International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

Real Time Audio Transmission in CELT using GNU Radio By USRP2

Author1: Mohammad Mosfiqur Rahman
Instructor, Department of Computer Technology
Barisal Polytechnic Institute, Barisal, Bangladesh

Author2: Dr Miftahur Rahman
Professor, Department of Mathematics and Physics
North South University, Dhaka, Bangladesh

Abstract
This paper presents the issue of the innovation policy of real time audio transmission process in CELT codec using GNU radio, a SDR by USRP2. As process technology evolves, processors become more computationally capable pushing the borderline between software and hardware closer to the antenna. To serve the purposes of innovation LPC is necessary for speech coding. The basic foundation of speech coding is to represent the speech signal with the fewest number of bits, while maintaining a sufficient level of quality of the retrieved or synthesized speech with reasonable computational complexity. To achieve high quality speech at a low bit rate, coding algorithms apply sophisticated methods to reduce the redundancies, that is, to remove the irrelevant information from the speech signal. The paper presents a comprehensive assessment of the innovation process that, audio transmission using CELP is a good codec, but for batter performance in real time audio transmission CELT is the one of the best codec cause of its codebook and other resources. In this innovation CELT is the perfect combination with GNU radio for this purposes.

Keyboard : GNU Radio, LPC,CELP, CELT, SDR, USRP2, GMSK, Sampling rate,

Introduction
Most of the wireless system research uses the simulation as an important tool to validate the system performance. The motivation of this topics is to build a flexible test bed for evaluating the novel algorithms under wireless transmission environment. The rapid growth in wireless communication systems demands a technology that is capable of conveying data at high speed and with reliability. The future of communication is wireless, therefore both research and testing focus on improvement of the techniques of wireless transmission. In GNU radio GNU Radio Companion (GRC) is a graphical tool for creating signal flow graphs and generating flow-graph source code. It is an open-source Visual programming language for signal processing using the GNU Radio libraries. The GNU Radio package is provided with a complete HDTV transmitter and receiver, a spectrum analyzer, an oscilloscope, a multichannel receiver and a wide collection of modulators and demodulators. The user interface is called GNU Radio companion or GRC. GNU Radio has several blocks that can generate data or read/write from/to in different formats, like binary complex values or WAV-files. This Graphical User Interface (GUI) can be used to recreate any model based on the need. The GRC helps to easily connect the different modules without the need of using the command line interface or directly writing the python codes.

GMSK (Gaussian Minimum Shift Keying) Modulation is a modulation technique that can provide large data rates with sufficient robustness to radio channel impairments.

GMSK had also achieved popularity for use in commercial high speed broadband wireless systems as the spectrum is utilized more efficiently. GMSK is now being widely implemented in high-speed digital communications. GMSK has been accepted as standard in several wire line and wireless applications. Thus GMSK is the next generation transport technology for wireless communications. GMSK is a special form of multicarrier modulation technique in which the available bandwidth is divided into many narrow sub carriers or sub-channels. This allows many users to transmit in an allocated band in an GMSK system. Each user is allocated several carriers in which to transmit their data. The separation of the sub-carriers is such that there is a very compact spectral utilization. With GMSK, it is possible to have no overlapping sub channels in the frequency domain, thus increasing the transmission rate. (Gina Colangelo) [1]

With a SCA (Software Communications Architecture) implementation like GNU radio project has emerged as one of the most exciting and promising technology. The GNU radio system provides an open source software platform which together with low cost hardware called USRP (Universal Software Radio Peripheral) can be used to develop various software radio applications and implement new technologies for testing purpose. The Software Defined Radio (SDR) allows to bring the code as close to the antenna as possible and because of this it becomes more convenient to be used for academic purposes. The code were generated using C++, Python and XML, which the includes the processes involved in formation of the GMSK signal for transmission in modulation techniques and also demodulation techniques for the received signal. Here is a technique where more than one sub carriers are used to transmit a single data. (Mutsawashe Gahadza , Minseok Kim, Jun-Ichi Takada,)[2]

