

# Video Streaming Application Design Report

## Karl Boatright and Wagner Colodette

### Problem Definition:

#### Problem Scope

We will be taking on the challenge of designing and developing an application that can execute streaming of video between a sender and a receiver.

#### Design Requirements

We need to create an application that can be distributed to multiple users (at least two) and can be executed easily. This application needs to be able to stream a video from one user (sender) to another user (receiver) byte by byte. Since we are to be streaming video files, the files may be potentially very large and therefore we will need to maintain a constant connection between sender and receiver. Furthermore, we need to consider that videos use different encodings, so our streaming application will need to be able to stream a range of encodings. In the case of encodings, we may need to implement a method for transcoding such as the use of FFmpeg, or some other solution to convert video files.

### Design Description

#### Overview

This application will be designed using Python as the main programming language. With regard to the streaming aspect, we will require the use of other libraries and APIs to develop this solution. The previously mentioned FFmpeg solution may also serve the purpose that we need for video streaming. According to their documentation, FFmpeg can stream via UDP/TCP directly to a specified single destination. This will provide us with the ability to transcode and send/receive videos using the same FFmpeg library. Additionally, there exists Python bindings for FFmpeg in the form of ffmpeg-python that can be attained via pip using: ***pip install ffmpeg-python***. Further documentation regarding the Python bindings for FFmpeg can be found at <https://github.com/kkroening/ffmpeg-python>.

Another solution for video streaming capabilities that we will explore is OpenCV. Like FFmpeg, OpenCV is able to stream videos, however, for any solution that we find fits best within our stated design requirements, we will need to use the Python socket library to open sockets on a sender and receiver machine, as well as implement a solution that will send the video in chunks using either UDP/TCP from the sender machine, then those chunks will be received by the receiver machine and properly handled so that the receiver will then be able to play that video.