

# Low Latency Camera Feed Development

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

# Researchers Introduction



Brayden Casaren



Sebastian Clarke



Rohit Karthickeyan



Ayush Iyer



## Our Mission

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

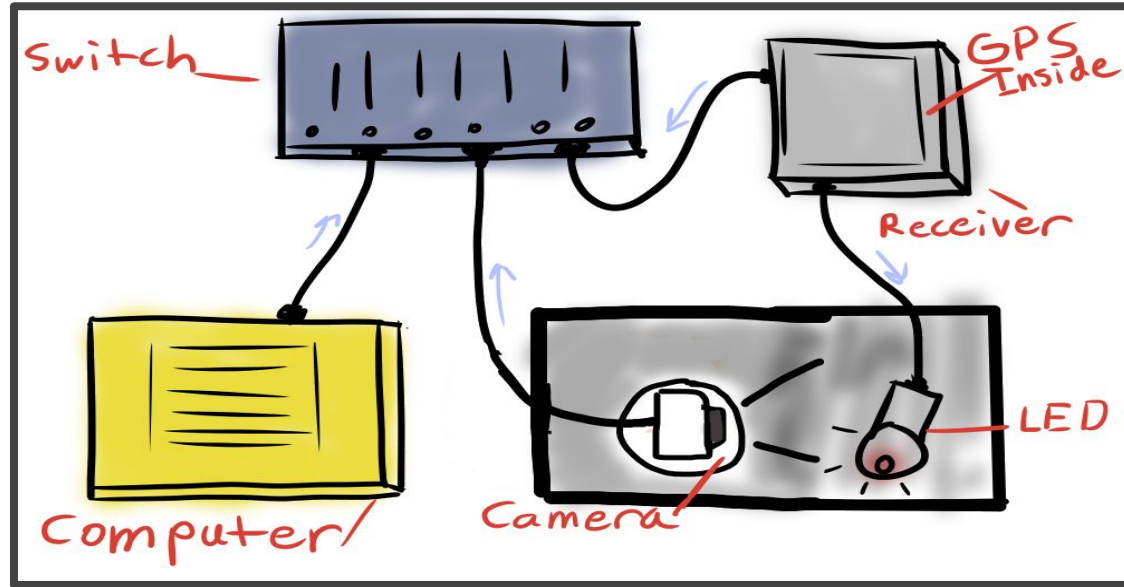


# The Importance of Low Latency Cameras

The modern day world requires low latency cameras for many applications including:

- Self driving cars
- Live Sporting Events and television
- Interviews and News
- Security and Surveillance
- Remote Work Operations(Like Zoom)

# Project Design



Testing Setup Diagram

# Packets

packs.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	18:06:08.343265	10.41.250.42	10.41.1.11	TCP	74	80 → 38656 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SAC...
2	18:06:08.343681	10.41.250.42	10.41.1.11	TCP	66	80 → 38656 [ACK] Seq=1 Ack=148 Win=6864 Len=0 TSval=303297038...
3	18:06:08.349015	10.41.250.42	10.41.1.11	HTTP	634	HTTP/1.1 401 Unauthorized (text/html)
4	18:06:08.349438	10.41.250.42	10.41.1.11	TCP	74	80 → 38668 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SAC...
5	18:06:08.349810	10.41.250.42	10.41.1.11	TCP	66	80 → 38668 [ACK] Seq=1 Ack=392 Win=6864 Len=0 TSval=303297039...
6	18:06:08.349873	10.41.250.42	10.41.1.11	TCP	66	80 → 38656 [FIN, ACK] Seq=569 Ack=149 Win=6864 Len=0 TSval=30...

▶ Frame 1518: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)  
▶ Ethernet II, Src: AxisComm\_c6:a4:97 (00:40:8c:c6:a4:97), Dst: Mellanox\_3b:81:cf (98:03:9b:3b:81:cf)  
▶ Internet Protocol Version 4, Src: 10.41.250.42, Dst: 10.41.1.11  
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 38668, Seq: 1978599, Ack: 392, Len: 1448  
Source Port: 80  
Destination Port: 38668  
[Stream index: 1]

```
0000 98 03 9b 3b 81 cf 00 40 8c c6 a4 97 08 00 45 00  ...;...@ .....E-
0010 05 dc 89 01 40 00 40 06 9c 93 0a 29 fa 2a 0a 29  ...@.@...)*.
0020 01 0b 00 50 97 0c 89 e0 a1 fc 98 2a 47 aa 80 10  ...P.....*G...
0030 06 b4 e6 8b 00 00 01 01 08 0a 12 13 f3 7d d0 32  ...}.....}..2
0040 a3 7a 2d 2d 6d 79 62 6f 75 6e 64 61 72 79 0d 0a  -z--mybo undary..
0050 43 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 69 6d  Content-Type: im
0060 61 67 65 2f 6a 70 65 67 0d 0a 43 6f 6e 74 65 6e  age/jpeg ..Conten
0070 74 2d 4c 65 6e 67 74 68 3a 20 31 35 36 37 36 0d  t-Length : 15676-
0080 0a 0d 0a ff d8 ff e0 00 10 4a 46 49 46 00 01 02  .....JFIF...
0090 00 00 01 00 01 00 00 ff fe 00 0f 0a 01 64 ca e1  .....d...
00a0 5b 26 64 ca e1 5b 26 01 ff fe 00 0f 0a 00 01 75  [&d..[&.....u
00b0 05 33 01 00 40 8c c6 a4 97 ff db 00 43 00 0a 07  -3..@... ..C...
```