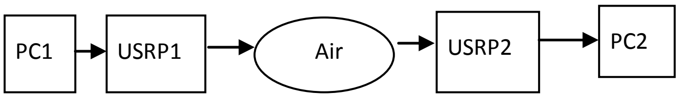

Figure 1 :Communication model with USRP

Implementing the GMSK signal and thus a better signal can be retrieved which would otherwise had been distorted and ultimately lost. Transmitters of this type use GMSK modulation and digital encoding to guarantee protection of transmitted data. Only special receiver, equipped with relevant decoder, can receive signals from such transmitters. Any other receiver provides “white noise” reception only. GMSK is a simple yet effective approach to digital modulation for wireless data transmission. GMSK has been adopted by many wireless data communication protocols. Key advantages include spectral efficiency, low phase distortion and coherence of the signal, it also improves noise immunity when demodulating.

Methods

In most modern paper in the field of audio transmission the methodological concepts competitively as a set of rules and practice followed by codec. **CELT** (Constrained Energy Lapped Transform) is an open, royalty-free audio compression format and a free software codec for use in low-latency audio communication. It is a lossy codec, utput is split in bands approximating the critical bands; . (Dr. Jean-Marc Valin,Gregory Maxwell, and Dr. Timothy B. Terriberry)[3]

Sampling rates from 32 kHz to 48 kHz and above can be use in CELT, adaptive bit-rate from 32 kbit/s to 128 kbit/s per channel and above. It uses ultra-low algorithmic delay (as low as 2 ms; scalable, typically from 3 to 9 ms).

One of the very low delay audio codec CELT designed for high-quality communications. Traditional full-bandwidth codecs such as Vorbis and AAC can offer high quality but they require codec delays of hundreds of milliseconds, which makes them unsuitable for real-time interactive applications like tele-conferencing. Speech targeted codecs, such as Speex or G.722, have lower 20-40ms delays but their speech focus and limited sampling rates restricts their quality, especially for music. Additionally, the other mandatory components of a full network audio system—audio interfaces, routers, jitter buffers— each add their own delay. For lower speed networks the time it takes to serialize a packet onto the network cable takes considerable time, and over the long distances the speed of light imposes a significant delay. In teleconferencing— it is important to keep delay low so that the participants can communicate fluidly without talking on top of each other and so that their own voices don't return after a round trip as an annoying echo. indeed a challenging area in audio codec design, because as a codec is forced to work on the smaller chunks of audio required for low delay it has access to less redundancy and less perceptual information which it can use to reduce the size of the transmitted audio. CELT is designed to bridge the gap between "music" and "speech" codecs, permitting new very high quality teleconferencing applications, and to go further, permitting latencies much lower than speech codecs normally provide to enable applications such as remote musical collaboration even over long distances. In keeping with the Xiph.Org mission—CELT is also designed to accomplish this without copyright or patent encumbrance. Only by keeping the formats that drive our Internet communication free and unencumbered can we maximize innovation, collaboration, and interoperability. . There is also a basic tool for testing the encoder and decoder called68 "testcelt" located in libcelt/: testcelt <rate> <channels> <frame size> <bytes per packet> input.sw output.sw, where input.sw is a 16-bit (machine endian) audio file sampled at 32000 Hz to 96000 Hz. The output file is already decompressed. For example, for a 44.1 kHz mono stream at ~64kbit/sec and with 256 sample frames: testcelt 44100 1 256 46 intput.sw output.sw Since 44100/256*46*8 = 63393.74 bits/sec.All even frame sizes from 64 to 512 are currently supported, although power-of-two sizes are recommended and most CELT development is done84 using a size of 256. The delay imposed by CELT is 1.25x - 1.5x the frame duration depending on the frame size and some details of CELT's internal operation. For 256 sample frames the delay is 1.5x or 384 samples, so the total codec delay in the above example is 8.70ms (1000/(44100/384)).

