ASSIGNMENT 2

"Impact of Network Loss on Multimedia"

Antonio Javier Samaniego Jurado
PERM: 6473680
CS176B

Abstract

In real data transmission and network protocols, it is impossible to achieve a 100% effective communication, that is, there is noise and other factors that affect the data resulting in data loss.
To solve this, data recovery methods can be used, so that both sent and received messages can be fairly similar, or at least similar enough to have a good understanding of it, regardless of the format.

Introduction

The purpose of this assignment is to implement the data recovery methods mentioned in the Abstract section of this report by coding a simulator that can model a network transmission and perform 3 different packet recovery alternatives depending on the packet size and loss rate.

Methodology

Using the following test audio files:

Files tested	*Title/Description*	*Type*
exclusive federer world no.1 interview.au [1]	Roger Federer interview after becoming oldest ATP number 1 in history.	Spoken word
pink_panther.au [2]	Pink Panther Soundtrack	Low fidelity musical
simple.au [2]	Poe's "The Raven"	High fidelity music with audio

I have implemented such simulator in Matlab, so that we can get numerical and visual results that are going to be evaluated based on the following parameters and control variables:

Assessment Factor	*Rating (1-5)*	*Criteria*
Intelligible	1	Lack of human understanding.
	2	Context and sentences cannot be discerned.
	3	Full words lost. Foreknowledge of what's being said is increasingly required.
	4	No understanding loss, good context.
	5	Words are perfectly clear
Enjoyable	1	Annoying
	2	Not enjoyable, too many pauses/breaks.
	3	Noticeable errors, enough to consider switching channels.
	4	Acceptable/infrequent deviations.
	5	Very enjoyable, great entertainment.

Signal Highest-Energy Frequencies	*Criteria*
Low	Higher frequencies/finishing touch of the sound is missing.
Medium	Medium frequencies/finishing touch of the sound is missing.
High	Lower frequencies/bass is missing.
Whole spectrum	Rich sound.

Control Variables	*Values Tested*
Packet Size (Bytes)	4-500 (Small), 1000-3000 (Medium), 6000-10000 (Large)
Loss Rates (%)	0-25 (Small), 25-50 (Medium), 50-75 (Big)
Packet Loss Reconstruction/ Concealment Methods	Alternative 1: Playing silence (writing all 0s). Alternative 2: Replaying last sample. Alternative 3: Playing the entire last packet received.

All of it using an algorithm that basically takes a specified packet (of different sizes) from the message (audio file), models packet loss through randomization (as the loss in over the network is mostly modeled as random noise), and either store such packet on the output (received message) or process/recover the data and then storing such alternate data, using a threshold based on the specified packet loss to make one or another decision.

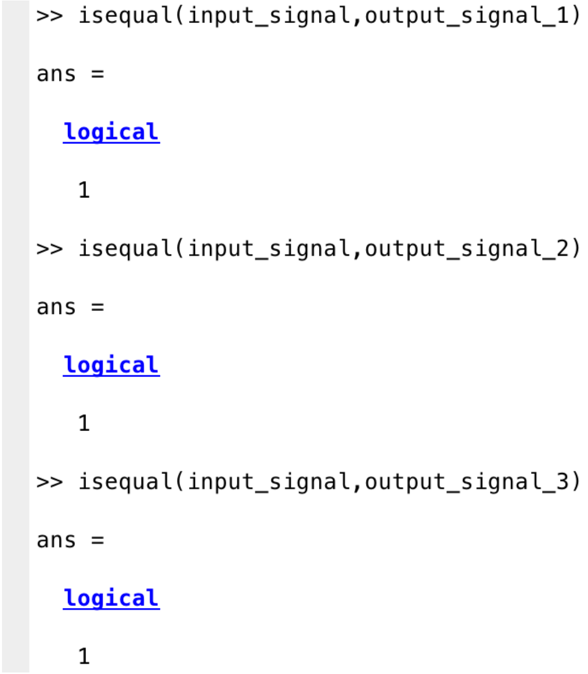
Basically, packet size and loss rate can be changed in my program to model different scenarios.

