Expertise
and insight
for the future

Danyal Nasir

Raspberry Pi and IP Camera

Metropolia University of Applied Sciences

Bachelor of Engineering

Electronics

Bachelor's Thesis

14 May 2020

Author Title Number of Pages Date	Danyal Nasir Raspberry Pi and IP Camera 33 pages + 2 appendices 14 May 2020
Degree	Bachelor of Engineering
Degree Programme	Electronics
Instructor	Janne Mäntykoski, Senior Lecturer

The main objective of this project was to create a program for Raspberry Pi that shows live video feed on a display as a form of surveillance using an IP (Internet Protocol) camera. The goal was to try to create a simple program using the portable Raspberry Pi and establish a connection with an IP camera over the local network and be able to acquire live video stream and deduce how secure it is.

This project consists of a Raspberry Pi 3B+ model as the main central processing device. The other main device involved in this project is a TP-Link NC450 IP camera which will be used to record the live video streaming sessions. Any screen which supports VGA or HDMI can be used for the display. The Pi model is powered by a Broadcom BCM2837B0 quad core A53 processor which is 64 bit. It consists of GB Ethernet port and a wifi chip for connecting to a network. The TP-Link NC450 camera is a wireless HD PTZ camera with a 75 degrees lens view. The camera device has an Ethernet port for wired connection and built in wifi for wireless network connectivity. The camera uses H.264 AVC video codec standard.

The main connection established between the camera and Pi is done mainly by RTSP. The camera has its own RTSP stream link which allows it to connect to RTSP servers online. An important software library, FFmpeg, is installed in the terminal of the Pi. This package acts as a H.264 video encoder/decoder and compresses the AVC format from the TP-Link NC450. The encoded format can then be used in a RTSP stream in a python script being run in the Pi. The script runs the stream and is able to display the live video feed.

The project resulted in a successful display of live video stream using RTSP. The stream was able to run up till native 30 fps as that is the maximum supported frame rate by the camera. There was video latency of less than a second in the display. The overall project can be employed as a means of private indoor surveillance system. Using the Pi as a headless system, the video stream can be viewed remotely.

Keywords	Raspberry Pi, TP-Link NC450, Python, Motion Eye OS, OpenCV, V4L, FFMPEG, RTSP, SIP

Contents

List of Abbreviation

AAC	Advanced Audio Coding
AVC	Advanced Video Coding
CLI	Command Line Interface
DVB	Digital Video Broadcasting
GCC	GNU Compiler Collection
GUI	Graphical User Interface
HDMI	High Definition Multimedia Interface
HDV	High Definition Video
HEVC	High Efficiency Video Coding
IEC	International Electrotechnical Commission
IoT	Internet of Things
IPTV	Internet Protocol Television
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	Telecommunication Standardization Sector

MPEG	Moving Pictures Expert Group
ONVIF	Open Network Video Interface Forum
OpenCV	Open Source Computer Vision Library
Pi	Raspberry Pi
PS	Program Stream
PTZ	Pan Tilt Zoom
Py	Python
RTP	Real Time Transport Protocol
RTSP	Real Time Streaming Protocol
SD	Secure Digital
SIP	Session Initiation Protocol
SSH	Secure Shell
TS	Transport Stream
USB	Universal Serial Bus
VNC	Virtual Network Computing
V4L	Video4Linux

1 Introduction

As the world has been progressing and moving forward with growing modern technologies and innovations, at the same time security has always remained a big threat and consideration. Many ideas and practices have been introduced over the years to improve security. The term security is not limited to one entity, its meaning is vast and can be described and explained into a lot of branches. However, in this particular project we will be focusing on security as an evidence of physical and/or potential risks on private grounds and security as a measure taken by installation to ensure safety from hostile influences. When it comes to monitoring by installation or deployment, there are several methods which have been devised over the years. A good example can be a watch dog hired for security measures to look after a private property. Although it is an effective way to monitor surroundings, like many things it too has its pros and cons. This project considers the introduction of camera surveillance systems. Depending on the nature of the enviroment being monitored, a camera can be installed strategically. It can be mounted and drilled on walls/ceiling, or just simply placed on a table top counter or a shelf. The main idea is for the camera to record or live stream footage and send it to a central system which is then viewed on a screen display. In the olden times, wired setup was common and coaxial cables were used to tie the wires up. Now with advanced systems, wirless tech has been integrated and we can say goodbye to old fashioned wired connections. There are many different types of camers used in surveillance systems. Some of these include box, bullet, dome, day/night, thermal FLIR(Forward Looking Infrared) and IP(Internet Protocol) cameras respectively.

This project has opted to use a wireless PTZ (Pan Tilt Zoom) IP camera for recording the live stream and Raspberry Pi as our main system which will then send the live footage to a HD (High Definition) display. The main system being used is a Pi 3B+. This whole setup is very cost effective as Pis are cheaper and convenient to use, being pocket sized. The system will be run on Linux OS and will imply the use of a python sript which will be able to display the live stream on a display connected via HDMI/VGA. The IP camera being used is a H.264 wireless camera from TP-Link, it also supports nightvision.

2 Theoretical Overview

2.1 Linux

Linux is an operating system platform affiliated with Unix-like systems. These systems are based on Linux Kernel which is furthermore an operating system which was released in September 17, 1991 by Linus Torvalds. They are usually packaged as Linux distributions. What originally started off for desktop computers soon was being used widely in many other platforms. Now it is the only operating system being used in TOP500 supercomputers. [1.]