This shows the raw value of the sequence number (tcp.seq\_raw), 4 bytes

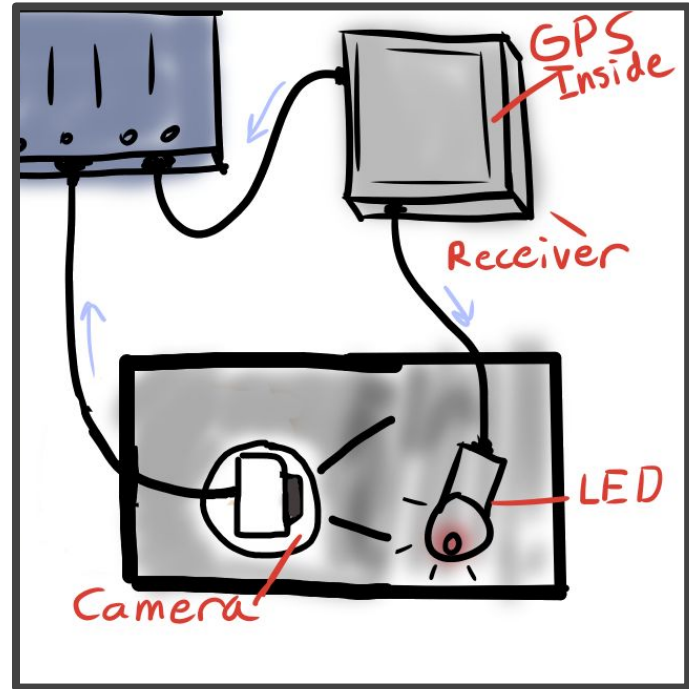
Packets: 1858 · Displayed: 1858 (100.0%) Profile: Default

# GPS, PPS, and PTP

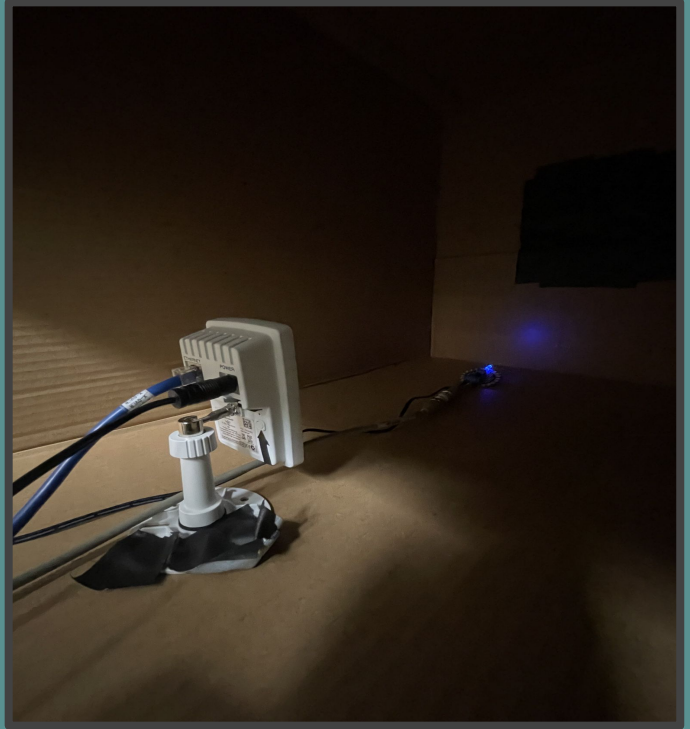
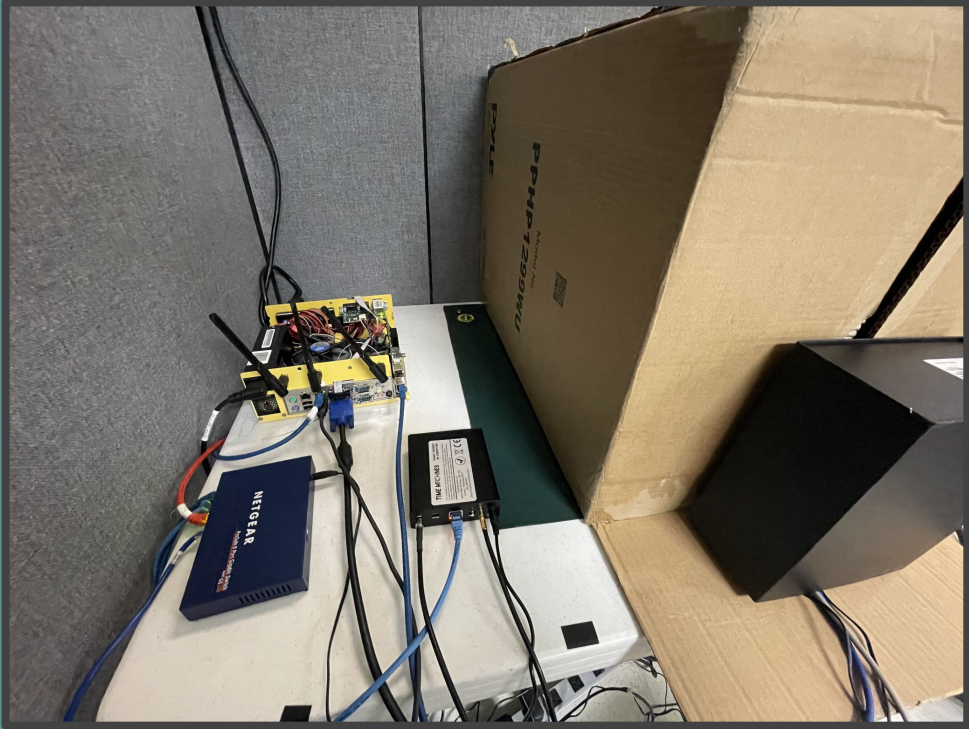
GPS: Global Positioning System

