# EXPERIMENT ON GENERATING PSEUDO G711 / G722 RTP PAYLOAD

**Şifa Serdar ÖZEN[1] Alptekin TEMİZEL[2]**
sifa.ozen@sistas.com.tr, atemizel@metu.edu.tr
**1** Sistas Sayisal Iletisim, Kusbakisi Caddesi 25/1, 34662, Altunizade, Istanbul, Turkey
**2** Graduate School of Informatics, Middle East Technical University, Ankara, Turkey

## Abstract

Testing receiving side elements of Voice over IP (VoIP) communication systems may require generation of controllable multi channel VoIP test packets. In this poster, experiment results of using GPU in generating pseudo multi channel RTP payload will be presented. in order to implement such a packet generation system, pseudo speech is generated using sinusoidal model. Using pseudo speech instead of naive random generated input becomes important in testing context sensitive elements such as voice recording systems or packet loss concealment algorithms. Generated pseudo speech is then encoded with g711a / g711u / g722 encoders in order to produce multi channel RTP payload.

## Overview

Flowchart of such a multi channel packet generating engine can be seen at Figure 1. Sinusoidal synthesis is performed using model suggested by McAulay & Quatieri [1]. Input parameters of this model are amplitude, phase and frequency of sinusoids that will be used. Encoders are chosen to be g711a, g711u and g722. By default, these coders operate on 20ms input data, which corresponds to 160 samples at 8kHz coders (g711a, g711u) and 320 samples at 16kHz coder (g722).

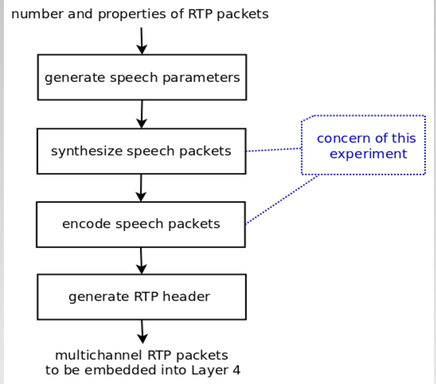



Figure 1. General flowchart of RTP packet generator engine

## Implementation Details

### Sinusoidal synthesis

Sinusoidal synthesis using McAulay & Quatieri model aims to generate synthetic speech using sum of sinusoids [1]. Input parameters of this model are amplitude, phase and frequency of sinusoids that will be used.

$$s[n]=\sum_{k=1}^{L} A_k[n]\cdot\sin(\theta_k[n])$$

In this model phase is assumed to be cubic function of time.

$$\theta[n]=a+b\cdot n+c\cdot n^2+d\cdot n^3$$

Model parameters are usually estimated from STFT analysis of two consecutive frames, named as frame i and i+1. Inter-frame distance is chosen as 20ms to be consistent to default RTP payload length (N = 160 for 8kHz and 320 for 16kHz). Amplitude interpolation is trivial but finding phase parameters need some computation. Starting with boundaries,

$$\theta[0]=\theta_i \qquad \theta[N]=\theta_{i+1}$$
$$\theta'[0]=\omega_i \qquad \theta'[N]=\omega_{i+1}$$

leads to following solution set where M should be chosen to achieve smoothest unwrapped phase possible.

$$a=\theta_i \qquad b=\omega_i$$
$$\begin{bmatrix} c \\ d \end{bmatrix}=\begin{bmatrix} \frac{3}{N^2} & \frac{-1}{N} \\ \frac{-2}{N^3} & \frac{1}{N^2} \end{bmatrix}\cdot\begin{bmatrix} \theta_{i+1}-\theta_i-\omega_i\cdot N-2\cdot\pi\cdot M \\ \omega_{i+1}-\omega_i \end{bmatrix}$$

### g711a encoding

G711a encoding is usually used in Europe at traditional telephone switch systems. Its operation requires 13bit PCM to 8bit 1 to 1 exponential mapping implementing following formula with usually A set to 87.7 ([2] [3])

$$F(x) = sgn(x)\begin{cases} \frac{A|x|}{1+\ln(A)}, & |x|<\frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A}\le|x|\le 1 \end{cases}$$

### g711u encoding

G711u encoding is used in North America and Japan. Its operation requires 14bit PCM to 8bit 1 to 1 exponential mapping using following encoding formula, where u is set to be 255 ([2][4]) Difference between g711a is due to slightly greater dynamic range at cost of increased distortion on smaller signal levels.

$$F(x) = sgn(x)\ \frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \qquad -1\le x\le 1$$

### g722 encoding

G722 is known as 7kHz wide band audio coder operating at 64kbps and usually used at LAN where there is availability for higher bandwidth to yield enhanced signal quality. Coder uses sub-band adaptive differential pulse code modulation to operate and may also be used in three different modes to carry data with audio ([5]). In this work, mode 1 (no data) operation is assumed where all 8bits corresponds to audio.