2.1.1 Introduction to Linux

Linux can be best described as an OS which runs on packages and distributions which are under open source licenses. It was written in C language. Debian, Ubuntu and Fedora are some of the widely used distributions which are used personally and commercially as well. It was initially developed on intel x86 platform but now has been based on a vast amount of different platforms. The UNIX like OS, in other words a Linux Kernel, is mainly responsible for a lot of important tasks for the Linux system. Some of these include handling file systems, accessibility to hardware and peripherals, processing control and networking. Drivers can be separately added to the Linux system via adding them as modules. Same way separate libraries and other software programs can be added as modules. The user interface for commonly used desktop systems employ the use of CLI(Command Line Interface) and GUI(Graphical User Interface) with GUI being the default. CLIs are commonly used via the terminal emulator which uses text for input and output to control the many various tasks and installations for the Linux kernel system. When it comes to programming, distributions on Linux support many programming languages thanks to GCC(GNU Compiler Collection). GCC allows languages Ada, C, C++, Go and Fortran to be compiled. However, many other languages like Php, Python, Java, Ruby etc also are readily available for Linux distributions. This is possible due to cross platform reference implementation support these languages provide for Linux. [1.]

2.1.2 Linux vs MacOS vs Windows

The three main operating system used throughout the globe are Windows, Mac OS and Linux Os respectively. The three very rightly differ from each other and each OS has its own pros and cons. Both, Mac OS and Windows started off as GUIs, whereas Linux was first designed for GNU developers. Windows is the most widely used OS in the globe and accounts for 90% usage with Mac OS leading at second with 7% and Linux distributions a mere 1% [2]. Because of the fact that both Windows and Mac OS have a huge user base, they are most prone to malware and spyware too. Linux on the other hand has a very low probability of catching malware. Mac OS is the most costly to use as Mac OS users are forced to purchase a Mac system built by Apple Inc.

Compared to Windows and Mac OS, Linux has a different file structure. The code base is completely different. The drives are all stored over a single tree file. The command prompt for windows, also known as Windows Command Processor, is used to execute commands and run other administrative functions. For Mac OS, it has a terminal interface which is used to run commands and explore directories. Just like Mac OS, Linux also uses terminal to run commands, explore directories, and install packages and other administrative functions. Figure 1 below explains the differences in registry between the three operating systems. [3.]

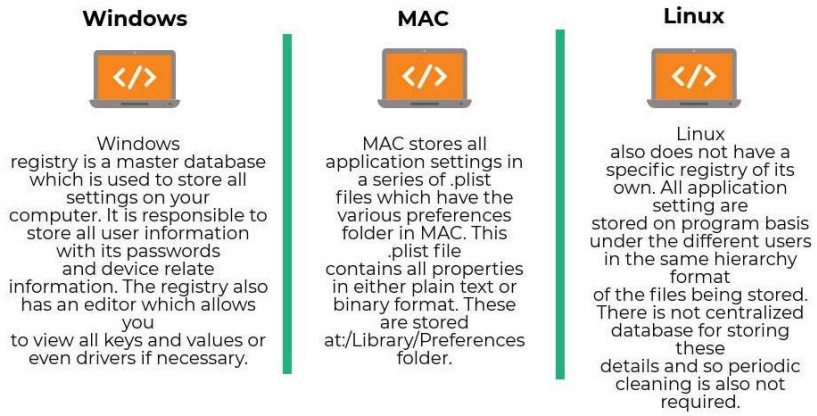


Figure 1: Registry comparison between the three operating systems. Reprinted from [3.]

For some users, having interchangeable interfaces could be a key factor in determining which OS to opt with. Having easily switchable GUIs means users can easily run multiple programs simultaneously without having much knowledge of programming. It also provides visual feedback and makes it easy to learn and operate the system. The figure 2 below contrasts the differences between interchangeable interfaces which would benefit the users [3].

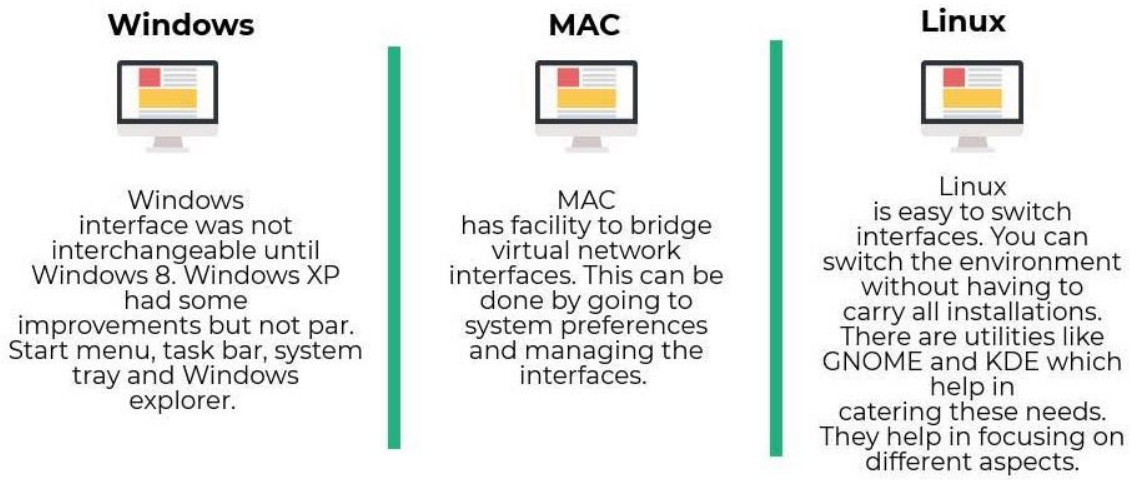


Figure 2: Ease of interchangeable interface in each OS. Reprinted from [3.]