PPS: Pulse per second

PTP: Precision Time Protocol







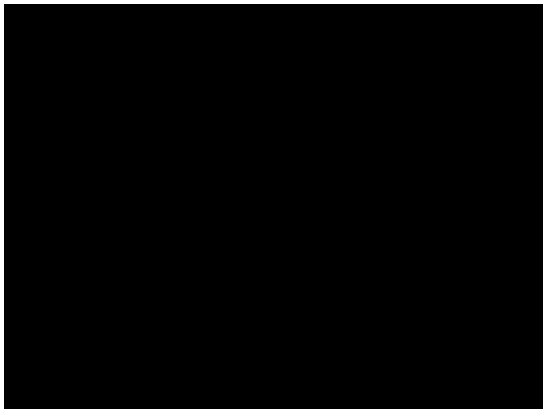




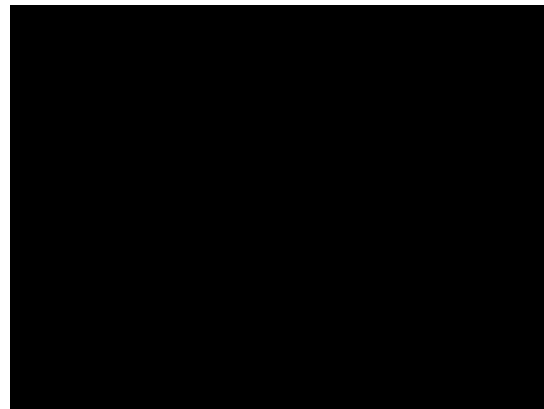
# Creating Artificial Video

Created videos of varying noises (0%-50%) which were streamed from our secondary node to our primary over VLC

- This setup will allow us to simulate how noise affects our actual camera.



No Noise

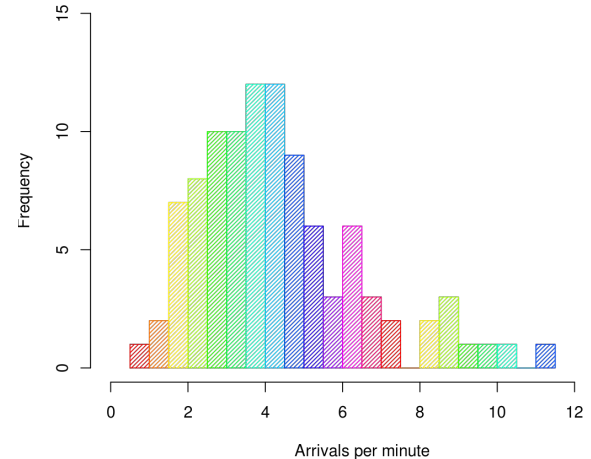


30% Noise



# Visualizing our data

- ❖ Using Matplotlib to create the histograms of our data (byte occurrences)
- ❖ Useful for visualizing data particularly for comparison and pattern recognition (more than million pieces of data)
- ❖ Want to gain information of how noise affects packets and eventually use for identify when changes occur in our video



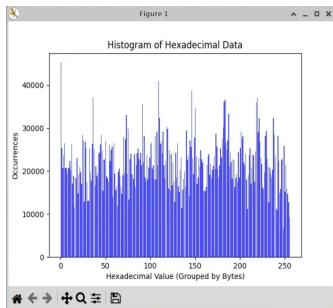
matplotlib



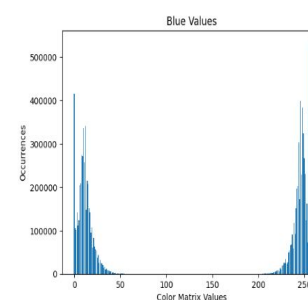
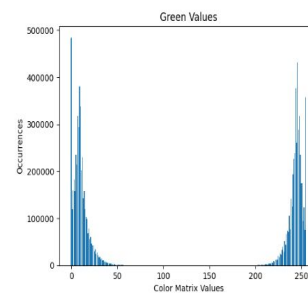
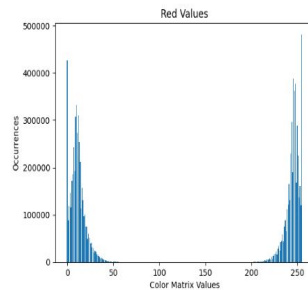
# Histograms

We changed the ways our histogram worked in a couple a ways but ended up decided on this:

- Measures RGB pixel values instead of bytes
- Creates three separate graphs for each color (RGB) of how often certain values occurred



Old 10% Graph



New 10% Graphs

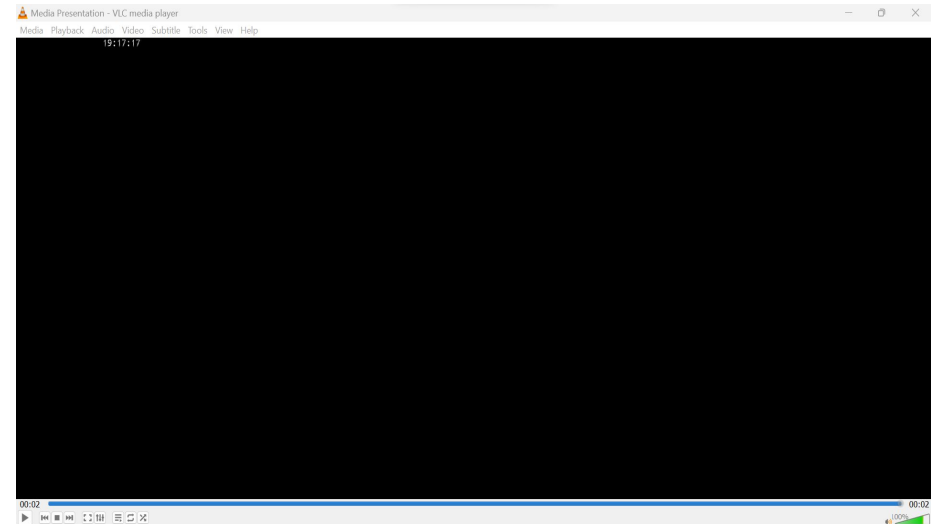
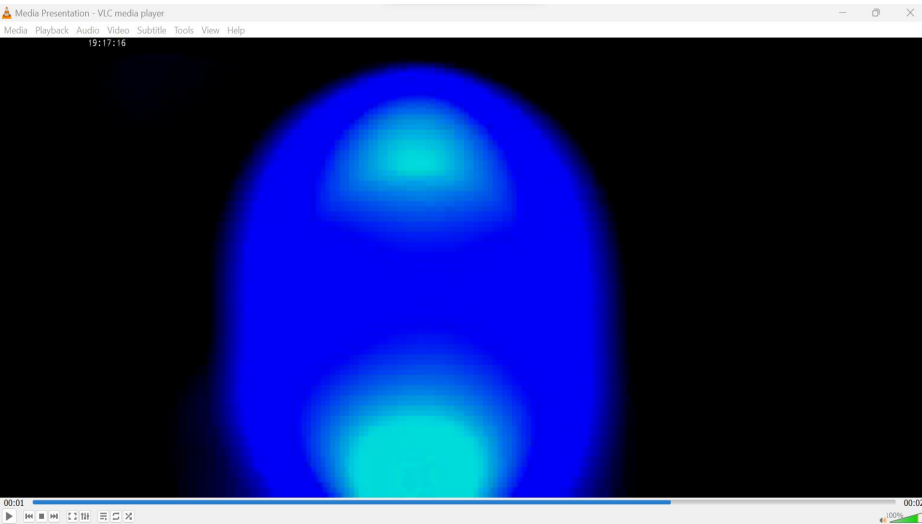


