Low Latency Camera Feed Development

Members: Brayden Casaren, Sebastian Clarke, Ayush Iyer, Rohit Kartickeyan
Advisors: Ivan Seskar and Jennifer Shane

This work was supported in part by the NSF REU program and the donation from nVERSES CAPITAL

Objective

To find the method(s) of reducing latency to a minimum in a unicast camera to computer connection over a network.

Data Analysis

These histograms show the number of occurrences in the 5-sec camera footage for each type of colored pixel value with no noise.

- Red/Green
  - The number of occurrences is mostly 0 because most of the camera footage is dark, though there are some high red/green pixel values due to some white light being present when the LED flashes on.

- Blue
  - The occurrences of blue pixel values is significantly higher because the LED emits blue light.

Tools

WireShark, TCPDUMP, xxd (HEXDUMP), Python, OpenCV, Matplotlib, VLC Media Player, FFMPEG

Project Design

- All devices are connected through a switch
- Our camera and LED light is sealed in a box with minimal light.
- As camera streams footage to computer, we can save the incoming video packets through a software called TCPDUMP.
- When the LED light turns on, the data inside the transmitting video packets will change, indicating that light is now being captured.

Calculating & Reducing Latency

By comparing the frame data to the packet data, we can find what exact packet correlates to the frame. By using the timestamp of the packet, we can then be able to find the latency.

Using this, we found our latency to be **45 milliseconds**.

Camera settings we theorised to reduce latency were:
- Changing codec (h264 vs MJPEG)
- Changing data compression rate (default 30)
- Changing Sharpness/Contrast/White balance/etc

Conclusion & Future Work

- The codec of our camera would likely yield the most difference in latency
- Study how different codecs would influence a camera’s latency
- Take the most optimal low latency conditions and apply them to developing a low latency camera.