CELT is already ahead of the competition. Its delay: Configurable, 1.3 ms to 24 ms, ~8 ms typical and quality (at equivalent rates): Much better than G.722.1C, as good as or better than AAC-LD, better than ULD. Its flexibility: 24 kbps to 160+ kbps, 32 kHz to 96 kHz, configurable delay, low-complexity mode The freedom: Open source (BSD), no patents and the transform codec (MDCT, like MP3, Vorbis)

Explicitly code energy of each band of the signal has coarse shape of sound preserved no matter what and code remaining details using vector quantization. Also uses pitch prediction with a time offset, CELT is similar to linear prediction used by speech codecs and helps compensate for poor frequency resolution

CELT is short block transform that only capable of resolving harmonics if the period is an exact multiple of the frame size. For any other period length, the current window will contain a portion of the period offset by some phase. We search the recently decoded signal data for a window that covers the same portion of the period with the same phase offset. While the harmonics will still not resolve into distinct MDCT bins, for periodic inputs the predictor will produce the same pattern of energy spreading. The pitch predictor is specified by a period defined in the time domain and a set of gains defined in the frequency domain. The pitch period is the time offset to the window in the recent synthesis signal history that best matches the current encoding window. We estimate the period using the frequency domain generalized cross-correlation between the zero-padded input window and the last Lp = 1024 decoded samples .

Two of the parameter sets transmitted to the decoder are encoded at variable rate: the energy in each band, which is entropy coded, and the pitch period, which is not transmitted if the pitch gains are all zero. To achieve a constant bit-rate without a bit reservoir, we must adapt the rate of the innovation quantization. We first assume that both the encoder and the decoder know how many 8-bit bytes are used to encode the frame. This number is either agreed on when establishing the communication or obtained during the communication, e.g. the decoder knows the size of any UDP datagram it receives. Given that, both the encoder and the decoder can implement the same mechanism to determine the innovation bit allocation. This mechanism is based solely on the number of bits remaining after encoding the energy and pitch parameters. A static table determines the bit-allocation in each band given only the number of bits available for quantizing the innovation. The correspondence between the number of bits in a band and the number of pulses is given by the [6]. For a given number of innovation bits, the distribution across the bands is constant in time. This is equivalent to using a psychoacoustic masking.

Each band's share of available bits is fixed, specially CELT transmits no side information for allocation and it equivalent to modeling within band masking. The signal-to-mask ratio for each band is roughly constant, so ignores inter-band masking and tone vs. noise effects.

In this communication one pc with Linux Ubuntu have proper setup of GNU Radio in tx side and rx side both. These pc are connected with USRP2 N210 . The communication media was wireless communication with 2.4 GHz frequency.

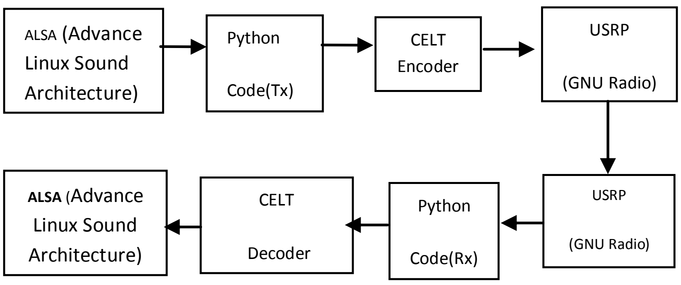

Figure2: Basic Architecture of CELT implemented in Linux platform

In present it is become truth that the USRP-2 (Daughter Board XCVR 2450) is capable to transmit & receive real time audio signal. It becomes true when in GRC the NBFM modulation is used in tx & rx end.