2.1.3 Raspbian

Raspbian is a popular OS (32-bit) designed specifically for Raspberry Pi. It is based on Debian package, which like all other Linux distros, is open sourced and free to use. [4] Although Raspbian's first build came into light in June 2012, it was not until September of 2013 when the first version of Raspbian was publicly released. There have been several versions of Raspbian put out over the years. The first being Wheezy followed by Jessie, Stretch and Buster being the latest. The versions are actively under further development and constantly being updated to support the many different models of Raspberry Pis. The Wheezy version allowed support for the Raspberry Pi 1/1+ as that was the only model available at that time. With the introduction of Pi2 models, newer versions of Wheezy were made available for support. Two years after the first Wheezy release, a new version, Raspbian Jessie, was introduced with initial supports for both the 1/1+ and Pi2 models. The following year 2016, both the Pi3 and Pi0 models were developed with Jessie support. In August of 2017 the third entry into the Raspbian family, Stretch, came into light which supported all Pi 1/1+, Pi2, Pi0 and Pi3 models

respectively and later on the Pi3+ model too. The latest version of Raspbian, Buster, was released at the same time as the latest Raspberry Pi model 4. It is due noted that all these versions of Raspbian are still actively running and under constant development.

With the high optimization capabilities of this OS, it makes it very suitable for low performance ARM Cortex CPU based Raspberry Pis. Raspbian employs the use of PIXEL (Pi Improved X- Window Environment Lightweight) for the desktop environment. It comprises of a latest LXDE desktop platform. [4.] The idea is for the Raspberry Pi users to be able to enjoy a desktop feel environment on their Pis which have low processing speed and memory. Figure 3 below shows the Raspbian desktop of Pi 3B+ used in this project.

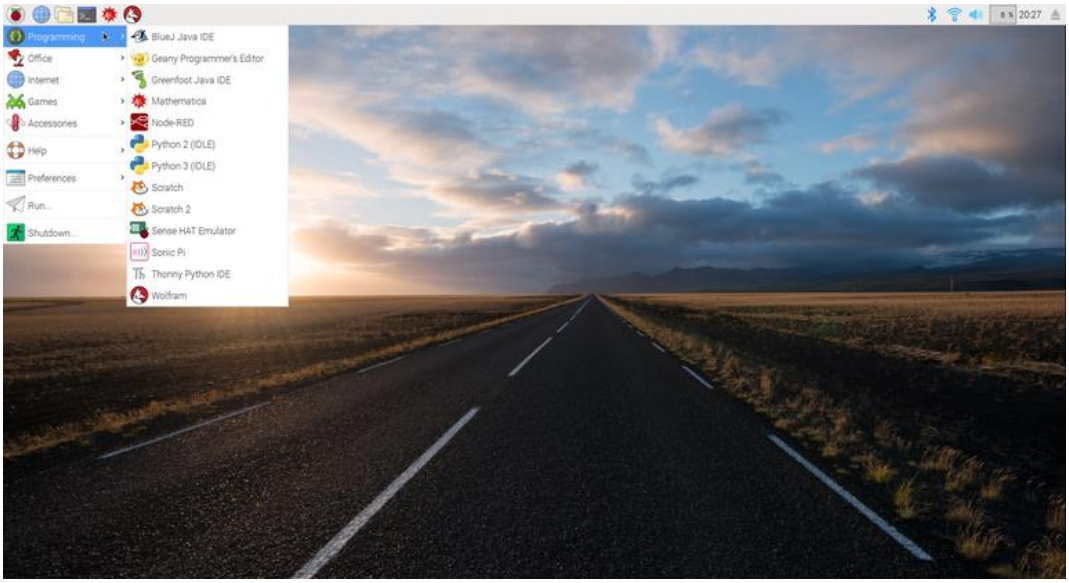


Figure 3: x86 image taken from the Raspbian desktop of my Pi 3B+

2.1.4 V4L

V4L which stands for Video4Linux is a set of computer programs (device drivers) which are written in C language and are used for real-time video capture for Linux. It also functions as an API (Application Programming Interface) for Linux. It has a support feature for most IP cameras and other TV/radio related device interfaces. Programmers are able to add video capture support to the applications they are using which are able to run on V4L armature. V4L was initially brought up for 2.1.x cycle of Linux Kernel with added support in the form of V4L1. Later a newer upgraded version known as V4L2

was introduced for 2.5.x Linux Kernel which smashed some design bugs. [5.] The latter includes a compatibility mode for V4L1 apps. Each IP camera has its own URI (Uniform Resource Identifier) which helps V4L drivers to identify the particular device. *http://username:password@host.domain:8080/path/file?action=stream* is the usual representation of a URI of a device. Some of the software programs which are supported by V4L include Zoneminder, FFmpeg, VLC media player, libav, OpenCV, Skype, Motion and many more.

2.1.5 OpenCV

OpenCV (Open Source Computer Vision Library), which was developed by Intel in 1999, can be best described as an open source software library used mainly for machine learning and computer vision. It is natively written in C/C++ language. OpenCV is compatible with Windows, MacOS, Linux, FreeBSD, NetBSD and OpenBSD. OpenCV contains more than 2500 algorithms which include both machine learning and computer vision algorithms. C++ is the primary interface for most of the algorithms in the software library although the API for binding interfaces of other programming languages like Java, Python and MATLAB can be found at the OpenCV official website. The APIs are available in the online documentation tab. [7.] Having grown popular over the years, its wide range of applications includes facial recognition system, motion detection system, image processing, mobile robotics, gesture recognition, video surveillance system, image stitching, object identification, augmented reality etc.

