Derek Wong
5516117
2-12-14

CS 176B HW 2 Report – Audio Streaming

Introduction

This report is an analysis of the effects on various types of audio sent through a simulator program written in Java. This program tries to simulate sending audio packets through a network. This is done in the program by reading in an audio file and then storing incremental information into a buffer, which acts as a packet. Then, a random percentage determines whether or not the packet successfully transmits within a given threshold. Any packet that makes it through will be written to file. Any packets that are not within the threshold will have their missing information written to file using other alternatives. Three files that have gone through this simulator program will have their results discussed in this report.

Experiment Variables

Packet Size – This is a preset buffer size that also acts as the packet size for the experiment. It is given in bytes. Because the Java AudioSystem library was used, the data is transmitted in frames. The amount of bytes per frame is dependent on the file given. The actual buffer size is then (bytesPerFrame*packetSize). The three sizes tested are 10, 100, and 1000 bytes.

Threshold – Set in the simulator as a double, representing the percentage of packets that should get through. This is tested against a loss calculation that has a maximum possible number of 100.0. Therefore, if the threshold is 100.0, all packets will make it through. If set to 90.0, 90% of packets will make it through, and so on. In this experiment, the threshold values will be set to less than 100, since we are trying to test the effects of lost packets. The three percentages tested are, 90%, 75%, and 50%.

Type – The type determines what will be written to file if the packet does not pass the threshold and is not transmitted. There are three types in this experiment:

1 - Play Silence. The sent packet is filled with zeros.
2 - Replay last value. The last value from the last buffer is repeated in a packet and written to file.
3 - Replay Last Packet. The last packet in the buffer is sent.

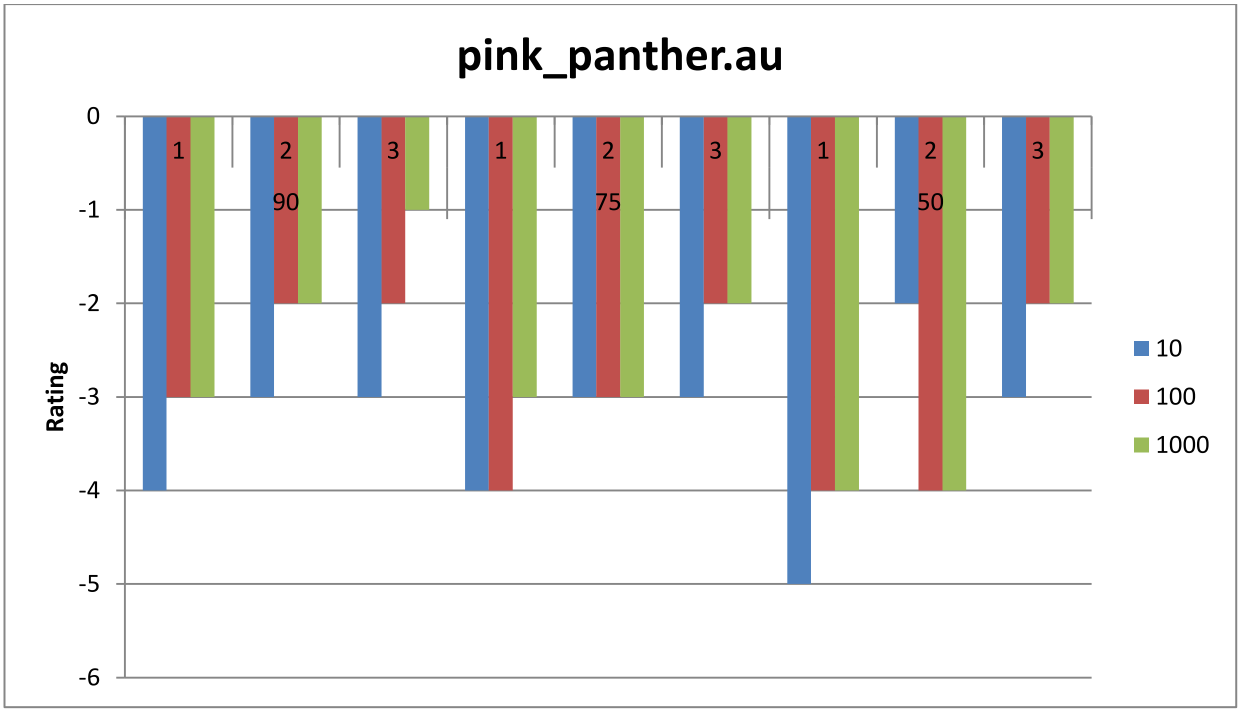
Rating – The quality of the received audio is measured with a modified subjective Likert Scale. This scale has values from 0 to -5 to represent diminishing audio quality.