Using narrow band frequency modulation the transmission side of the GRC blocks Audio source, Rational Resampler, NBFM transmit, UHD:UHD Sink are used. There the sample rate of audio source is 48KHz and the sample rate of UHD:UHD Sink is 195.312KHz. And in receiving side the GRC blocks UHD:UHD Source, NBFM Receiver, WX GUI Scope Sink, Rational Resampler, Multiply Constant, Delay, Audio Sink are used. Here sample rate of the UHD: UHD Source is 195.312 KHz and sample rate of audio sink is 48KHz.The output of this transmission & receiver's quality was so poor in quality. Very much noise is in the output signal. Delay, echo is in the signal, so next choice was WBFM.

Using wide band frequency modulation the transmission side of the GRC blocks are Audio source, WX GUI Scope Sink ,Rational Resampler, WBFM transmit, UHD:UHD Sink. There the sample rate of audio source is 48KHz and the sample rate of UHD:UHD Sink is 195.312KHz.In receiving side the GRC blocks UHD:UHD Source, WBFM Receiver, WX GUI Scope Sink, Rational Resampler, Multiply Constant, Delay, Audio Sink are used. Here sample rate of the UHD: UHD Source is 195.312 KHz and sample rate of audio sink is 48KHz.The output of this transmission & receiver's quality is still poor in quality. Still noise is in the output signal, delay, echo is in the signal, so next choice was GMSK.

Using Gaussian Modulation Shift Keying the transmission side of the GRC blocks are TCP source, Packet Encoder, GMSK Mod, UHD:UHD Sink . There the sample rate of packet encoder is 512KHz and the sample rate of UHD:UHD Sink is 195.312KHz . Address of TCP Source is 127.0.0.1.In receiving side the GRC blocks UHD:UHD Source, GMSK Demod , WX GUI Scope Sink, Packet Decoder, TCP Sink are used. Here sample rate of the UHD: UHD Source is 195.312 KHz and address of TCP sink is 127.0.0.1.The output of this transmission & receiver's quality is not so good quality. Still noise is in the output signal, delay, echo are lightly in the signal. These are happened for miss match of GNU radio internal code which automatic generated when circuit is design in the GRC and our code . So there is a opportunity to make it better by using code directly.

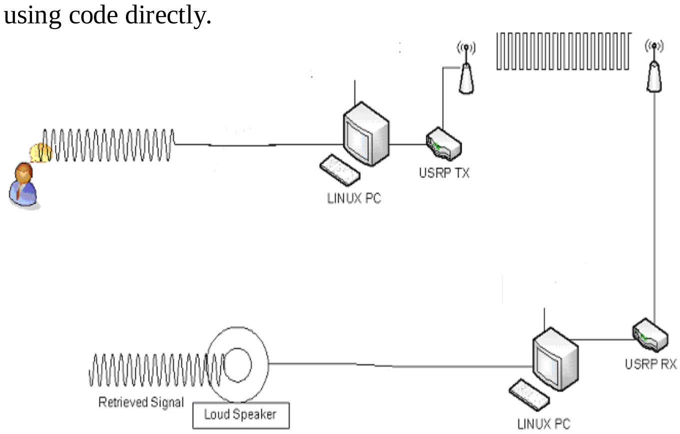

Figure3 : Implemented Structure of GNU Radio

For analysis of CELT performance should have some tasks for

1. Measure the highest energy with respect to normalized energy
2. Measure the fft of .wav file we get in output.
3. Divide others quality respect to the highest quality.
4. Find out the best quality energy.
5. Energy change at the respect of frequency change.

There are two lossy audio codecs that are being used presently. These are classified in two broad classes.:

1. Low delay (15-30 ms) speech codecs that include G.72x,GSM,AMR,Speex with low sampling rate of 8 KHz to 16 KHz with limited fidelity. These codecs do not support music due to low sampling rate. Low delay is a critical factor for live interaction due to low collision rate during conversation and reduced echo cancellation. Low delay codecs are suitable for live music synchronization that requires delay of less than 25ms.[5]

2. General purpose codec's that include MP3, AAC, Vorbis with high delay (>100ms) and high sampling rate of 44.1 KHz and higher. These codec's support CD quality music with higher fidelity.

