Internet Radio Expriment

WG 5

Source Code

```c
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <stdio.h>

#include <sys/socket.h>
#include <netdb.h>
#include <fcntl.h>

struct addrinfo *target;
int sender;
int file;
unsigned bufSize;
int sendSize;
int delay;

static inline void error(const char *msg) {
    perror(msg);
    exit(-1);
}

int setTarget(char *host, char *port) {
    struct addrinfo hints;
    memset(&hints, 0, sizeof(struct addrinfo));
    hints.ai_socktype = SOCK_DGRAM;
    hints.ai_flags = AI_ADDRCONFIG;
    if (getaddrinfo(host, port, &hints, &target) != 0) error("fail to resolve remote address");
    if ((sender = socket(AF_INET, SOCK_DGRAM, 0)) == -1) error("cannot create socket");
}

int main(int argc, char **argv) {
    if (argc != 7) return printf("invalid arguments\n");
    setTarget(argv[1], argv[2]);
    file = open(argv[3], O_RDONLY);
    bufSize = atoi(argv[4]);
    sendSize = atoi(argv[5]);
    delay = atoi(argv[6]);

    if (sendSize > bufSize) error("cannot send more than read");
    char *buf = malloc(bufSize);

    int readSize = 0;
    while ((readSize = read(file, buf, bufSize)) > 0) {
        int sent = 0;
        while (readSize > sent) {
            while (sendto(sender, buf + sent, (readSize - sent) > sendSize ? sendSize : (readSize - sent), 0,
                          target->ai_addr, target->ai_addrlen) == -1);
            sent += sendSize;
            usleep(delay);
        }
    }
    return 0;
}
```

P1 is buffer size in bytes, P2 is size of transmission packet in bytes, and P3 is delay between to transmission in microsecond.

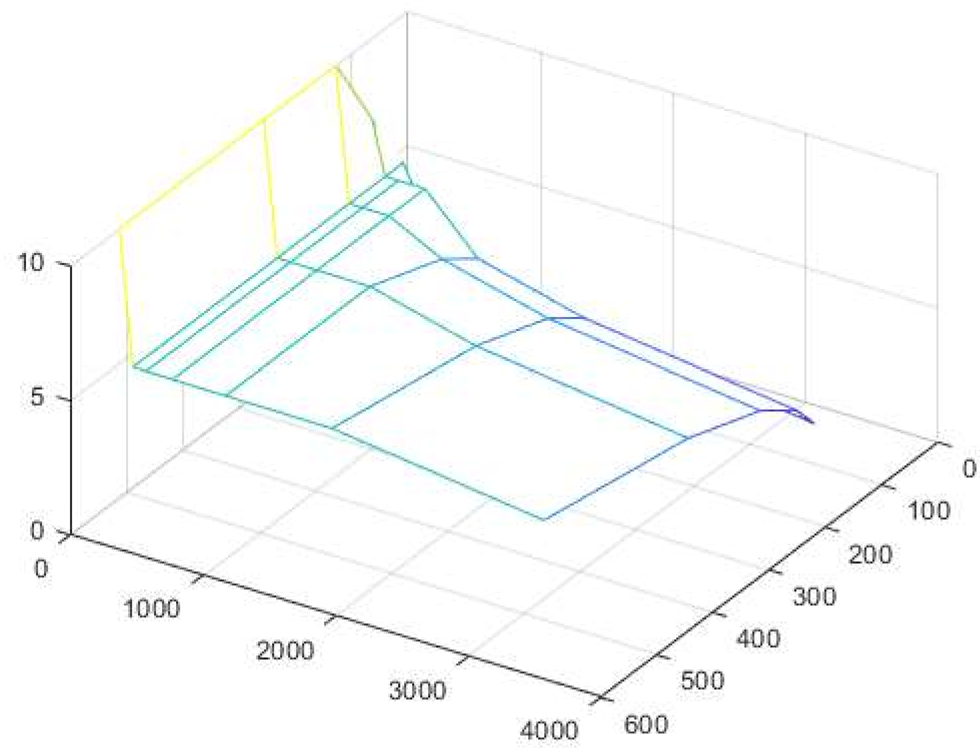
Graphs

In all below experiments, we set the buffer size of vlc to 1000 ms.

mp3 format

We test the quality of transmission with small buffer sizes and a large buffer size.

