.	Completed
A	Find the best modulation scheme for Polar code and use that to transmit images on a USRP. The structure of the image needs to be exploited (position of the frozen bits can be modified) using a script.	Completed
A+	Implement the Polar Code encoder/decoder with a modulation scheme on a RFNoC platform and transmit/receive images.	Image and video streaming is done. Implementation of decoder block in the RFNoC is partially done.

4. Conclusion:

The main objective of this project is to study the performance of the polar codes as forward error correction mechanism for wireless video streaming. We modified the polar code encoder and decoder blocks present in the GNU Radio platform to eliminate the CRC check and LLR based computation scheme was added. Also a dynamic frozen bit selection scheme was established in this project even though they didn't provide any significant improvements in improving the bit error rate performance.

Wireless transmitter and receiver blocks were implemented on the GNU Radio / MATLAB platform to enable transmission of information over wireless channel using USRPs. Instead of performing the study over simulated wireless channels in MATLAB, the study was done on real wireless medium.

Polar codes have higher error correction efficiency as the block length increases. Different wireless modulation techniques were implemented using the GNU radio platform to check their performance when used along with Polar Code as the channel coding scheme. It was found that the OFDM with 16-QAM had higher error correction performance.

Images were transmitted using the Polar code as FEC blocks and found that polar coding scheme reduced the errors significantly.

Video streaming was done using the USRPs over the wireless medium and we found that the packet loss were reduced when Polar Code was used as FEC. FPGA implementation of the decoder and demodulation blocks showed improvement in performance.

5. Future Directions:

The implementation done in the GNU Radio platform has lots of room for improvement. Also the FPGA implementation can be optimized further to fit the entire receiver chain and thereby 4K resolution videos can be streamed with less lag.

There are multiple channel decoding architectures proposed by various researchers to be used for polar codes. An analysis of different architectures can be made and an architecture having multiple decoders can be implemented. Thus applications such as VR gaming, video chatting will go through the higher performance low latency architecture and applications such as web browser will go through low performance architecture. Through parallelism higher throughput can be achieved.

References:

[1] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Transactions on Information Theory, 55(7):3051–3073, 2009.
[2] I. Tal and A. Vardy. List decoding of polar codes. IEEE Transactions on Information Theory, 61(5):2213–2226, 2015.
[3] Liu, Jingbo, and Jin Sha. "Frozen bits selection for polar codes based on simulation and BP decoding." IEICE Electronics Express 14, no. 6 (2017): 20170026-20170026.
[4] https://wiki.gnuradio.org/index.php/OutOfTreeModules#Tutorial_1:_Creating_an_out-of-tree_module