Therefore in summarization the two above mentioned codec categories, one may observe the following advantages and disadvantages:

i. G.722.1C (ITU-T) with 40 ms delay and upto 32 KHz sampling frequency

ii. AAC-LD (MPEG) with 20-50ms delay and sampling frequency up to 48KHz.

iii. ULD (Franhofer) with delay less than 10ms and sampling frequency up to 48 KHz.

iv. On the other hand CELT is an open source codec with a lot of potential to be competitive with respect to the existing codec. It has configurable delay in the range of 1.3 ms to 24 ms with much better quality than G.722.1C, AAC-LD, and ULD. The data rate range from 24kbps to 160kbps and higher.

Result

In this part the transmitted signal is being transformed using FFT to get the magnitude. Though FFT is in different point so magnitude will act as an energy. Here the audible frequency have much more energy as can be seen from Fig 4 to Fig 8The sampling rate of the original signal is 44.KHz, then it is verified with 8 KHz, 16 KHz, 32 KHz and 41 KHz. The result of energy versus frequency as functions of sampling frequencies of 8 KHz,16 KHz, 32 KHz,41 KHz and 44.1 KHz are shown in Fig 4 to Fig 8 respectively. when we find out these energy variations with different sampling rate then we find out the normalized energy over the sampling rates. The normalized energy variation over sampling rate in GMSK is shown in Figure 9