2.1.6 FFmpeg

Similar to OpenCV, FFmpeg is an open source software library comprising of many libraries and programs which aim in handling only multimedia streams and audio/video files. At the center of FFmpeg is the FFmpeg program itself which employs the use of command line tools. The tools have the ability to encode, decode, transcode, mux, demux, stream, filter etc. Even the oldest formats of files are supported by FFmpeg. Its high portability means it is available in many operating systems which include Windows, MacOS, Linux, BSDs, and Solaris etc. Some of the important FFmpeg libraries include *libavcodec*, *libavformat*, *libavutil*, *libavdevice*, *libavfilter* and *ffmpeg*. The self named *ffmpeg* is perhaps the core of FFmpeg. It is a command line based tool which is used for transcoding and conversion of multimedia files between formats. There are

two more command line tools, *ffplay* and *ffprobe*. *Ffplay* is a media player which employs the use of FFmpeg libraries. *Ffprobe* is a multimedia analyzer which displays media information. [8.] When utils support for v4l is added to FFmpeg, the input device option *–use_libv4l2* is made available for usage. The v4l-utils support can be created by applying the *–enable-libv4l2* option. When a v4l2 device is connected to the system bearing FFmpeg, it will be identified and created as a file device node by Linux. The device which can be an IP camera or any other video surveillance device will exist in the format */dev/videoN*. N can be any number assigned automatically to the device (usually 0). For testing the frames the v4l2 device can provide and the widthxheight size it supports, the list option *–list_formats all* can be used. Different devices support different standards and video codecs hence to find out all the standards, the list option *–list_standards all* can be applied. Listings 1 and 2 below are examples of v4l devices being used with FFmpeg. [15.]

```
ffplay –f video4linux2 –list_standards all /dev/video2
```

Listing 1: A listing command being used to list the standard(s) the v4l2 device supports

```
ffmpeg –f video4linux2 –framerate 60 –video_size hd1080 input_format mjpeg –i
/dev/video2 out.mpeg
```

Listing 2: ffmpeg command used to record input from video2 at 60 fps 1080 video size

2.2 Introduction to Raspberry Pi

2.2.1 What is Raspberry Pi

A Raspberry Pi also commonly abbreviated to Pi or Raspi can be best described as a pocket sized computer developed by the British Raspberry Pi Foundation. Its size can be compared to a credit card and is easy to carry anywhere. All the Pi needs is a set of hardware plug ins which include a keyboard and mouse, a monitor display which can be vga/lcd/led etc and a power output to operate. Users can be able to surf on the internet, program using many popular languages, play and even develop their own games, create spreadsheets and also capture real-time videos using packages and separate software programs. Perhaps the most beneficial aspect of using a Pi comes from its ability to be able to interact with any IOT device and the outside world. Users can install various open source packages and programs which allows the Pi to be used

in many digital projects. The raspberry Pi model being used for this project is the model B+ from the Raspberry Pi3 family. It employs the use of a 64-bit quad-core 1.4 GHz ARM Cortex-A53 processor. The Pi 3B+ board has four USB 2.0 ports along with an ethernet port. For audio output, a 3.5mm jack is available. Composite video jack, a HDMI port and a DSI(Display Serial Interface) for LCD panels are supported. The model 3B+ has a 40 pin GPIO. There is no need for a separate wifi dongle as wifi/bluetooth is already integrated. Figure 4 below shows the Pi used in this project.

Figure 4: Raspberry Pi 3B+ model used for this project

2.2.2 Python Programming

Python is a programming language first made public in 1991. Python has a standard library and programmers would notice that python's main programming paradigm in-

cludes structured, functional and object-oriented programming. Other added extensions means paradigms like logic programming can be utilized in python. With a simpler syntax and grammar, python has become popular over the years. Compared to other programming languages, python uses the English language rather than punctuation for formatting. Some expressions and functions used in python are similar to those used in other popular languages like C and Java, while others are not. The numerous built in high level data structures means programmers can easily learn and program using python. The language is run under an interpreter rather than a compiler like other programming languages. Also it is important to note that one does not have to declare a variable in python. An example of a python script is shown in listing 3 below. [6.]

```
n = int(input('Type a number, then its factorial will be printed: '))

if n < 0:
    raise ValueError('You must enter a positive number')

fact = 1
i = 2
while i <= n:
    fact = fact * i
    i += 1

print(fact)
```

Listing 3: An example script written in python. Reprinted from [6.]

Figure 5 shows the Thonny IDE python environment in Pi 3B+ used in the project.

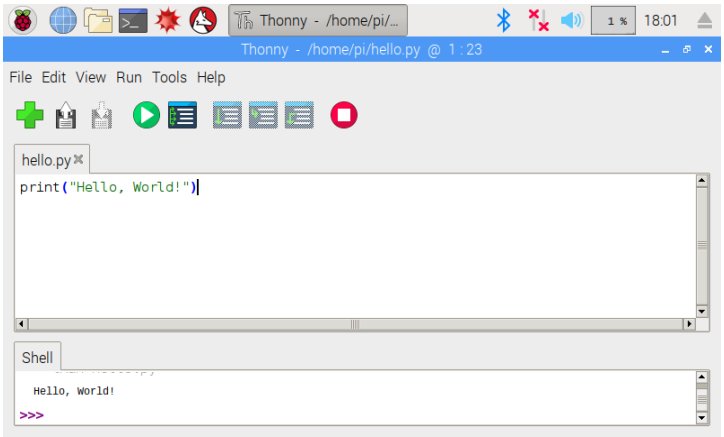


Figure 5: Snippet of Thonny IDE on Raspberry Pi 3B+ used in this project

2.3 IP Camera

IP camera which stands for Internet Protocol camera is a type of digital camera which receives data and then sends it over a local LAN. As the name implies, an IP camera

receives data and sends it in the form of image or video using the internet. When IP cameras were relatively new, they were connected to the network via a wired connection like the Ethernet. Newer more improved wireless IP cameras are widely used now. Major uses include for monitoring and/or for security reasons and surveillance systems. IP cameras can be used privately in private households and properties or commercially on a large scale in industries and important organizations. Previous generations of cameras used older TV formats like NTSC, PAL and SECAM. Now the formats range from HD, 1080p Full HD, 4K Ultra HD and 16:9 formats. One of the main reasons why IP cameras have become common in households is how easy the installation process is. Users can easily mount cameras on walls/ceilings/roof or simply place them on top of flat surfaces and shelves. A big advantage of using wifi IP cameras is that users can easily access the live video feed and control their camera/cameras on their mobile devices. There are mobile applications relative to the camera in use through which users can control the camera given that they are connected to the local network via wifi or Ethernet.