0 – Audio quality indistinguishable from original file. Implies perfect transmission.
-1 – Slight static or distortion of original. Otherwise identical to original.
-2 – Noticeable static and distortion. Underlying audio can still be made out.
-3 – Significant distortion. Original audio can be barely made out.
-4 – Heavy distortion and noise. Original audio cannot be made out.
-5 – Complete noise and distortion. No original audio left.

Results

Audio Clip 1 – pink_panther.au (Music: Pink Panther Theme, 899KB)

	pink_panther.au								
	90% Threshold			75% Threshold			50% Threshold		
	Type 1	Type 2	Type 3	Type 1	Type 2	Type 3	Type 1	Type 2	Type 3
10 B	-4	-3	-3	-4	-3	-3	-5	-2	-3
100 B	-3	-2	-2	-4	-3	-2	-4	-4	-2
1000 B	-3	-2	-1	-3	-3	-2	-4	-4	-2



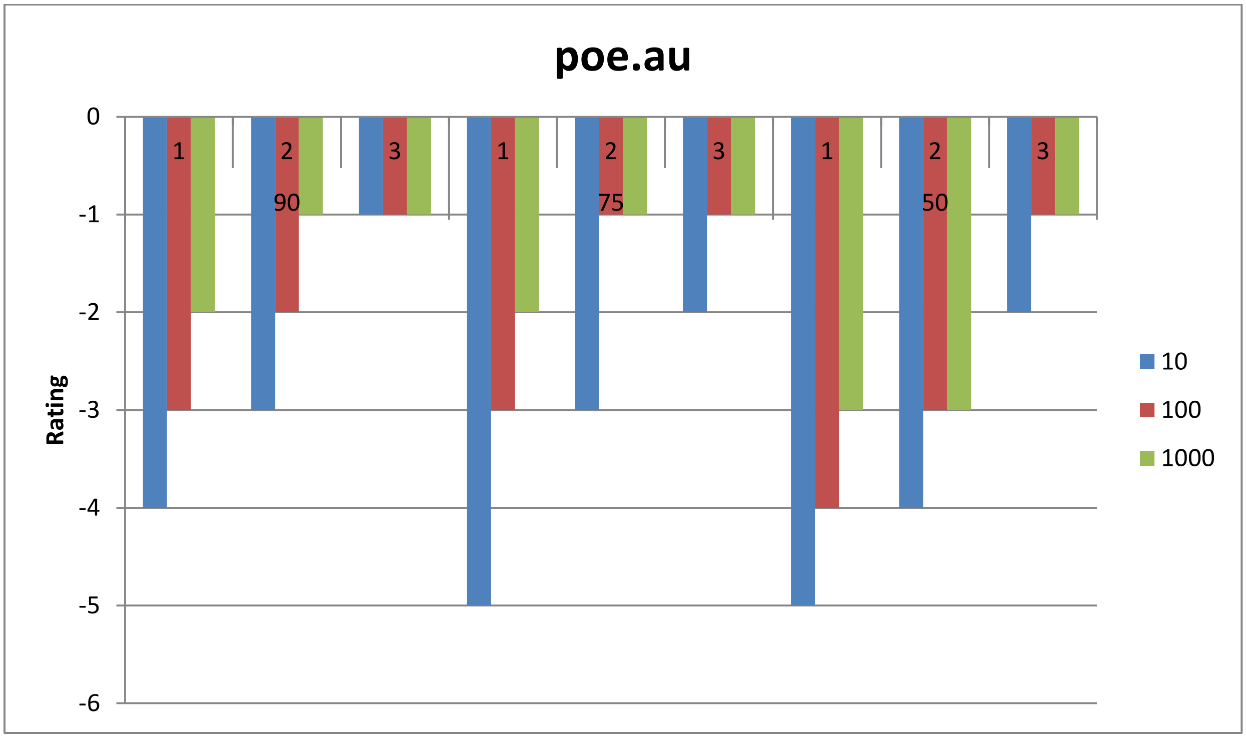
Overall, audio quality suffered as expected as the threshold became lower and more packets were lost. Most of the noise and distortion was laid over the original track. While the background audio could be heard in most cases to reflect the original, the noise seemed to manifest as an overlay. For a packet size of ten, the audio was mostly static that rose with volume. The small packet size seemed to affect audio quality the most across all three thresholds.