Results and DISCUSSION

After computing the previously explained algorithm in order to perform the simulation for the three methods, I now show the results and discuss what and why they are so.

Firstly, let's take a look at the extreme cases, in order to get a better understanding of what is happening:

When the loss rate is 0%, that means input and output are exactly the same, which is something we can simulate but is impossible in practice. For such case, after simulating with loss_rate=0, I have checked whether or not "input_signal" and "output_signal" are exactly equal, using Matlab in-built function "isequal", as follows:

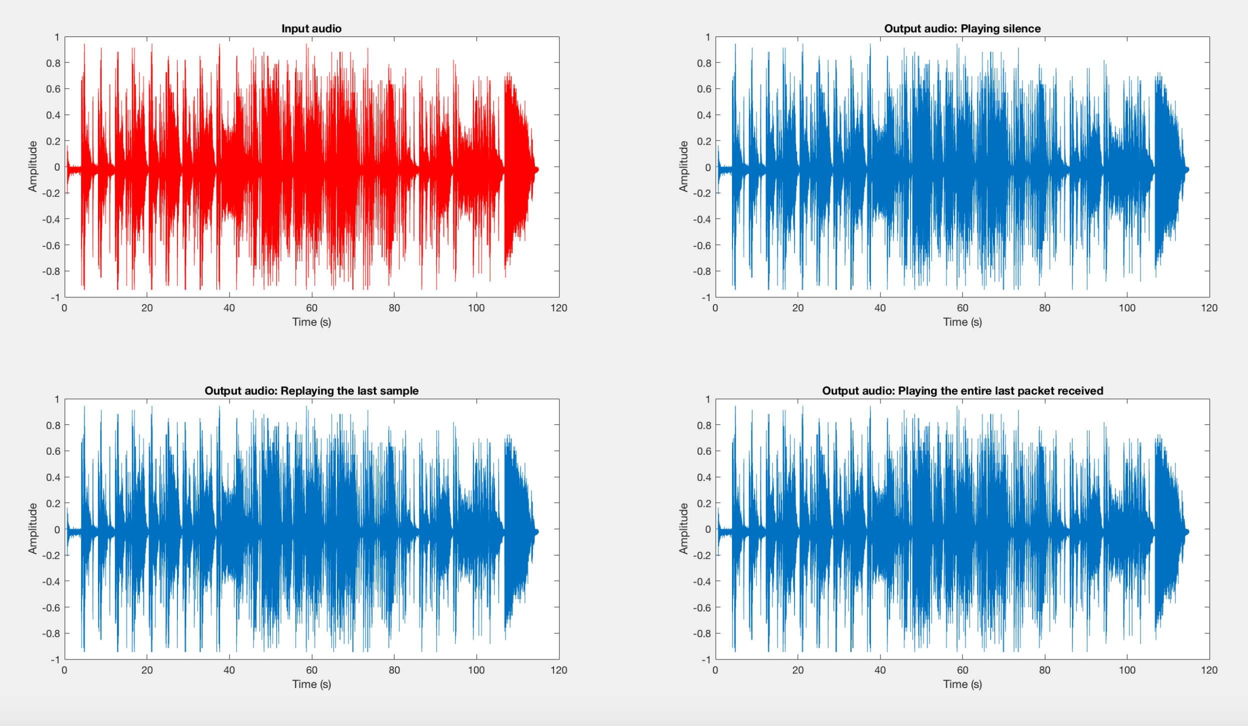


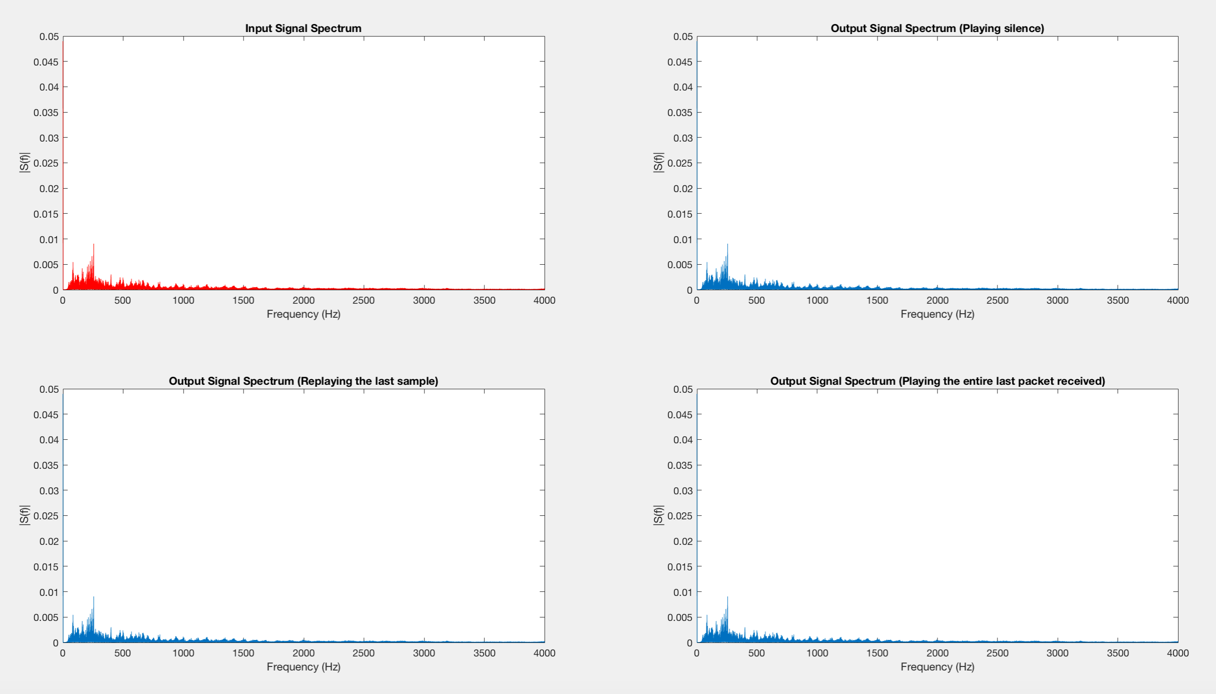
```
>> isequal(input_signal,output_signal_1)

ans =

  logical

   1

>> isequal(input_signal,output_signal_2)

ans =

  logical

   1

>> isequal(input_signal,output_signal_3)

ans =

  logical

   1
```

Such function turned out to return True (logical 1) for the three methods, which proves both input and output are the same.

Moreover, if we take a look at input and output in the time and frequency domains when the loss rate is both 0% and 100% (extreme cases), we get the following results (using "pink_panther.au" as test file):