2.3.1 IP Cameras and Video Codec Standards

As briefly mentioned in the introduction, there are several different types of IP cameras. They range from dome cameras, bullet cameras, box cameras, POE cameras and PTZ cameras etc. Some cameras are manufactured for outdoor use while others are best suited for indoor surveillance. The installation process may differ for each type of camera. Also it is due noted that before purchasing an IP camera, there are some things to take into consideration such as the resolution of the camera, lens of the IP camera, video recording system of the camera, dynamic range and low light capability. One of the major differences in different IP cameras can be identified by the audio/video standards they actually support. With newly developed IP cameras, they are bound to support the latest audio/video compression standard. Therefore it is important to understand and compare the respective audio/video standards supported by IP cameras.

MPEG-2

MPEG-2 is best defined as a digital audio/video compression standard used for storage or transmission of digital video and the associated audio format. Its equivalent is the H.222/H.262 as per ITU (International Telecommunication Union). It is important to

note that MPEG-2 is not to be confused with MP2, the audio layer two of MPEG-1. MPEG-2 was introduced by MPEG (Moving Pictures Expert Group) as the second standard. MPEG-2 has standards which come under the (ISO/IEC 13818). There are eleven parts with each part covering the specifications of MPEG-2. Parts 1, 2, 3 and 7 are explained as follows: [9.]

- Part 1 is the Systems section of the MPEG-2 and is also known as H.222 as per the ITU. This section describes two container formats known as TS (Transport Streams) and PS (Program Streams). TS are a digital media container format which is mainly used for storage and transmission of audio and video data files. TS are responsible for carrying less reliable streams. This transmission is employed by satellite broadcast and broadcast systems like IPTV (Internet Protocol Television), DVB (Digital Video Broadcasting) and HDV (High Definition Video). TS use .ts, .tsv and .tsa as filename extensions. TS use M2TS for Blue-Ray disks and HDV. The other container format, PS, is responsible for more reliable random access storage such as flash memory and hard disk drives. PS are used for multiplexing digital audio and video. PS use .mpg, .mpeg, .m2p and .ps filename extensions which are then extended to VOB, MOD and EVO. PS are commonly used for DVD- Video and HD DVD disks. [9.]
- The second part of MPEG-2 standards is the video section also referred as H.262 (ITU- T Rec). It is very similar to the previous MPEG-1 standard. It has support for interlaced video which means for a certain video display, it can double the received frame rates without using extra bandwidth. This way the video contains two fields of a frame captured consecutively. This makes it ideal for analog broadcast televisions as flicker is reduced too. MPEG-2 video decoders are downwards compatible and devices with MPEG-2 video decoders can also play MPEG-1 video samples. [9.]
- The audio section is the part 3 of the standard. It allows multichannel audio up to 5.1 channels. It is also downward compatible. This allows audio to be compressed and MPEG-1 decoders can decode and the result is that stereo audio tracks can be played. [9.]
- Part 7 which is also known as AAC (Advanced Audio Tracking) is a distinct audio format. Unlike the part 3 audio section, AAC is not downwards compatible. However, with support up to 48 channels at a higher sampling rate, multilingual and multi program capabilities, AAC proves to be more efficient than its predecessors. [9.]

MPEG-2 is used in a wide variety of applications. Applications range from DVD-Videos, HDVs, MOD, DVB, ATSC, Blue-Rays etc. However, when Blue-Rays were relatively new, MPEG-2 video was commonly used. Now Blue-Rays employ the better H.264 which is going to be explained in detail briefly. Only the part 2 of MPEG-2 standard allows support for Blue-rays. Out of the two container streams, TS are used on Blue-Ray disks. Parts 3 and 7 do not support Blue-Ray which limits the MPEG-2 capabilities especially knowing how common and widely used Blue-Ray disks are in the present. Common filename extensions used in MPEG-2 are .mpg, .mpeg, .mp2, .mp3 etc.

H.263

H.263 is a video codec which was designed for communication over video, video conferencing. It was designed to operate at a low sampling bit-rate. It is a successor to the previous standards H.261 and H.262 and therefore was designed with improvements. It consists of three versions to date. The first version released in 1995 replaced the H.261. It was further improved into version H.263v2 and H.263v3 respectively. H.263 was also used as a basis for MPEG-4 part 2. H.263 can be used for bidirectional visual communication. Because it was designed for lower bit-rate sampling video conferencing, operating at more than 50kbps data rate may disrupt performance of the system and low end computers. It has a support for FFmpeg which means that libraries like libavcodec can be used to decompress the H.263 video format thus it is used by media players like VLC and MPlayer. Moreover, H.263 is also used by internet application giants like Youtube, MySpace, Google and Google Video for their flash videos. H.263 also sub sequentially saw its use in protocols like RTP (Real Time Transport Protocol), RTSP (Real Time Streaming Protocol) and SIP (Session Initiation Protocol). Comparing with H.261, H.263 provides a better video quality. [10.]