For the 100 byte packets, the static turned into intermittent popping that overlaid the original audio. This turned into intermittent stops and gaps for the 1000 byte packets. Overall there was small change in perceived quality across the thresholds for the different types used. The differences in type were carried over regardless of packet size within each threshold group.

Type 3 substitution (writing the last packet) was the most effective means of audio quality across all thresholds for this test. Although there was noticeable skipping, the quality was maintained and consistent, sounding better than the audio produced using the other type methods.

Audio Clip 2 – poe.au (Narration: The Raven, 2272KB)

	poe.au								
	90% Threshold			75% Threshold			50% Threshold		
	Type 1	Type 2	Type 3	Type 1	Type 2	Type 3	Type 1	Type 2	Type 3
10 B	-4	-3	-1	-5	-3	-2	-5	-4	-2
100 B	-3	-2	-1	-3	-1	-1	-4	-4	-1
1000 B	-2	-1	-1	-2	-1	-1	-3	-3	-1



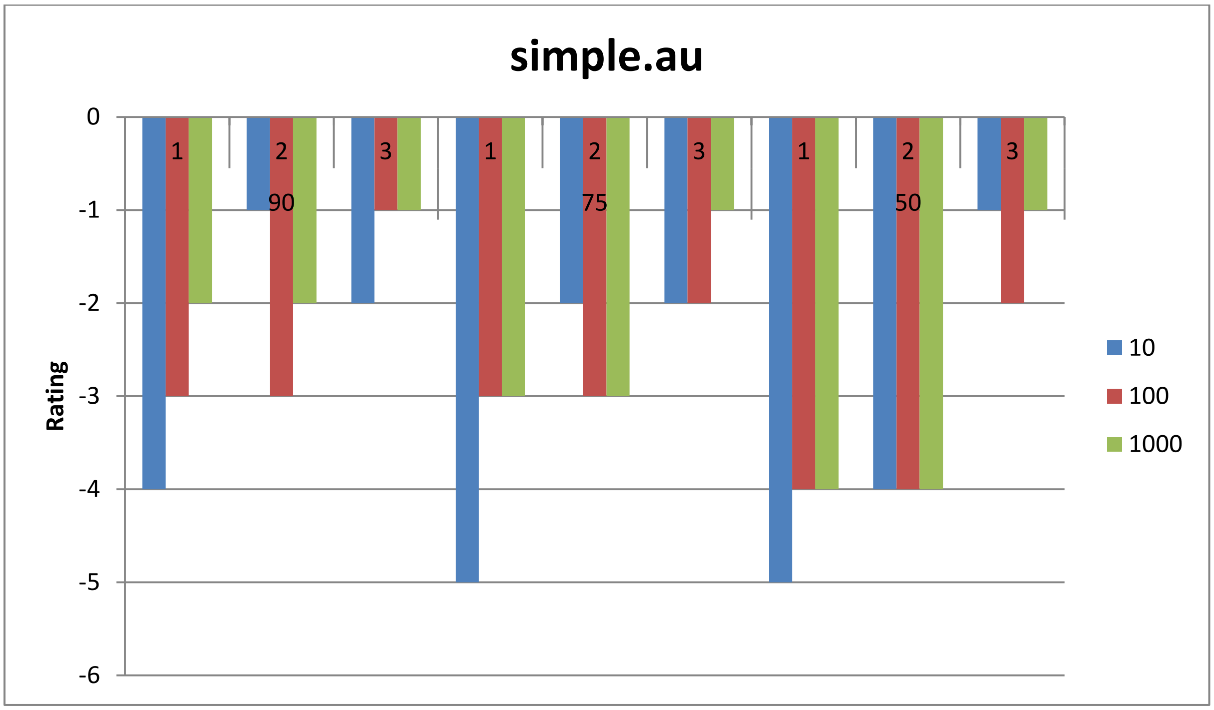
Like the pink panther results, quality suffered as the threshold decreased. Noise generally decreased proportionally along with the type used. Unlike the pink panther results, the poe.au narration suffered from high underlying static that was not linked to any music. This was especially true for the entire row of ten byte packet transmissions. Again, small packet size was the biggest factor in the quality of the audio, independent of the threshold used.