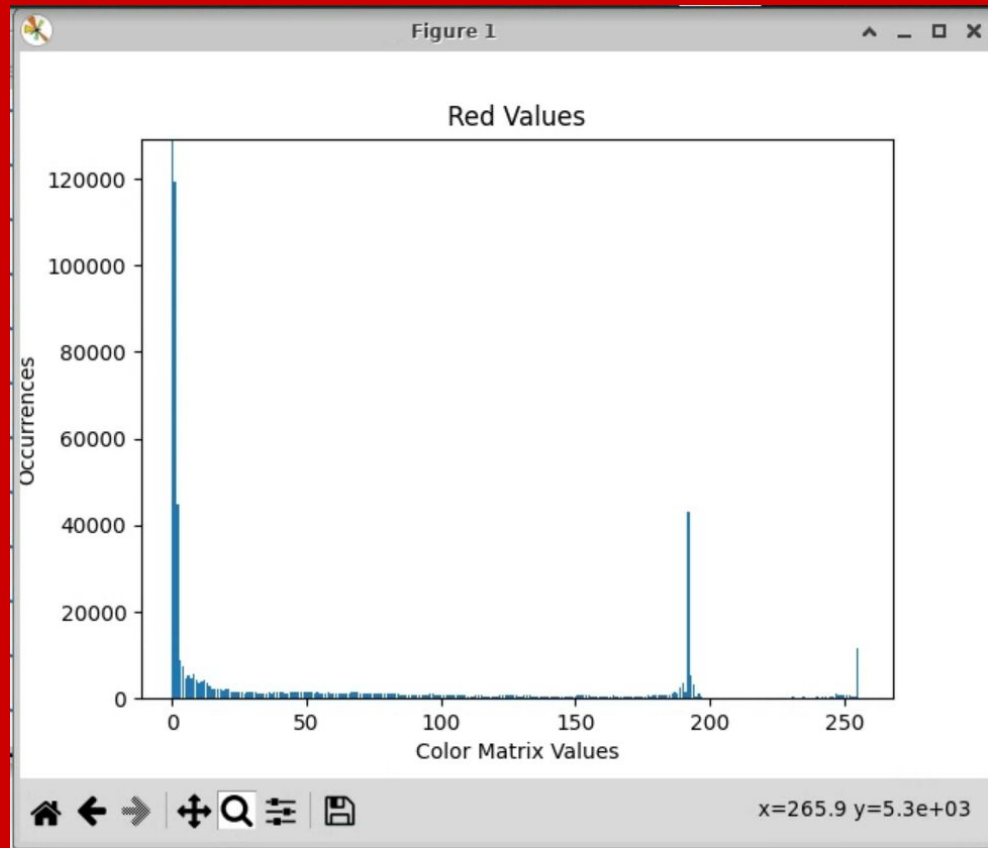
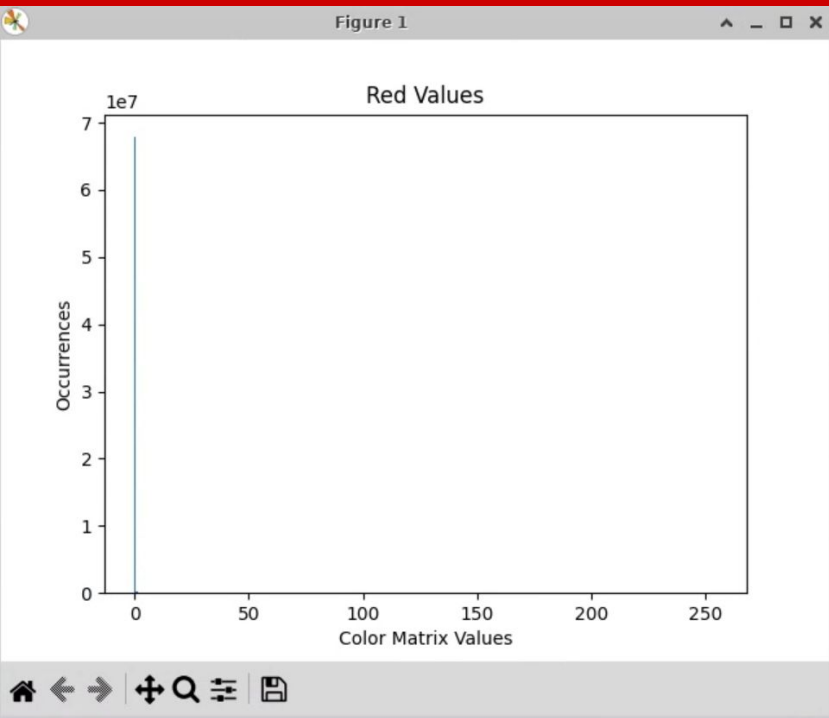
# More changes we made