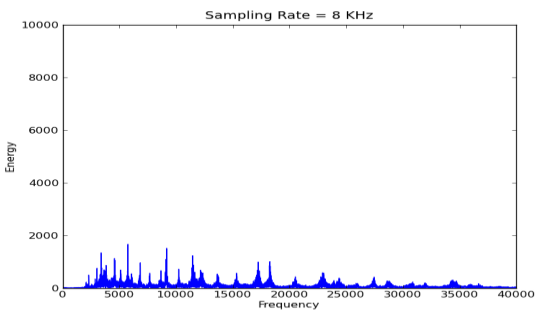

Figure 4: Energy variation with sampling rate=8 KHz

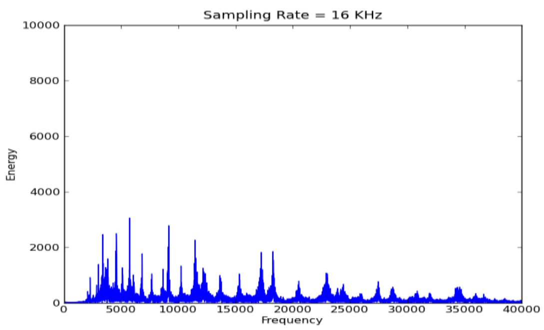

Figure 5: Energy variation with sampling rate= 16 KHz

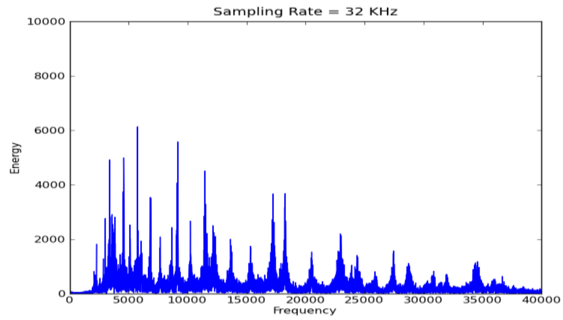

Figure 6: Energy variation with sampling rate 32 KHz

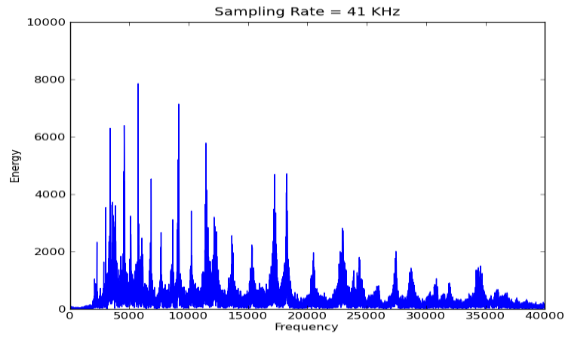

Figure 7: Energy variation with sampling rate 41KHz

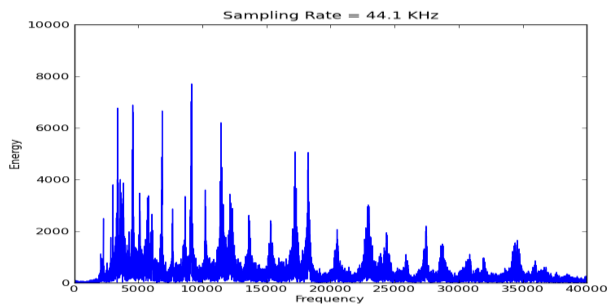

Figure 8: Energy variation with sampling rate=44.1KHz

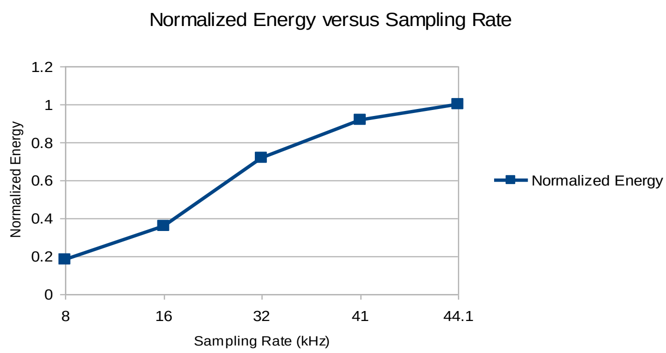

Figure 9 Normalized energy versus sampling rate

In GMSK data always transmit from one side to another. When the modulation period come then there will some small chunk. The number of byte in every chunk is packet size. When a packet size increases then GMSK packet encoding delay will also increases. When packet size is

International Journal of Engineering Research and Technology. ISSN 0974-3154 Volume 10, Number 1 (2017)

small then in GMSK there is a calculation in between header and others tasks of modulation for every part. So throughput become decrease. Here it has been taken a different number of packet size like : 26bytes, 28 bytes, 30 bytes, 32 bytes, 34 bytes, 36 bytes 38 bytes,40 bytes, 42 bytes, 44 bytes, 46 bytes, 48 bytes, 50 bytes. Then we transmit the data and receive it from receiver. The receiver pc shows the maximum throughput when it **received** data after 5 **second** average in each calculation showed in Table 1

Table 1: GMSK maximum throughput

Packet size in Bytes	Maximum throughput in kilobit	Maximum throughput in kilobyte
26	51.63	6.45
28	55.61	6.95
30	44.67	5.58
32	47.65	5.95
34	50.63	6.32
36	53.61	6.70
38	56.58	7.07
40	59.56	7.44
42	62.54	7.81
44	65.52	8.19
46	54.84	6.85
48	57.23	4.15
50	59.61	7.45

After getting the maximum GMSK throughput in this variation of different packet size, we make a chart. In Fig 10 there we get that the upward position of GMSK maximum throughput when packet size is 44 bytes.

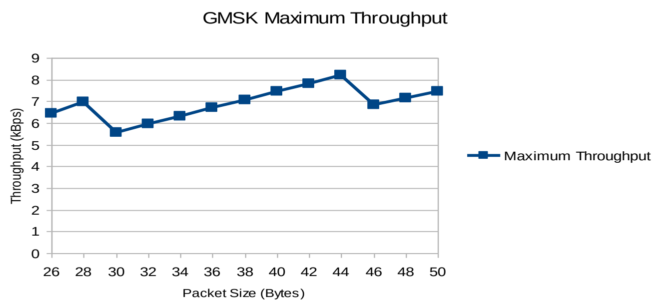

Figure 10: GMSK Maximum Throughput

In Table 2 and also in Fig 11 Bit rate (kbps) versus sampling rate (kHz) is shown with constant packet size(42 bytes) and sample rate is changing to find out the necessary throughput and measure the specific sample rate getting in real time audio transmission in GNU Radio. In this case delay is not available if there is under flow, because when required bit rate is less then achieved bit rate then underflow happen. Achieve bit rate is not flat in a certain point because the maximum achieved bitrates will be in a maximum sampling rate and it is the maximum speed for transmission using GNU radio for a particular packet size