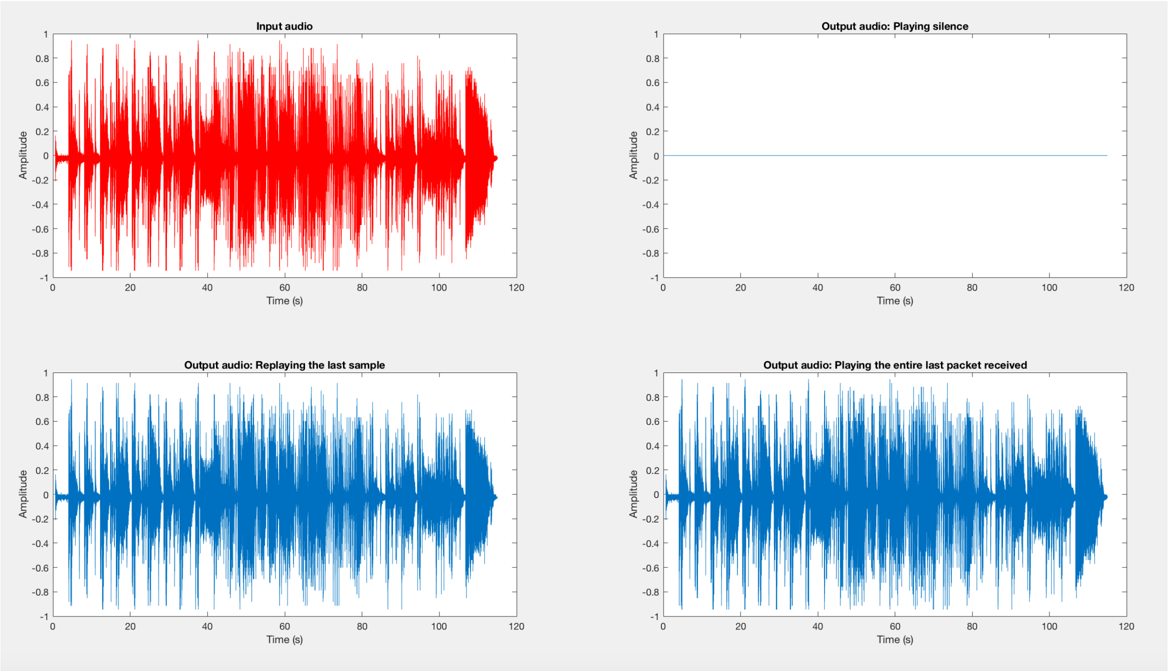
▪ Loss Rate of 0%

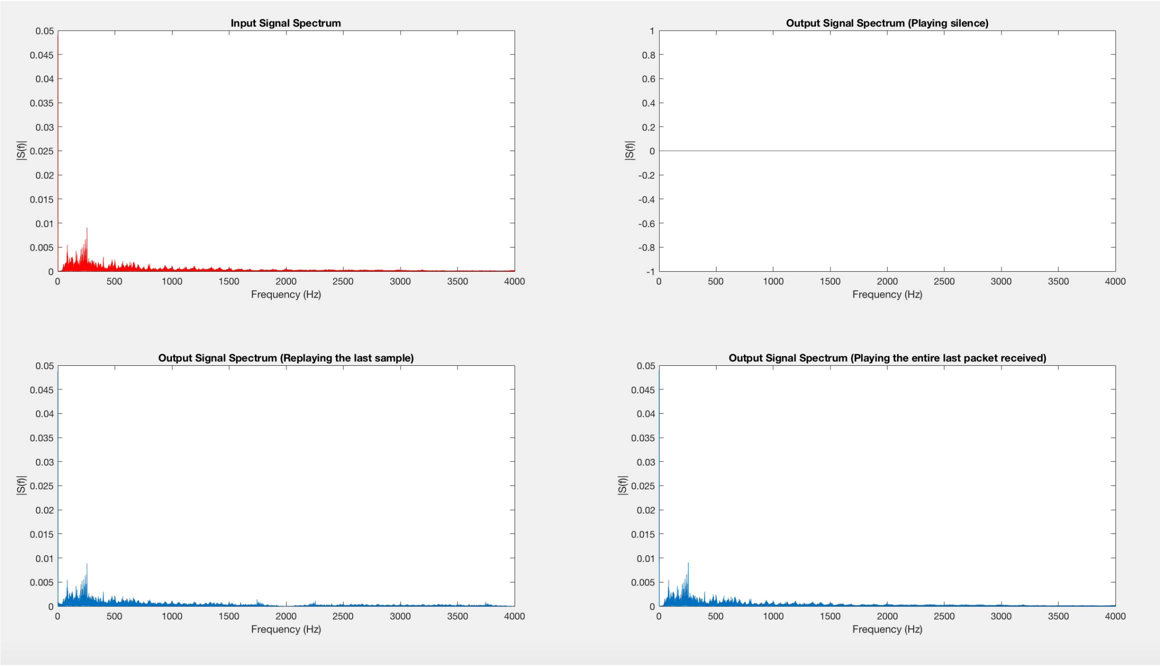




Again, we can see how in both domains when there are no packets lost, input and output look exactly the same.