Through camera settings made the noise observed very little

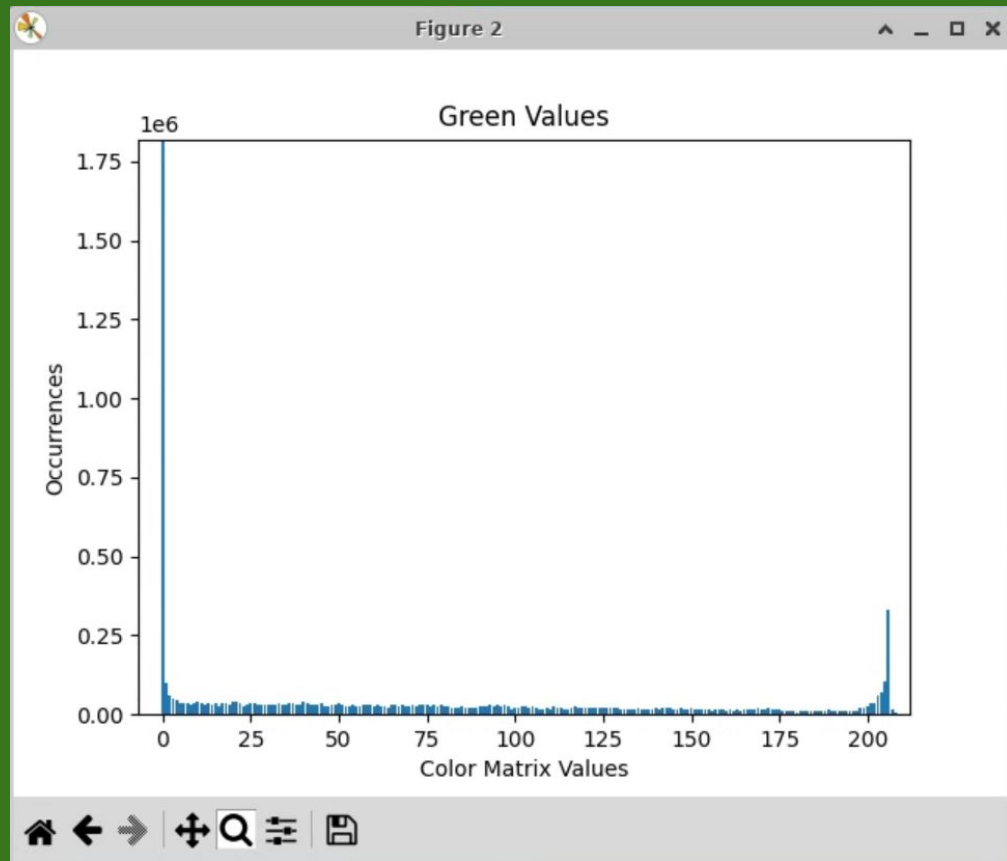
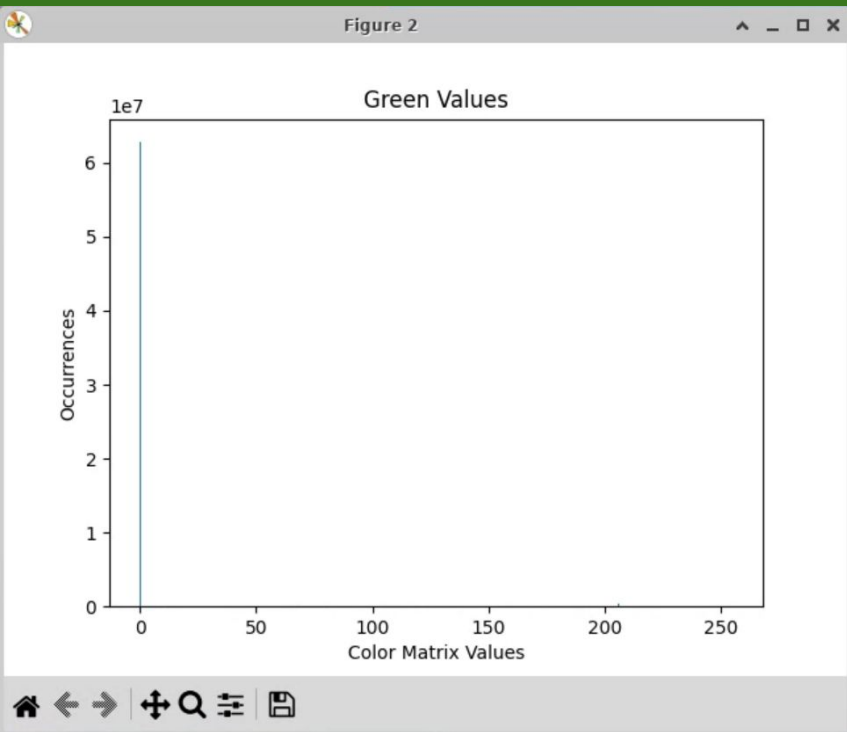
Changed LED setup to take up more of the camera's view