For the 100 and 1000 byte rows, the static turned into noticeable popping, with slight skipping. The skipping was very short and intermittent, with each individual disturbance not adversely affecting the speech of the narrator. Overall words were still coherent and understandable. The sound quality most reflected a weak radio signal rather than a digital modification.

Again, type 3 substitution was the most effective in preserving audio quality. Compared to the panther sample, all tests utilizing type 3 produced better quality independent of both the threshold or packet size. Even when half of packets were missing, the narration was still understandable, even though it was noticeably repeating.

Audio Clip 3 – simple.au (Music – Simple Minds – Don't You (Forget About Me), 2010KB)

	simple.au								
	90% Threshold			75% Threshold			50% Threshold		
	Type 1	Type 2	Type 3	Type 1	Type 2	Type 3	Type 1	Type 2	Type 3
10 B	-4	-1	-2	-5	-2	-2	-5	-4	-1
100 B	-3	-3	-1	-3	-3	-2	-4	-4	-2
1000 B	-2	-2	-1	-3	-3	-1	-4	-4	-1



These results are consistent with the general trends seen in the other two tables. Once again, the smaller packet size was the greatest indicator of adverse audio quality. Combined with the type 1 method, the results are consistent with the other two tables across the different thresholds. This table is most similar to the poe.au narration, even though they vastly differ in terms of content.

Much like the poe.au narration, this test also showed signs of underlying static, especially with the first row of ten byte packets. The static was over most of the original song, which was barely audible, and became incoherent with the type 1 method. As the threshold increased for type 1, gaps in the audio also began to appear. Again, like the poe.au narration, this most reflected a weak radio signal.

Like the pink panther theme, type 3 substitution produced generally good audio quality, even as parts of the song were repeated. This was just as noticeable as the repeating in the poe.au narration. However, the repeating in the song was more tolerable since it followed a predictable audio pattern. It was easy to anticipate when the chorus, drums, or guitar was playing, so a repeat of these elements was much easier to listen to.

Analysis

Between all three results, there were common predictors for general audio quality. A smaller packet size contributed most to noise and static because of the extremely high frequency at which the audio was broken up. This resulted in the radio noise effect, which is what a song might sound like if it was heard with poor signal reception on the radio. This was common with both type 1 and type 2, independent of packet size.

The second biggest contributor to degradation was the threshold. This was the most expected and easily predictable effect that could be observed with the data. No matter the type method used to replace the lost packets, audio quality generally suffered because the method was being employed at a much higher rate to replace the original audio. This is most evident in the music audio clips, which requires an expected cadence to the songs.

However, the type method used offered the most variance in terms of quality. Both type 1 and type 2 simply caused gaps in the received audio, while type 3 allowed for very noticeable stuttering repeats. In terms of fidelity, type 3 produced the most consistent audio, at the cost of continuous repeat. Type 1 and type 2 produced much more coherent and complete audio, at the expense of quality.

Conclusion

The simulator for this project was able to produce varied results given the different conditions that were used. Many of the reported factors appeared to be consistent across the tests based on subjective quality alone. In the end, this presents the biggest issue for deciding which replacement method is the best. Type 1 and Type 2 replacement offers coherence and completeness at the expense of quality, while Type 3 trades coherence for higher quality for individual frames. Whichever one is better at handling audio is a subjective opinion that depends on the listener. It can be argued that one method is better suited over another based on the type of audio being played. For example, intermittent static might be best for live streaming music, while stuttering repeats could be best for live narration. This assumes that static is a better indicator of poor connection rather than continuously repeating audio as a form of feedback.

Other replacement methods not implemented in this project for dealing with lost packets might be to buffer incoming packets in their intended order, send smaller and more packets, or to not play something until it is fully downloaded. However, these present other problems which must be experimented on and addressed.