H.264

H.264 is the part 10 of MPEG-4 standard started in 2003. It was developed jointly by ITU-T and ISO/IEC. It is the most popular and widely used video codec. It is commonly referred to as AVC (Advanced Video Coding). It is mainly used for recording, compressing and distribution of HD video content. The main aim for the development of this video codec was to enhance the video compression capabilities at even lower bit-rate compared to older H.263 and H.261 video codecs. This was made possible by the ad-

dition of the DCT (Discrete Cosine Transform) integer. Flexibility was another achievement for the H.264 as it intended on offering higher/lower sampling bit rates, higher/lower video resolution, DVD storage and be able to be employed by a vast amount of internet applications and online video streaming services. As previously mentioned, it is one of the three formats which are supported by Blue-Ray disks and it is the most widely one used to do so. Popular streaming services such as Netflix, Hulu, Amazon Prime Video, Youtube and ITunes Store all greatly use H.264. In the CCTV video surveillance market, many CCTV cameras and IP cameras have included the H.264 format for the latest products. H.264 has mainly three so called profiles, main, extended and baseline profiles. Main profile is used for broadcasting like HDTV. Extended profile is used for video streaming purposes. Baseline profile is employed for services such as video conferencing. [11.] Comparing H.264 with H.263, the latter was designed to operate at low bit rates only whereas H.264 has the option for encoding both high and low bit rate videos. H.264 also allows lower overall cost at a much more efficient video yield. Compared with its predecessor, H.264 requires lower bandwidth, lower storage for video compression and lower download durations. Although H.264 has had a major improvement in the video compression and streaming industry, it still struggles for UHD (Ultra High Definition) content. It consumes more bandwidth while providing lower fps (frames per second). To counter the issues, an ever better and more enhanced H.265 video codec has been designed.

H.265

H.265 or MPEG-H part 2 also commonly referred to as HEVC (High Efficiency Video Coding) is a standard of video compression. It is a successor to the widely used and popular H.264. Its design further enhances the video compression capabilities of the H.264. This meant that H.265 was able to operate at even lower bit sampling rates while yielding much better video quality. It offers support up to 8k UHD and is able to compress 4k UHD video content on approximately 10 Mbps. Comparing with AVC, HEVC is able to better compress data from 25% to 50%. Similar to AVC, HVEC also employs the use of DCT but with the addition of DST (Discrete Sine Transform). [12.]

The working principle is same as that of the H.264. Both operate in such a way as by identifying redundant areas in a video sample by comparison between different parts of a frame. This process takes place between single and consecutive frames. Instead of replacing redundant places with original pixels, they are replaced by short description. Now the major difference that separates HVEC from AVC is the fact that HVEC uses pixel sizes from 16x16 to 64x64 for pattern comparison. Other changes also include enhanced motion vector prediction and motion compensation filtering. [12] However, H.265 as a whole is very costly compared to the cheaper H.264. AVC is still the most widely used video compression standard followed by HVEC. Comparing the three H.263, H.264 and H.265, as the numbers in ascending order suggest, the increasing number of video codec is more efficient than the previous one making HVEC the most efficient out of all the others. Example of H.263 device is shown in figure 6 below [16].

Figure 6: A video conferencing phone device that supports H.263/H.264. Reprinted from [16].

2.3.2 RTSP

RTSP which stands for Real Time Streaming Protocol is an application level data transfer protocol used to transfer multimedia streams over the network directly. As the name implies, the media being transferred and stream is done in real-time. The protocol establishes communication between the two endpoints, the client and the server. Multi-

media streams which can be audio and video can be transported either from client to a server (voice/video recording) or server to client (video on demand). RTSP servers do not stream media directly. Usually the RTSP servers employ the use of other data transfer protocols such as RTP (Real Time Transport Protocol) and RTCP (Real Time Control Protocol). The media being transferred can be in the form of saved clips or live video feeds. [13.] RTSP can be considered similar to HTTP as both use TCP (Transmission Control Protocol) for maintaining end to end connection. However, unlike HTTP where no session information is retained by the client, RTSP uses an identifier which labels a session for a maintained server. During a session, the protocol can open and close many other transport connections to deliver RTSP requests. Where RTP and TCP are mainly used for streaming and multimedia transport, UDP (User Datagram Protocol) is also used for data transfer and is connectionless. [13; 14.]

RTSP mainly supports three operations. These include a proposition of a media server to an existing conference. With this possibility, a media server can be requested to be added into a conference. It can then either record a part of or all of the conference or play back the media content for the existing conference. This makes it very ideal for online teaching purposes and business meetings. The protocol can also be able to add media to a live presentation. The server can inform the client about any additional media being added to the presentation. A common operation the RTSP follows is the fetching of media from a server. The client may fetch a description from the server using HTTP or other means. In the case of multimedia streams, the description will contain port numbers and multicast addresses. [14.] There are several important basic RTSP requests which indicate what method to be used on the source. These are identified by the Request-URI. All of these requests are case sensitive. They are mentioned and explained as follows:

- OPTIONS request will return all and any other request methods that the server can accept. An OPTIONS request can be made at any time. OPTIONS request can be issued in both directions, from client to server and vice versa. It is a required request for the connection. [14.]

- DESCRIBE request accounts for the media initialization of RTSP. It includes description of a presentation in the form of reply data. The data is typically in the form of SDP (Session Description Protocol) format. The description of resource is fetched by the server as a response. The RTSP URL (rtsp ://) is found

in the DESCRIBE request method. The DESCRIBE request is a recommended one and it takes the direction of client to server only. [13; 14.]

- A SETUP request is responsible for the transportation of the media being streamed. The request contains a stream URL and a specifier which contains ports for fetching RTP and RTCP data respectively. The parameters which are acceptable to the client will be specified and provided in the response by the server. The SETUP request is used in the client to server direction and it is a required method. The SETUP request must be successful before any PLAY request can be made. [14.]