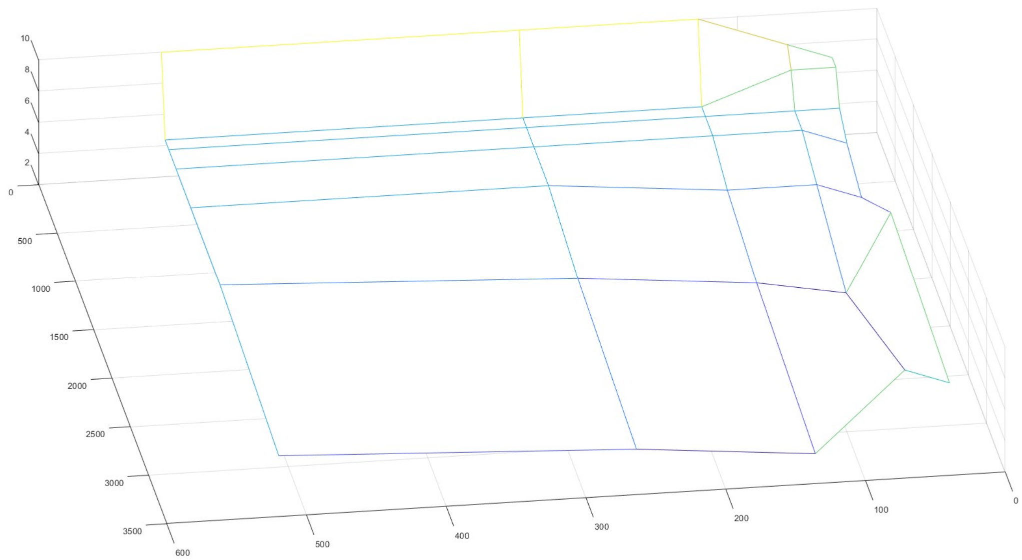
Delay / Buffer	0	100	200	400	800	1600	3200
32	5	5	4	3	2	1	0
64	7	5	5	5	3	2	1
128	10	5	5	5	4	3	2
256	10	5	5	5	5	4	3
512	10	5	5	5	5	5	4



When packets are arriving faster than timeline, VLC seems to make a fast forward, causing a sharp decrease of quality from 10 to 5. That seems to be also the reason why the quality is not 10 for buffer size 32 and 64 when delay is slightly larger than 0.

Below is the test result for buffer size=4096.