· Loss Rate of 100%





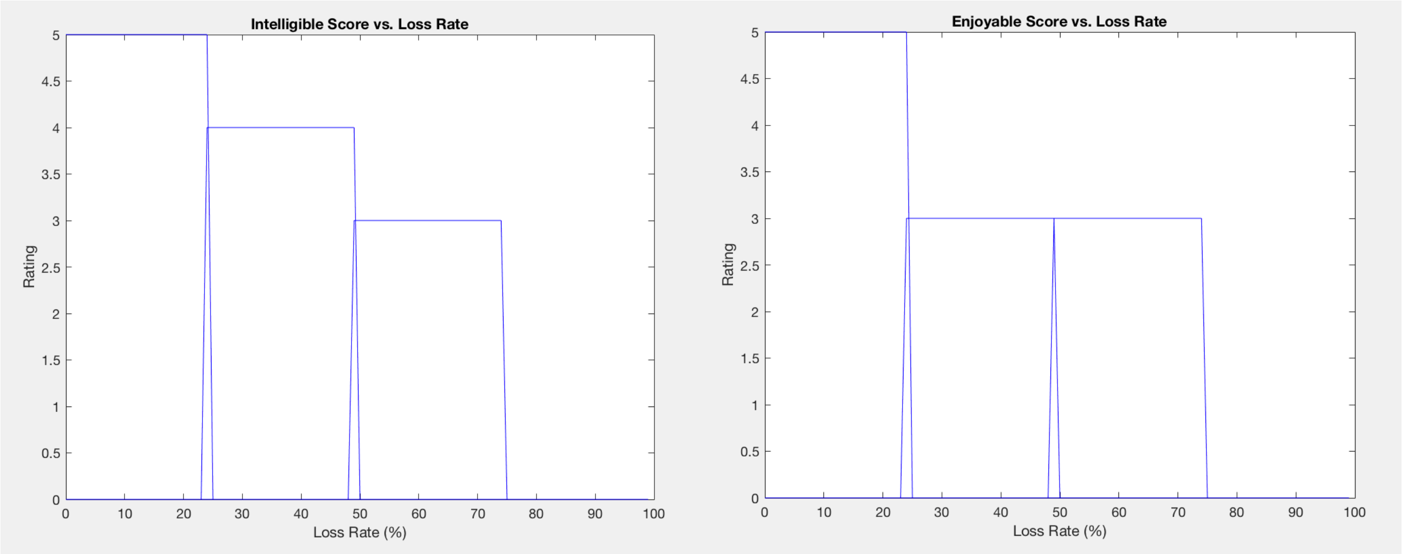
In this case, when all packets are lost, then we can see how the whole output is processed by the algorithm, i.e. no packets from the input are stored, but the alternate data resulting of the algorithm, with the most obvious effects when using alternative 1 (replace with 0s/playing silence).

Now, for the rest of the cases and for each packet size (from small to large), it is of interest to study how the quality of the received audio changes as the loss rate increases. All results are based on feedback after testing all test audio files that were mentioned in section Methodology of this report.