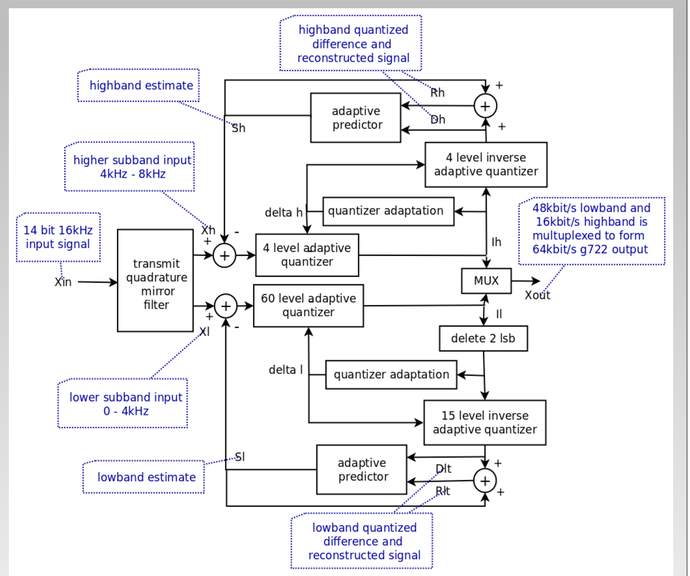



**Figure 2. G722 encoder block diagram**

## Experiment

Experiments are performed in a Debian 7.5 Wheezy 64-bit system with 4GB RAM, using AMD Athlon core for CPU calculations and GTX650Ti Boost based graphics card for GPU calculations. Execution times are calculated to reflect all necessary operations, taken from the host side as a function call execution time (in case of CUDA, a wrapper function is called that performs device operations which includes all memory transfers and kernel executions). For g722 case execution time also includes transfer of state vectors to and from GPU.

All experiments were performed with randomly initialized input data and repeated 100 times. Displayed values are the mean values of those simulations. Iterations that can run at real time (execution time less than 20 ms) are displayed in bold fonts. Reference CPU code of g711 and g722 encoders can be found at [6]. In this work, modified and more or less optimized versions are used, and GPU versions are prepared which can be seen at github [7]. Experiment flow may be seen at Figure 3.

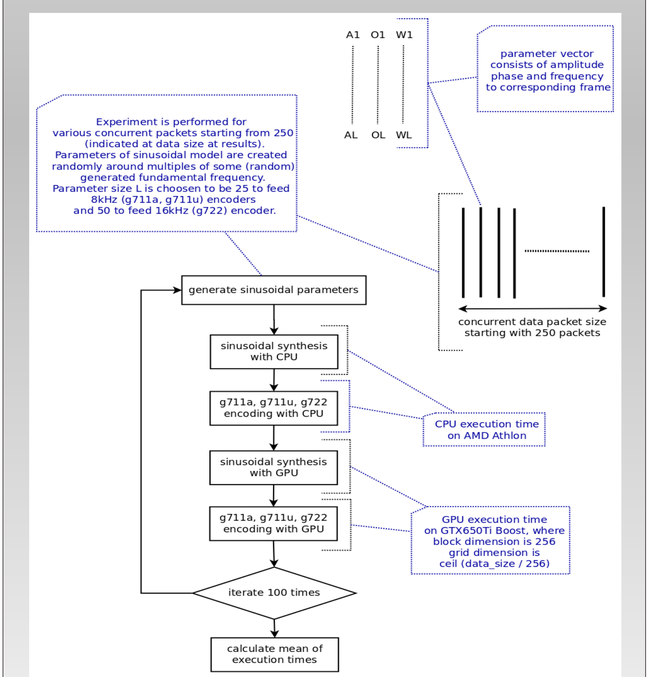
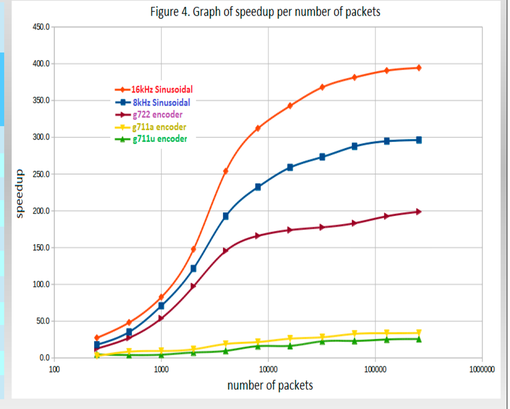