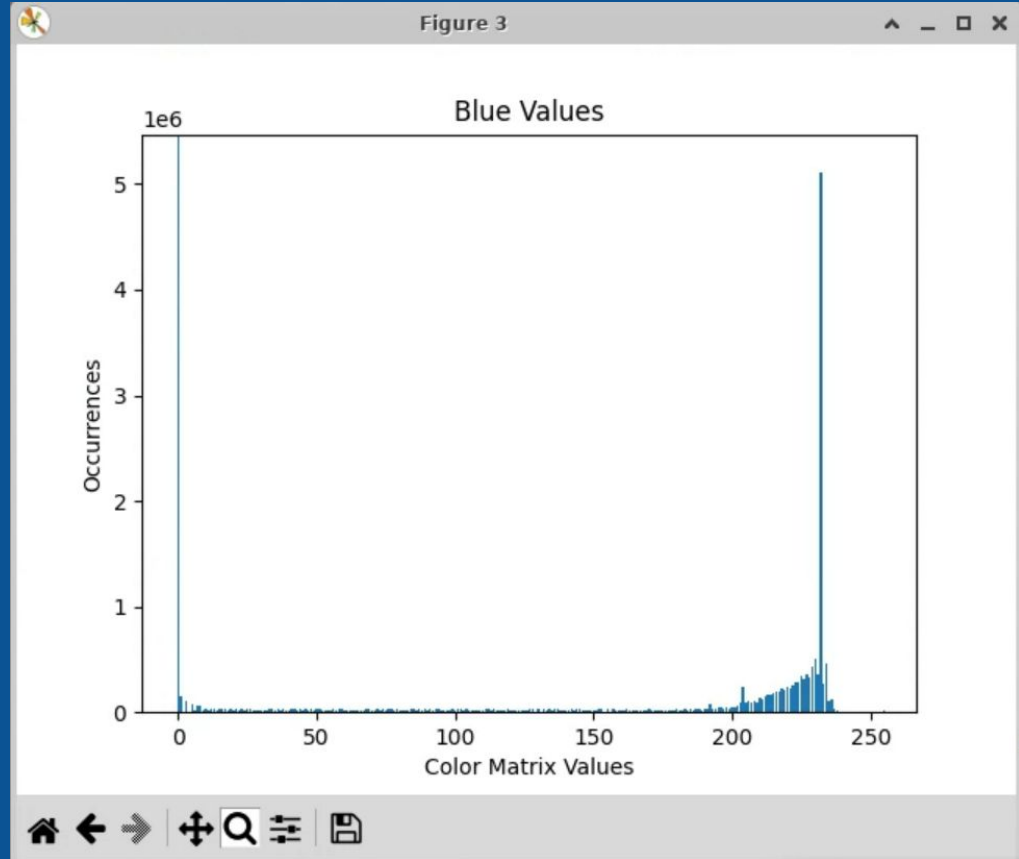
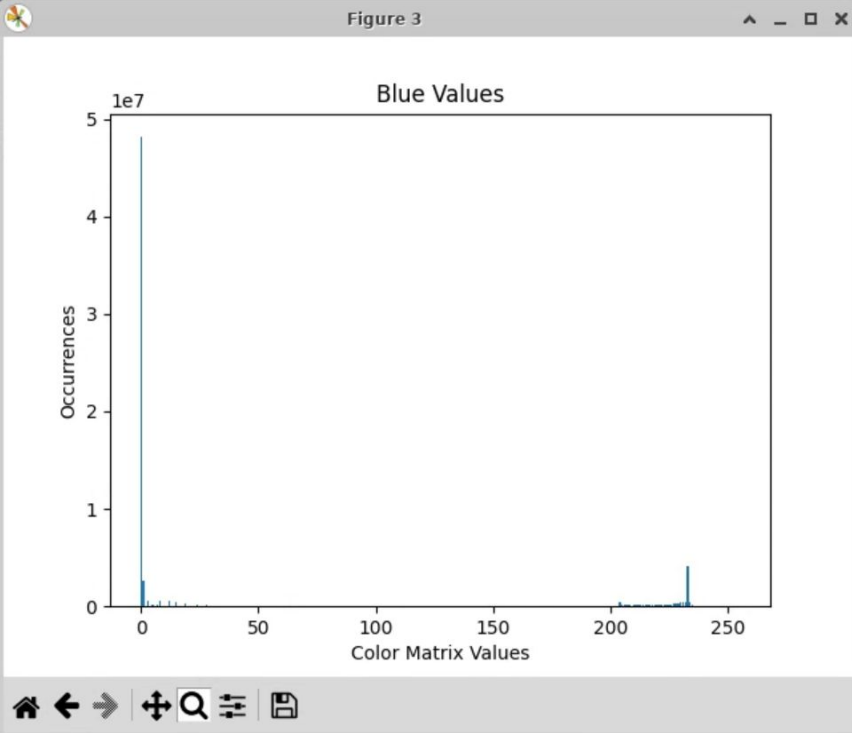
Used FFMPEG to get video into one MJPEG or and its constituent JPEG files





x=265.9 y=5.3e+03







# Calculating Latency

The screenshot displays the Wireshark interface for a capture named 'packs.pcap'. The main pane shows a list of network packets. The selected packet is a TCP ACK with sequence number 2313200124.