- PLAY request will allow the media stream to be played after the SETUP request has been successful. PLAY requests can be edited using range and will play the stream in the specified range. As soon as the range is completed, the playback is automatically paused. If no range is specified, PLAY request will continue to play media from beginning to end. This request can also allow queuing which means that if a PLAY request is active and another PLAY request is issued, it will play the second request after the first one has been completed. This method is required and operates in client to server direction. [14.]

- PAUSE request as the name suggests will immediately stop or pause the playback of a media stream. It can stop playback of one or more streams depending on the number of streams being active. If a PAUSE request is used on a media stream which is named, it will stop the equivalent stream. Media stream can resume playback with the PLAY request after being paused. For audio, PAUSE request will cause muting. Range can be specified in the request thus the stream will be halted in the specified range. This request method is recommended. [14.]

- RECORD method allows media stream to be recorded. A range has to be specified for the request to record a certain time stamp of the description. If media stream is already active, recording of playback will take into effect immediately. The recording can be stored under a URI which can be requested or not. If a path is not specified, server will prompt for a request URI and the recording will be stored in requested URI. The RECORD request method is optional. [14.]

- ANNOUNCE method is used for two things. When the client sends the request to server, the request posts the presentation description to the server. The request URL recognizes the media object and that is also displayed by the ANNOUNCE request. The second task for the request is to update the session description when server sends it to client. This update process happens in real time. This method is optional. [14.]

- For a specific URI, the TEARDOWN request will terminate the transmission of media stream. Session identifiers relative to their respected sessions will no longer be valid and in order to initiate a new session, a SETUP request with the required parameters will have to be issued. TEARDOWN request can only be sent from client to server and it is a required method. [14.]

- GET_PARAMETER request allows reclamation of parameter of the media stream or presentation. Reply-response pair is taken care of by the implementation. If no entity is used while making this request, it can be used to check the time it takes for network packets being sent and received by the server and client. The request is optional and can be sent from client to server or vice versa. [14.]

- The SET_PARAMETER request asks a value of a parameter of the media stream or presentation to be set. This request is usually used for determination of any other failed requests. A single parameter set using this request method will allow client to identify the reason of a failed request. However, if more than one parameter is being set, it is important to confirm that all parameters are set successfully so the server can act on the request. This request is optional in RTSP. [14.]

- REDIRECT request is used to notify the client to be redirected to another location for the server. It includes the header location which will show the path to where requests should be issued by the client for that particular URL. Like other requests, a range can be set to let client know when to redirect to a new location. If no range is set and client opts to continue transmission of media stream for the active URI, TEARDOWN request must be used to terminate the current session and a new SETUP request will have to be issued for new session at the requested host. [14.]

All of the above requests in RTSP can be sent from client to servers. All of the methods operate on presentation and stream objects with the exception of SETUP, which can only operate on stream. RTSP has brought about a major step forward in the computing industry. Before RTSP, media over the internet had to be saved and downloaded in order to be viewed. RTSP made it possible for multimedia streams to be accessed over the network. There are a lot of reasons to explain why this protocol has grown to be successful and so widely implemented now. The protocol is extendable meaning newer methods and parameters can be added. Due to its similarity with HTTP, it can be parsed by HTTP parsers. RTSP is transport independent and can use either unreliable UDP or reliable TCP for transporting media streams. RTSP has ability to handle multiple servers at one time. Several ongoing sessions can be established with servers by the RTSP client. Furthermore RTSP can also manage VCR (Video Cassette Recording) devices and those devices that allow recording or playback only. As mentioned earlier, RTSP is useful for online communication and services mainly video conferencing, online lecturing and can be used to allow digital editing on media streams remotely. RTSP has control over servers in such a way that they are able to play and stop a server independently. [14.]

2.3.3 How RTSP Communicates With IP Camera

Every media stream or IP camera devices have their own RTSP URLs. In this context, a presentation description can be explained as a set of media streams being controlled. The IP camera has its own RTSP URL which will allow RTSP clients to identify it. It is important to understand that every presentation and media stream has a presentation description file. This file defines the contents and properties of the IP camera's media stream and basically the overall presentation. The description file can be acquired by the RTSP client through email or HTTP. It may not be stored on the server. A presentation may contain more than one media stream, similarly a presentation description can comprise of more than one presentation. Assuming a presentation description file is reporting one presentation as it is in this case with IP camera, the file comprises of that camera's language being used, encodings, transport method(s) the server can handle and parameters. These parameters allow the RTSP client to decide which media is most applicable to choose from. When RTSP URL is identified and stream is being controlled by RTSP, the server handling is notified by the RTSP URL about the media stream being managed and any names stored on server by the stream. Audio and video streams of the camera can be found in different server locations which indicate

that media streams can be found on different servers. Apart from the description file consisting of media parameters, some other factors also need to be determined. These include the port number of the IP camera and network destination. In the event of live media streaming, the media server selects the port and multi cast address. [14.] Figure 7 shows the popular VLC media player being used as a RTSP client.