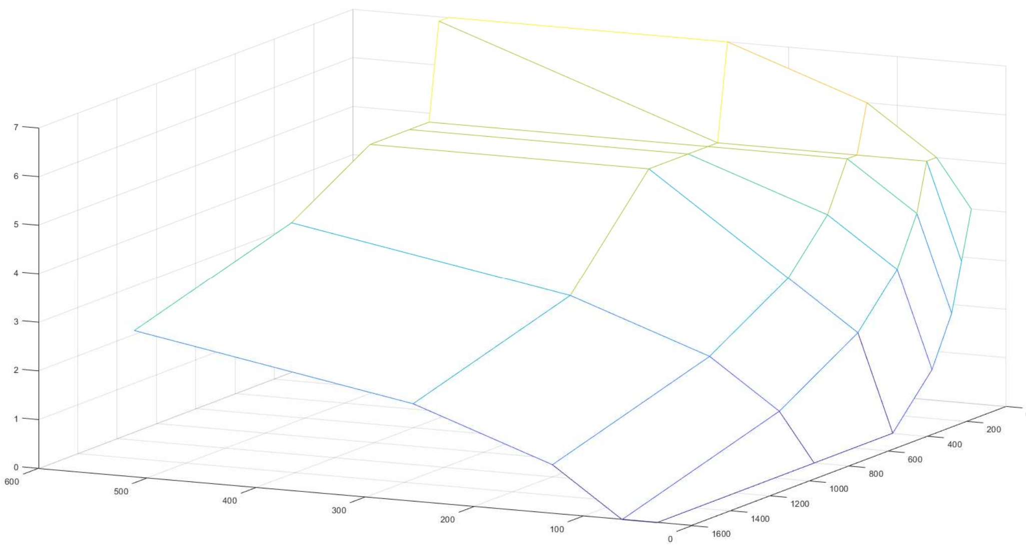
Delay Packet	0	100	200	400	800	1600	3200
32	7	7	5	4	3	7	6
64	8	7	5	5	4	2	7
128	10	5	5	5	4	3	2
256	10	5	5	5	5	4	3
512	10	5	5	5	5	5	4



lossless formats

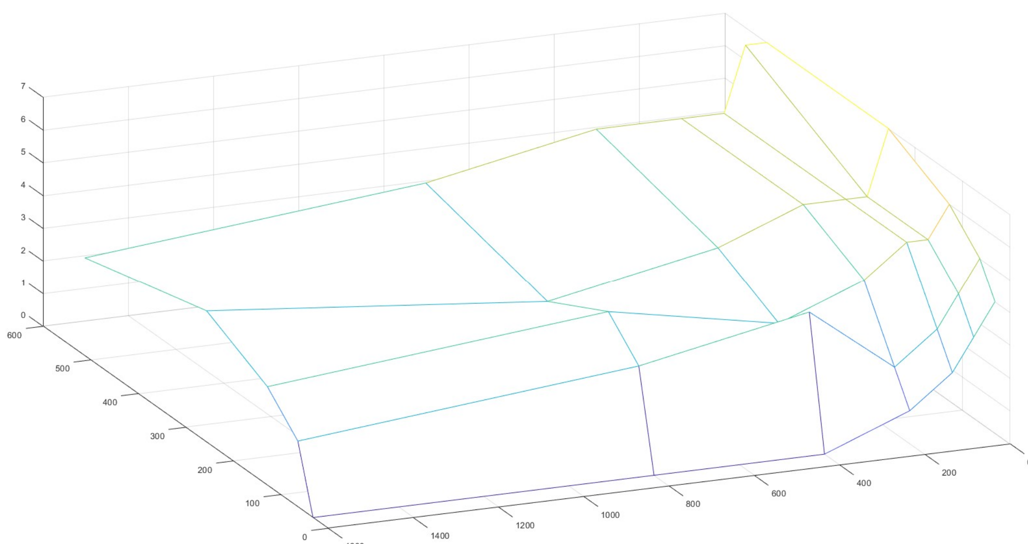
We choose WAV for experiments about lossless formats. The WAV file is converted from mp3 file using ffmpeg.

Delay Buffer	0	50	100	200	400	800	1600
32	4	3	2	1	0	0	0
64	5	5	4	3	2	1	0
128	6	5	5	4	3	2	1
256	7	5	5	5	5	3	2
512	7	7	5	5	5	4	3



Below is the test result for buffer size=4096.

Delay / Packet	0	50	100	200	400	800	1600
32	4	3	2	1	0	0	0
64	5	4	3	2	4	3	2
128	6	5	5	4	3	4	3
256	7	5	5	5	4	3	4
512	7	7	5	5	5	4	3



Conclusion

Normally a larger buffer size will reduce latency caused by file operation, which improve the quality slightly.

A lager packet size will usually improve the quality, but may result in a fast forward.

Usually the longer the delay is, the worse the quality becomes. But an appropriate delay will dramatically reduce fast forward.