**Figure 3. Experiment flowchart**

## Result of Experiment

| | 8kHz sinusoidal synthesis | | | 16kHz sinusoidal synthesis | | | g711a encoding results | | | g711u encoding results | | | g722 encoding results | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Data size** | **CPU execution time (s)** | **GPU execution time (s)** | **speed up** | **CPU execution time (s)** | **GPU execution time (s)** | **speed up** | **CPU execution time (s)** | **GPU execution time (s)** | **speed up** | **CPU execution time (s)** | **GPU execution time (s)** | **speed up** | **CPU execution time (s)** | **GPU execution time (s)** | **speed up** |
| 250 | 0.0991 | **0.0055** | 18.0 | 0.3740 | **0.0137** | 27.3 | **0.0021** | **0.0008** | 2.6 | **0.00190** | **0.0004** | 4.8 | 0.1278 | **0.0100** | 12.8 |
| 500 | 0.1977 | **0.0056** | 35.3 | 0.7568 | **0.0157** | 48.2 | **0.0043** | **0.0005** | 8.6 | **0.00320** | **0.0008** | 4.0 | 0.2576 | **0.0094** | 27.4 |
| 1000 | 0.3969 | **0.0056** | 70.9 | 1.5214 | **0.0184** | 82.7 | **0.0076** | **0.0008** | 9.5 | **0.00670** | **0.0015** | 4.5 | 0.5164 | **0.0096** | 53.8 |
| 2000 | 0.7903 | **0.0065** | 121.6 | 2.9986 | 0.0203 | 147.7 | **0.0165** | **0.0014** | 11.8 | **0.01190** | **0.0016** | 7.4 | 1.0322 | **0.0106** | 97.4 |
| 4000 | 1.5802 | **0.0082** | 192.7 | 5.9906 | 0.0236 | 253.8 | 0.0323 | **0.0017** | 19.0 | 0.02510 | **0.0026** | 9.7 | 2.0662 | **0.0142** | 145.5 |
| 8000 | 3.1593 | **0.0136** | 232.3 | 11.9861 | 0.0384 | 312.1 | 0.0650 | **0.0030** | 21.7 | 0.04980 | **0.003**1 | 16.1 | 4.1345 | 0.0249 | 166.0 |
| 16000 | 6.3199 | 0.0244 | 259.0 | 23.9965 | 0.0700 | 342.8 | 0.1285 | **0.0049** | 26.2 | 0.09900 | **0.0060** | 16.5 | 8.2788 | 0.0476 | 173.9 |
| 32000 | 12.6518 | 0.0463 | 273.3 | 47.9875 | 0.1304 | 368.0 | 0.2591 | **0.0092** | 28.2 | 0.19720 | **0.0087** | 22.7 | 16.6116 | 0.0935 | 177.7 |
| 64000 | 25.3116 | 0.0880 | 287.6 | 96.2932 | 0.2525 | 381.4 | 0.5173 | **0.0158** | 32.7 | 0.39430 | **0.0170** | 23.2 | 33.2639 | 0.1816 | 183.2 |
| 128000 | 50.6252 | 0.1718 | 294.7 | 194.3070 | 0.4974 | 390.6 | 1.0363 | 0.0308 | 33.6 | 0.79000 | 0.0314 | 25.2 | 66.7991 | 0.3467 | 192.7 |
| 256000 | 101.1623 | 0.3415 | 296.2 | 390.2581 | 0.9895 | 394.4 | 2.0734 | 0.0611 | 33.9 | 1.58212 | 0.0614 | 25.8 | 135.3841 | 0.6812 | 198.7 |




## Conclusions

- Generating multi channel pseudo RTP packet is an inherently parallel task and usage of GPU decrease process times significantly even in low compute complexity parts such as g711a and g711u encoders.
- Speedup increases when compute complexity increases. Sinusoidal synthesis and g722 encoding is much complexer processes than g711 encoding.
- Sinusoidal synthesis and encoding results are calculated isolated. In real time continuous systems, old states for sinusoidal synthesizers and g722 encoders may be kept in GPU. Besides, output of sinusoidal synthesis will better feed directly encoder without being transferred to host. This will roughly half memory transfers so yield better result.
- Use of sinusoidal synthesis enables possible future integration to text-to speech systems for better fidelity.

## References

[1] Speech analysis/synthesis based on sinusoidal representation, McAulay, R., Quatieri, T.F. IEEE Transactions on ASSP, 1986
[2] ITU-T G.711: Pulse code modulation (PCM) of voice frequencies, www.itu.int/rec/T-REC-G.711
[3] http://en.wikipedia.org/wiki/A-law_algorithm
[4] http://en.wikipedia.org/wiki/M-law_algorithm
[5] ITU-T G.722: 7kHz audio-coding within 64kbit/s, www.itu.int/rec/T-REC-G.722
[6] ITU-T G.191: Software tools for speech and audio standardization, www.itu.int/rec/T-REC-G.191
[7] https://github.com/sifaserdarozen/GenIt