Abstract

In this experiment the goal is to simulate different scenarios that could affect the quality of audio files sent over a network.

Intro/Background

The experiment consists of a variety of simulations with a variety of different parameters. Some of the varying parameters of the experiment were: The size of the packets that were sent, the rates (probability) that packets can be dropped, and the different methods/policies to take on the receiver end when a packet fails to arrive in time for playback.

Methodology

For this simulation, the chosen programming language was Python. I made use of sunau library to read and write audio files with the au file format. The pseudocode for the main part of the application is as follows:

```
main():
    au_reader.open()
    au_writer.open()
    au_writer.header = au_reader.header
    loss_rate= get_rate()
    packet_size= get_size()
    while not au_reader.eof():
        data= au_reader.readsample(packet_size)
        loss = chance_of_dropping()
        if (loss < loss_rate):
            au_writer.write(data)
        else:
            au_writer.use_alternate_policy()
    au_reader.close()
    au_writer.close()
```

As per the simulation requirements I created three policies to deal with packet loss. The pseudocode for these policies is as follows:

```
silence_policy(packet_size):
    if packet_size > available_space:
        packet_size= available_space
```

```
        write_null_bytes(packet_size)

repeat_packet_policy(packet_size):
        if at_pos_zero():
                silence(packet_size)
        if packet_size > available_space:
                packet_size = available_space
        data= get_last_sent_sample(packet_size) #Last packet sent
        write_data(data)

repeat_frame_policy(packet_size):
        if at_pos_zero():
                silence(packetsize)
        if packet_size > available_space:
                packet_size= available_space
        sample= get_last_sent_sample(1) #last sample (1 byte)
        data= sample * packet_size
        write_data(data)
```

Some of the packet_sizes that I based my observations on were: 50, 500, 1500 and 3000 byte packets. As for the loss_rate values I utilized, they were: 3%, 10%, 25%, 50%, 70%

Results

Voices Mean Opinion score

Packet Size (bytes)	Loss rate%	Policy	**Intelligible**	**Enjoyable**
50	3	Silence	5	5
50	3	repeat_sample	5	5
50	3	repeat_package	5	5
50	25	Silence	4	4
50	25	repeat_sample	4	4
50	25	repeat_package	4	4
50	50	silence	3	2
50	50	repeat_sample	3	2

50	50	repeat_package	4	4
500	3	silence	4	4
500	3	repeat_sample	4	4
500	3	repeat_package	4	4
500	25	silence	3	2
500	25	repeat_sample	3	2
500	25	repeat_package	4	4
500	50	silence	1	1
500	50	repeat_sample	2	1
500	50	repeat_package	4	3
3000	3	silence	4	3
3000	3	repeat_sample	4	4
3000	3	repeat_package	4	4
3000	25	silence	2	1
3000	25	repeat_sample	2	1
3000	25	repeat_package	3	2
3000	50	silence	1	1
3000	50	repeat_sample	1	1
3000	50	repeat_package	3	1

Music Mean Opinion Score

Packet Size (bytes)	Loss rate%	Policy	**Intelligible**	**Enjoyable**
50	3	Silence	5	5
50	3	repeat_sample	5	5
50	3	repeat_package	5	5
50	25	Silence	4	4
50	25	repeat_sample	4	3
50	25	repeat_package	4	3
50	50	silence	3	2
50	50	repeat_sample	3	2
50	50	repeat_package	4	3
500	3	silence	4	3
500	3	repeat_sample	4	3
500	3	repeat_package	4	4
500	25	silence	4	2
500	25	repeat_sample	4	2
500	25	repeat_package	4	4
500	50	silence	2	1
500	50	repeat_sample	2	1
500	50	repeat_package	4	3
3000	3	silence	4	3
3000	3	repeat_sample	4	3
3000	3	repeat_package	5	4
3000	25	silence	3	1
3000	25	repeat_sample	4	2

3000	25	repeat_package	3	3
3000	50	silence	1	1
3000	50	repeat_sample	3	3
3000	50	repeat_package	3	2

As expected, I found that the audio sounded ‘worse’ the higher the loss rate was for a particular simulation. At low loss rates (between 1-5%) there was no discernible decrease in audio quality regardless of the policy selected, but as loss rate increased repeat_policy appeared to be the best fit for mid size packets.

At small packet sizes (100 bytes or less) and low loss rates (less than 8% loss) the differences between different policies were minute. At mid size packet sizes (about 500 bytes) and low loss rates (10% less) the repeat_packet policy is noticeably better than the rest, whereas the other two are almost identical with both making sudden pauses. At larger packet sizes (1000 bytes or more) and higher loss rates (25% and up) repeat_packet policy decreases the quality of the audio as the packets are big enough to lead to repeated words/letters. The other two are not any better as they add a lot of noise and pauses.

Conclusion

Based on the results of my experiment, it appears that as the packet loss percentage increases, it is best to use smaller packet sizes to keep audio quality reasonable. With a smaller packet size using repeat_packet or repeat_sample might be small enough so that the lost packet is not much different than the previous one received and will not affect playback. Not every policy or packet size works best for all types of audio. From the experiment it appears that music and voice recordings have different ideal scenarios and configurations that improve audio quality.