Table 2: Throughput of audio Tx & Rx chain of various sample rate

Packet size constant (Bytes)	Sample Rate	Required Bit rate	Achieved Bitrate for 5 sec average (kBps)	Delay	Underflow
	36k	6.04	5.77	No	Yes
	38k	6.32	6.09	No	Yes
	40k	6.71	6.41	No	Yrs
	41k	6.88	6.57	No	Yes
42	41.5k	6.96	6.64	Yes	No
	42k	7.04	6.73	Yes	No
	44k	7.38	7.05	Yrs	No
	46k	7.72	7.37	Yes	No
	48k	7.99	7.69	Yes	No

.

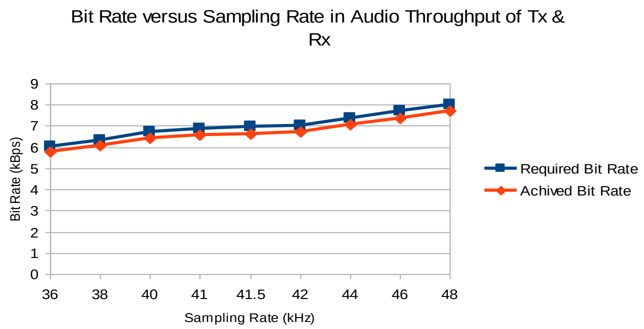

Figure 11: Bit rate vs sampling rate in audio throughput of tx & rx

In Table 3 and in Fig 12 sampling rate is 40 kHz with 256 sample per frame and different size of packet achieved bitrates is nearly similar as well as the required bitrates. For audio throughput sample per frame 256 is a optimum point of compression and delay. From this table we found here that how many data encoded by 256 sample . If the number of byte is increases then quality become well, but we have to transmit more data in network transmission. Here it need such fixed byte per frame which can keep sufficient level of quality and performance.

Table 3: Throughput of audio Tx & Rx chain of various packet size:

Sampling Rate kHz	Samples per frame	Packet size (Byte)	Required bit rate (kBps)	Achived bit rate for 5 sec average (kBps)	Delay	Under flow
		32	**5.12**	**5.13**	**No**	**Yes**
		36	**5.76**	**5.77**	**No**	**Yes**
		38	**6.08**	**6.08**	**No**	**Yes**
		40	**6.40**	**6.41**	**No**	**Yes**
40	**256**	**42**	**6.72**	**6.72**	**No**	**Yes**
		44	**7.04**	**7.05**	**No**	**Yes**
		46	**7.36**	**7.37**	**Yes**	**No**
		48	**7.69**	**7.69**	**Yes**	**No**

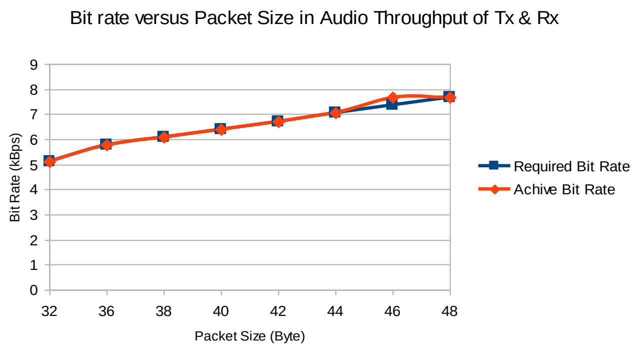

Figure 12: Bit rate vs packet size in audio throughput of tx & rx

In Table 4 the limit for sample per packet of CELT is 64 to 512. If the number of sample per packet increases to 512 then the encoding delay will increase. Here encoding delay means the time to encode a packet means a sound file transmit in the CELT and come back