No.	Time	Source	Destination	Protocol	Length	Info
220	18:06:09.012572	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=280469 Ack=392 Win=6864 Len=1448 TSval=3...
221	18:06:09.012754	10.41.250.42	10.41.1.11	TCP	1314	80 → 38668 [ACK] Seq=281917 Ack=392 Win=6864 Len=1248 TSval=3...
222	18:06:09.013577	10.41.250.42	10.41.1.11	TCP	68	80 → 38668 [PSH, ACK] Seq=283165 Ack=392 Win=6864 Len=2 TSval=3...
223	18:06:09.044308	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=283167 Ack=392 Win=6864 Len=1448 TSval=3...
224	18:06:09.044389	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=284615 Ack=392 Win=6864 Len=1448 TSval=3...
225	18:06:09.044510	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=286063 Ack=392 Win=6864 Len=1448 TSval=3...
226	18:06:09.044641	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=287511 Ack=392 Win=6864 Len=1448 TSval=3...
227	18:06:09.044758	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=288959 Ack=392 Win=6864 Len=1448 TSval=3...
228	18:06:09.045008	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=290407 Ack=392 Win=6864 Len=1448 TSval=3...
229	18:06:09.045118	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=291855 Ack=392 Win=6864 Len=1448 TSval=3...
230	18:06:09.045246	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=293303 Ack=392 Win=6864 Len=1448 TSval=3...
231	18:06:09.045364	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=294751 Ack=392 Win=6864 Len=1448 TSval=3...
232	18:06:09.045487	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=296199 Ack=392 Win=6864 Len=1448 TSval=3...
233	18:06:09.045593	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=297647 Ack=392 Win=6864 Len=1448 TSval=3...
234	18:06:09.045777	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=299095 Ack=392 Win=6864 Len=1448 TSval=3...
235	18:06:09.045902	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=300543 Ack=392 Win=6864 Len=1448 TSval=3...
236	18:06:09.046026	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=301991 Ack=392 Win=6864 Len=1448 TSval=3...
237	18:06:09.046026	10.41.250.42	10.41.1.11	TCP	122	80 → 38668 [ACK] Seq=303439 Ack=392 Win=6864 Len=56 TSval=303...
238	18:06:09.046977	10.41.250.42	10.41.1.11	TCP	68	80 → 38668 [PSH, ACK] Seq=303495 Ack=392 Win=6864 Len=2 TSval=3...
239	18:06:09.077581	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=303497 Ack=392 Win=6864 Len=1448 TSval=3...
240	18:06:09.077621	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=304945 Ack=392 Win=6864 Len=1448 TSval=3...
241	18:06:09.077791	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=306393 Ack=392 Win=6864 Len=1448 TSval=3...

Sequence number (raw): 2313200124

Packets: 1858 · Displayed: 1858 (100.0%) Profile: Default

No.	Time	Source	Destination	Protocol	Length	Info
196	18:06:08.945515	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=249009 Ack=392 Win=6864 Len=1448 TSval=3...
197	18:06:08.945602	10.41.250.42	10.41.1.11	TCP	1314	80 → 38668 [ACK] Seq=250457 Ack=392 Win=6864 Len=1248 TSval=3...
198	18:06:08.946447	10.41.250.42	10.41.1.11	TCP	68	80 → 38668 [PSH, ACK] Seq=251705 Ack=392 Win=6864 Len=2 TSval=...
199	18:06:08.977719	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=251707 Ack=392 Win=6864 Len=1448 TSval=3...
200	18:06:08.977810	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=253155 Ack=392 Win=6864 Len=1448 TSval=3...
201	18:06:08.977934	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=254603 Ack=392 Win=6864 Len=1448 TSval=3...

Source Port: 80  
Destination Port: 38668  
[Stream index: 1]  
[TCP Segment Len: 1448]

Sequence number: 251707 (relative sequence number)

0020	01 0b 00 50 97 0c 89 c6	48 50 98 2a 47 aa 80 10	...P... HP.*G...
0030	06 b4 7e 17 00 00 01 01	08 0a 12 13 f2 4d d0 32	...M.2
0040	97 a2 2d 2d 6d 79 62 6f	75 6e 64 61 72 79 0d 0a	...-mybo undary..
0050	43 6f 6e 74 65 6e 74 2d	54 79 70 65 3a 20 69 6d	Content-Type: im
0060	61 67 65 2f 6a 70 65 67	0d 0a 43 6f 6e 74 65 6e	age/jpeg ..Conten
0070	74 2d 4c 65 6e 67 74 68	3a 20 31 35 36 36 33 0d	t-Length : 15663.
0080	0a 0d 0a ff d8 ff e0 00	10 4a 46 49 46 00 01 02	.....JFIF...
0090	00 00 01 00 01 00 00 ff	fe 00 0f 0a 01 64 ca e1	.....d...
00a0	58 23 64 ca e1 58 23 01	ff fe 00 0f 0a 00 01 75	X#d..X#..u
00b0	05 33 01 00 40 8c c6 a4	97 ff db 00 43 00 0a 07	.3..@...C...
00c0	07 08 07 06 0a 08 08 08	0b 0a 0a 0b 0e 18 10 0e	.....
00d0	0d 0d 0e 1d 15 16 11 18	23 1f 25 24 22 1f 22 21	.....#.%\$".!"
00e0	26 2b 37 2f 26 29 34 29	21 22 30 41 31 34 39 3b	&+7/&)4) !"0A149;
00f0	3e 3e 3e 25 2e 44 49 43	3c 48 37 3d 3e 3b ff db	>>>%.DIC <H7=>;..
0100	00 43 01 0a 0b 0b 0e 0d	0e 1c 10 10 1c 3b 28 22	.C.....;("

This shows the raw value of the sequence number (tcp.seq\_raw), 4 bytes

Packets: 1858 · Displayed: 1858 (100.0%)

Profile: Default



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help



Apply a display filter ... &lt;Ctrl-/&gt;

No.	Time	Source	Destination	Protocol	Length	Info
223	18:06:09.044308	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=283167 Ack=392
224	18:06:09.044389	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=284615 Ack=392
225	18:06:09.044510	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=286063 Ack=392
226	18:06:09.044641	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=287511 Ack=392
227	18:06:09.044758	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=288959 Ack=392
228	18:06:09.045008	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=290407 Ack=392
229	18:06:09.045118	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=291855 Ack=392
230	18:06:09.045246	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=293303 Ack=392
231	18:06:09.045364	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=294751 Ack=392
232	18:06:09.045487	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=296199 Ack=392
233	18:06:09.045593	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=297647 Ack=392
234	18:06:09.045777	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=299095 Ack=392
235	18:06:09.045902	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=300543 Ack=392
236	18:06:09.046026	10.41.250.42	10.41.1.11	TCP	1514	80 → 38668 [ACK] Seq=301991 Ack=392
237	18:06:09.046026	10.41.250.42	10.41.1.11	TCP	122	80 → 38668 [ACK] Seq=303439 Ack=392