▪ Alternative 1: Playing silence (writing all 0s):

Packet Size (bytes)	*Loss Rate (%)*	*"Intelligible" Rating (1-5) for each Loss Rate*	*"Enjoyable" Rating (1-5) for each Loss Rate*
Small (4-500)	Small (0-25)	5	5
	Medium (25-50)	4	3
	Big (50-75)	3	3
Medium (1000-3000)	Small (0-25)	3	3
	Medium (25-50)	3	2
	Big (50-75)	2	2
Large (6000-10000)	Small (0-25)	1	1
	Medium (25-50)	1	1
	Big (50-75)	1	1

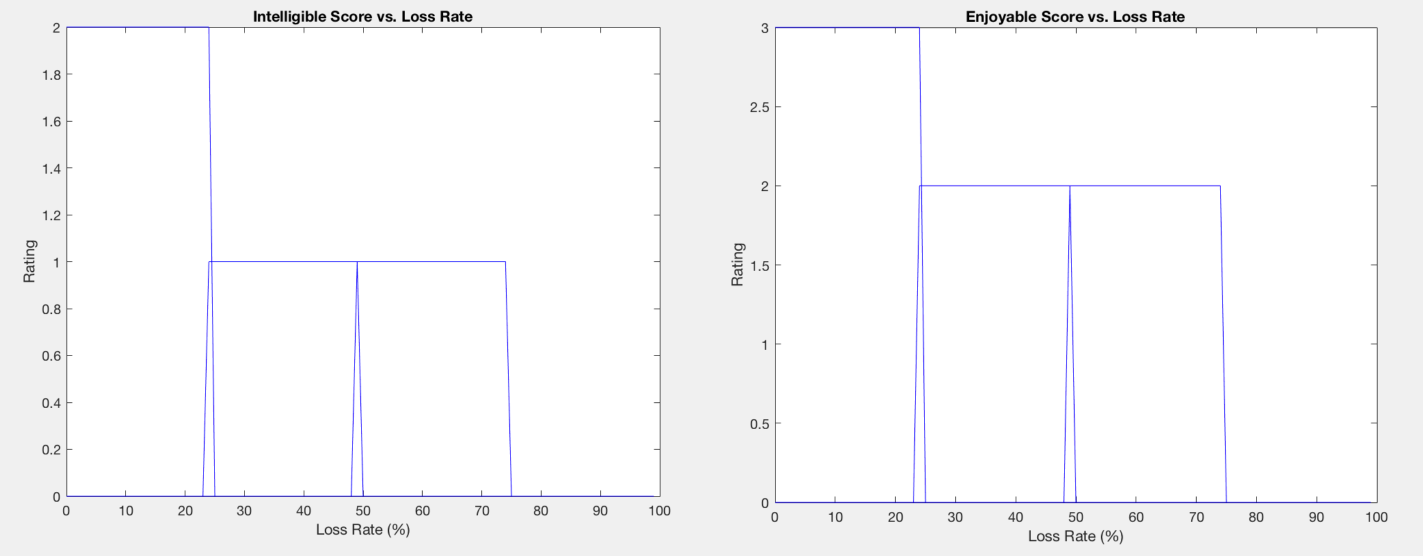
To have a better reference of this particular method trend, for this method I have plotted the results for small packets:



- Alternative 2: Replaying last sample:

Packet Size (bytes)	*Loss Rate (%)*	*“Intelligible” Rating (1-5) for each Loss Rate*	*“Enjoyable” Rating (1-5) for each Loss Rate*
Small (4-500)	Small (0-25)	3	3
	Medium (25-50)	3	3
	Big (50-75)	2	2
Medium (1000-3000)	Small (0-25)	2	3
	Medium (25-50)	1	2
	Big (50-75)	1	2
Large (6000-10000)	Small (0-25)	2	1
	Medium (25-50)	1	1
	Big (50-75)	1	1