Table 4: Constant sample rate vs various frame size

Sample per frame	Frame size (Byte)	Performance
256	8	Bad
	16	Poor
	32	Fair
	64	Nice
	128	Good
	256	Best

In Table 5 the fixed sampling rate in different frame size with different packet size the delay is not available when there is underflow. when packet size increasing the delay is coming and underflow has decreasing. If delay is present then echo also present and if delay is absent then echo is also absent.

Table 5: Constant sample rate in different frame size in various packet size

Sampling Rate	Frame size	Packet size (Byte)	Delay	Under flow	Echo
41000	256	16	No	Yes	No
		40	No	Yes	No
		42	No	Yes	No
		44	No	Yes	No
		46	Yes	No	No
		50	Yes	No	Yes
	300	44	No	No	No
		48	No	Yes	No

Discussion

CELT exploits the fact that the ear is mainly sensitive to the amount of energy in each critical band. The MDCT spectrum is thus divided into 20 bands of roughly one critical band each, although the lower frequency bands are wider due to the low MDCT resolution. We refer to these bands as the *energy bands*

CELT is still in an early state of development. At this point, two ways of getting involved are: helping design the algorithm (requires strong DSP knowledge) or building applications using CELT. Our feedback can help define the future direction the codec will take. It applies some of the CELP principles, but does everything in the frequency domain, which removes some of the limitations of CELP. CELT is suitable for both speech and music [4]

There are two program languages used in GNU Radio, C++ and Python which play different roles in the whole system. All the signal processing and performance-critical blocks are written in C++. Python is used to create a network or graph and glue these blocks together.

Conclusion

The real time audio transmission using CELT was done successfully using GNU radio system with its required hardware and software components. The result shows us the significant performance of the model used to transmit audio transmission in CELT. From the results and analysis it is quite clear that the GNU Radio is usable for real life audio transmission model since the results showed that the original signal could be retrieved almost without noises. The model was tested in 2.4 GHz with CELT. CELT brings CD-quality sound to VoIP-style low-delay applications and better than MP3 and <10 ms delay.

Future work in this area might look into the application of GNU Radio in consumer facing applications. Exploring extremely low latency transport-layer-adaptive synchronized audio-video transmission might also be of interest. Aspects of CELT that can be improved include dynamic rate allocation, stereo coupling and pitch prediction. The transport layer software can also be tested at other center frequencies such as 5GHz. Some emerging uses of software defined radio are 4G LTE, WiMax, WiFi, Digital TV, HDTV, mobile TV etc. The application of low latency codecs such as CELT in these areas could lead to groundbreaking results. The combined application of exceptional codecs such as CELT and adaptive communication systems such as GNU Radio can revolutionize the communications sector.

References

I. Gina Colangelo "**Introduction to GMSK: Gaussian Filtered Minimum Shift Keying**" (EE194 – SDR)

II. Mutsawashe Gahadza , Minseok Kim, Jun-Ichi Takada, "**Implementation of a Channel Sounder using GNU Radio Opensource SDR Platform**", Graduate School of Engineering, Tokyo Institute of Technology, 2009.

III. Dr. Jean-Marc Valin,Gregory Maxwell, and Dr. Timothy B. Terriberry "**CELT: A Low-latency, High-quality Audio Codec "**, 2010

IV. Jean-Marc Valin, Member, IEEE, Timothy B. Terriberry, Christopher Montgomery, Gregory Maxwell " **A High-Quality Speech and Audio Codec With Less than 10 ms Delay" 2010**

V. Jean-Marc Valin, Timothy B. Terriberry, Gregory Maxwell "**A FULL-BANDWIDTH AUDIO CODECWITH LOW COMPLEXITY AND VERY LOW DELAY" 2010**

VI. www.gnuradio.org, February, 2011