▶ Frame 237: 122 bytes on wire (976 bits), 122 bytes captured (976 bits)

```

0000  98 03 9b 3b 81 cf 00 40 8c c6 a4 97 08 00 45 00  ...;...@ .....E.
0010  00 6c 84 00 40 00 40 06 a7 04 0a 29 fa 2a 0a 29  .l..@.@. ....)*.
0020  01 0b 00 50 97 0c 89 c7 12 64 98 2a 47 aa 80 10  ...P....d.*G...
0030  06 b4 44 30 00 00 01 01 08 0a 12 13 f2 54 d0 32  ..D0.....T.2
0040  98 04 8a 00 28 a2 8a 00 28 a2 8a 00 28 a2 8a 00  ..(..(.(.(.
0050  28 a2 8a 00 28 a2 8a 00 28 a2 8a 00 28 a2 8a 00  (.(..(.(.(.
0060  28 a2 8a 00 28 a2 8a 00 28 a2 8a 00 28 a2 8a 00  (.(..(.(.(.
0070  28 a2 8a 00 28 a2 8a 00 ff d9                    (.(..(.(.(.

```

```
hexfile= open("output_0018.txt", "r")
packetsfile=open("P223.txt", "r")
hex=""
img=""
packets=[]
file=packetsfile.read()
file=file.split()
for i in file:
    hex=hex+i.upper()
hex=hex[hex.index("FFD8"):]
image=hexfile.read()
image=image.split()
for i in image:
    img=img+i.upper()
if(img[:len(hex)]==hex):
    print("Match")
else:
    print("No Match")
```

```
winlab@node1-11:~/FFMPEGplusTCPDUMP/testjpg$ python3 compare.py
Match
```



# Reducing Latency:

Viable methods:

Changing Codec, Camera can run MJPEG and H264

Changing Framerate, Camera can operate between 0-30 fps

Changing Compression amount, Camera defaults to 30 on scale of 0-100

# Switching Codec (H264)

- 3 Types of frames: I, P, and B
- Frame referencing and motion vectors with P and B frames to reduce size
- Calculated latency to be around 34 ms for just I frames, 60 ms for GOV of 3

The screenshot shows the Wireshark interface with a packet list table and a packet details pane. The packet list table displays the following data:

Time	Source	Destination	Protocol	Length	Info
25	19:41:56.787471	10.41.250.42	H264	63	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48212, Time=480641078, Mark non-IDR-Slice
26	19:41:56.820973	10.41.250.42	H264	63	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48213, Time=480644076, Mark non-IDR-Slice
27	19:41:56.850551	10.41.250.42	H264	1442	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48214, Time=480647074 FU-A Start:IDR-Slice
28	19:41:56.850689	10.41.250.42	H264	1442	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48215, Time=480647074 FU-A
29	19:41:56.850814	10.41.250.42	H264	1306	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48216, Time=480647074, Mark FU-A End
30	19:41:56.887813	10.41.250.42	H264	127	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48217, Time=480650072, Mark non-IDR-Slice
31	19:41:56.920855	10.41.250.42	H264	66	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48218, Time=480653069, Mark non-IDR-Slice
32	19:41:56.954000	10.41.250.42	H264	66	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48219, Time=480656067, Mark non-IDR-Slice
33	19:41:56.987631	10.41.250.42	H264	63	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48220, Time=480659065, Mark non-IDR-Slice
34	19:41:57.020569	10.41.250.42	H264	63	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48221, Time=480662061, Mark non-IDR-Slice
35	19:41:57.054389	10.41.250.42	H264	1442	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48222, Time=480665061 FU-A Start:non-IDR-Slice
36	19:41:57.054524	10.41.250.42	H264	1442	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48223, Time=480665061 FU-A
37	19:41:57.054673	10.41.250.42	H264	1442	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48224, Time=480665061 FU-A
38	19:41:57.054673	10.41.250.42	H264	369	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48225, Time=480665061, Mark FU-A End
39	19:41:57.087667	10.41.250.42	H264	1442	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48226, Time=480668058 FU-A Start:non-IDR-Slice
40	19:41:57.087668	10.41.250.42	H264	96	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48227, Time=480668058, Mark FU-A End
41	19:41:57.120904	10.41.250.42	H264	386	PT=DynamicRTP-Type-96, SSRC=0xC1CE6595, Seq=48228, Time=480671056, Mark non-IDR-Slice

The packet details pane shows the following information for the selected packet (Frame 27):

- Stream setup by RTSP (frame 7)
- 0020 01 0b c3 f4 58 84 05 80 7b f7 80 60 bc 56 1c a6 ...X...{...V...
- 0030 17 a2 c1 ce 65 95 7c 85 88 83 00 00 4f fe 1f c2 ...e|...0...
- 0040 2a 28 00 08 0f fc 00 11 a0 8e f3 f2 11 d1 8b db ...(...)...0...
- 0050 c0 62 b5 2f 4d dd cc a9 8a 77 ff a1 c0 01 13 08 ...b/M...w...
- 0060 cf 77 29 1d 58 9d 82 0b a0 60 ea ae 5d 70 f4 df ...w).X...|p...

Summary: Frame (1442 bytes) | Unescaped RSP Data (1386 bytes)

```
FU-A
FU-A
FU-A
, Mark FU-A End
FU-A Start:non-IDR-Slice
, Mark FU-A End
, Mark non-IDR-Slice
, Mark non-IDR-Slice
, Mark non-IDR-Slice
, Mark non-IDR-Slice
, Mark non-IDR-Slice
FU-A Start:IDR-Slice
FU-A
, Mark FU-A End
, Mark non-IDR-Slice
, Mark non-IDR-Slice
, Mark non-IDR-Slice
, Mark non-IDR-Slice
FU-A Start:non-IDR-Slice
```



# THANK YOU

Any Questions?