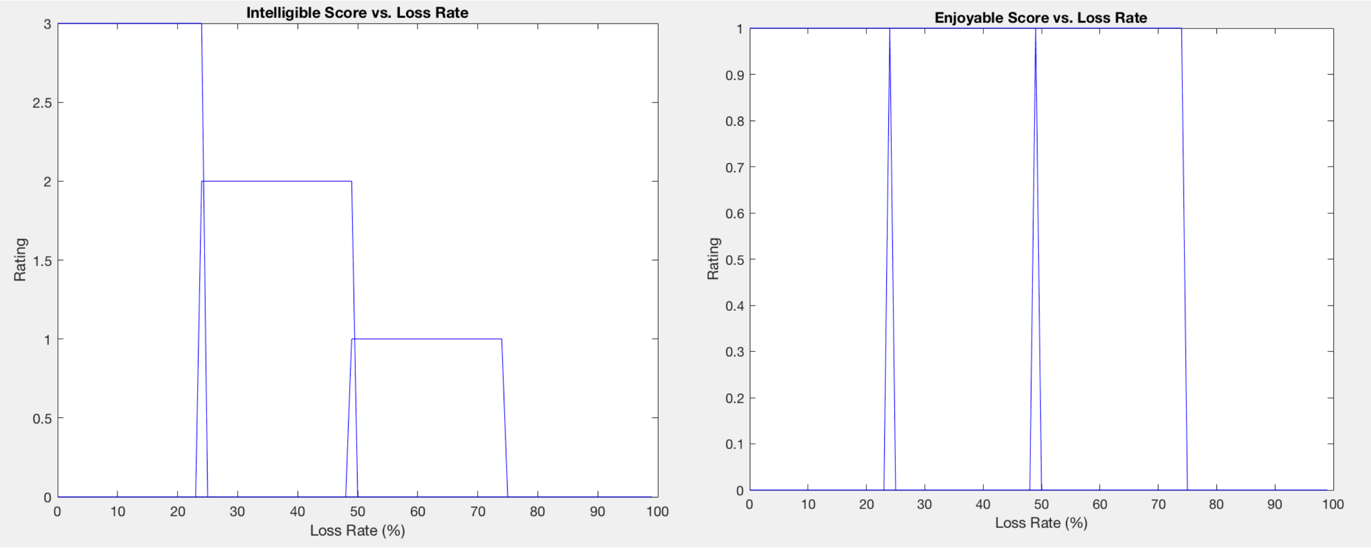
To study the trend, for this method I have plotted the results for medium packets:



- Alternative 3: Playing the entire last packet received:

Packet Size (bytes)	*Loss Rate (%)*	*"Intelligible" Rating (1-5) for each Loss Rate*	*"Enjoyable" Rating (1-5) for each Loss Rate*
Small (4-500)	Small (0-25)	5	5
	Medium (25-50)	5	4
	Big (50-75)	4	4
Medium (1000-3000)	Small (0-25)	3	4
	Medium (25-50)	3	2
	Big (50-75)	3	2
Large (6000-10000)	Small (0-25)	3	1
	Medium (25-50)	2	1
	Big (50-75)	1	1

To study the trend for this method and to better understand how large packets can affect the data, I have plotted the results for large packets:



We can see how the effects of using alternative 1 (using 0s) are mostly worse than the other 2 methods, as it is more “brute”. This is because putting zeros in the packets translates into silence and that is not very enjoyable nor intelligible in practice.

A similar effect can be appreciated using alternative 2, every time the last sample was a zero.

Simulating using alternative 3, as it plays/repeats the entire last packet received, when the packet size gets big enough, that translates into a sort of an “eco” effect. Another thing to point out is that too large packets can deteriorate the final received data even when using a decent data recovery method such as alternative 3.

Eventually, although depending on the message we can transmit (voice, music, etc), one or another thing is of interest, I would say method 3 is mostly the most effective, specially for voice (since even though it may have this repeating effect, it does avoid silence), and for less quality requiring messages methods 0 and 1 could be of better use. However, the computational cost of all of them is similar.

Conclusion

Modeling network loss on multimedia through this simulator has lead us to conclude that using method 3 (replacing with the whole last packet) is in general more effective, although it depends on the kind of information that wants to be transmitted.

They are all of a lot of use in multimedia streaming and that is why it is of interest to implement them in practice.

References

[1] youtube.com
[2] www.cs.ucsb.edu/~almeroth/classes/W18.176B/hw/samples/