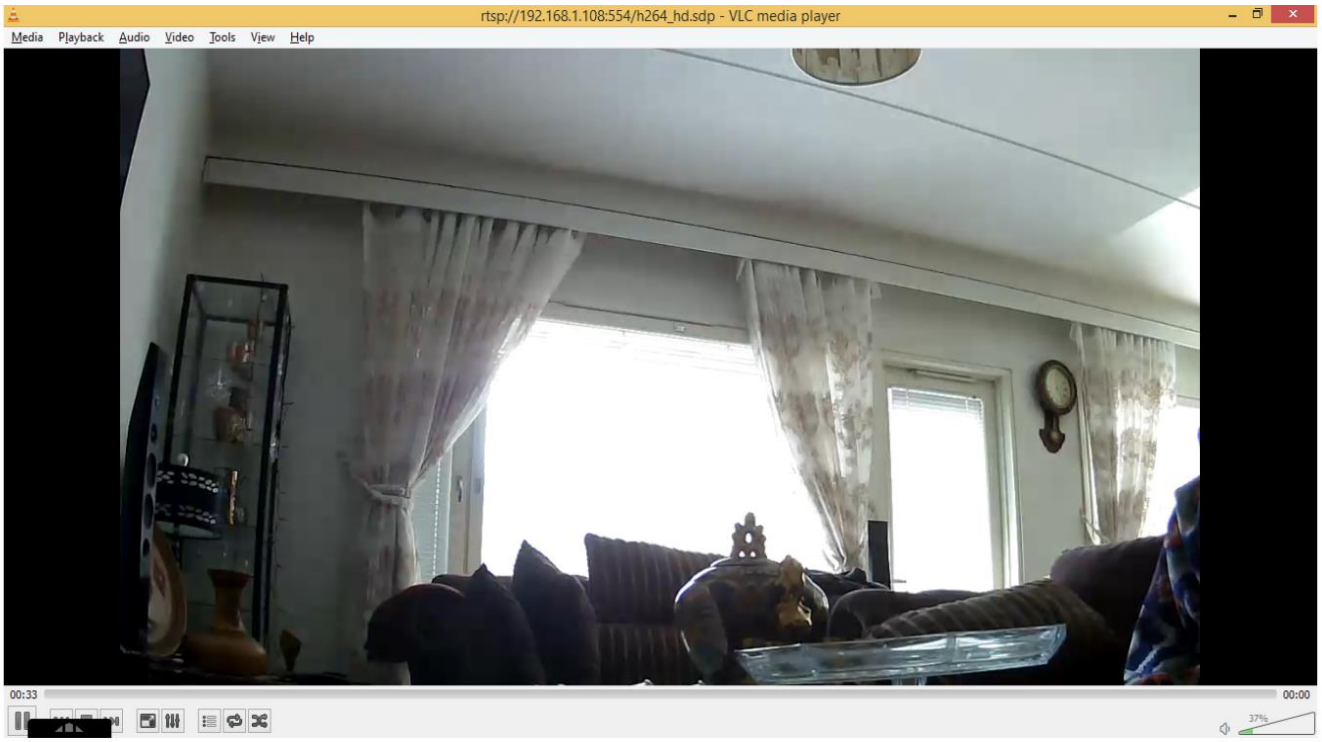


Figure 7: VLC media player is a perfect example of a RTSP client

When a RTSP client like VLC media player will send RTSP request to server which is connected to the IP camera source, it will dictate which request methods such as PLAY, PAUSE, RECORD etc the server can accept via RTSP. The server will send these requests back to VLC. The server which is streaming media from IP camera will then receive a description file request from VLC. The server will then send the description file to VLC which will contain media parameters, language and encodings. VLC will then issue a SETUP request and server will indicate which transport methods (UDP/TCP) it can support. After SETUP request is completed successfully, the session will be started when server sends bit stream to client. Bit stream will be transmitted to VLC using the transport method the server can manage and that which was specified in the SETUP request. The whole RTSP session process is shown in figure 8 below.

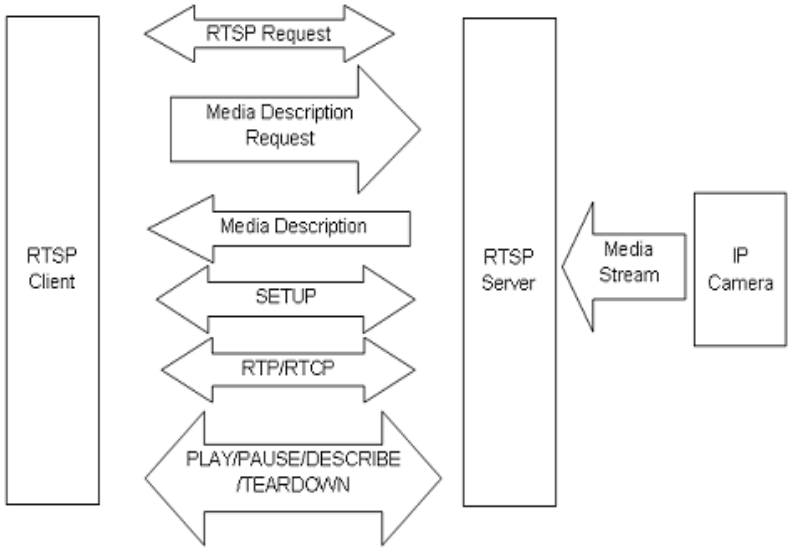


Figure 8: RTSP session process between IP camera and RTSP client

2.3.4 SIP

SIP (Session Initiation Protocol) is communication protocol of the application layer that is mainly responsible for establishing, maintaining and ending real-time end to end sessions in IP telephony. The protocol is mainly used in ordinary telephone connections, video calling, and voice calling over the internet. Instant messaging and text messaging also employ the use of SIP protocol. It is a text based protocol and shares many similarities with HTTP and SMTP (Simple Mail Transfer Protocol). Like RTSP, SIP session can also contain multiple streams of media content. SIP operates with other protocols such as SDP (Session Description Protocol), UDP, TCP and SCTP (Stream Control Transmission Protocol). SDP operates in providing media type and setup of media for SIP. SDP further uses RTP for carrying media streams such as audio and video data between terminals. SIP is able to hand unicast and even multicast streams. For an ongoing session, SIP can allow changes being made which involve changes to any port, adding in more recipients and addition or deletion of other streams. SDP is responsible for specifying the format of the stream, the codec standard of the stream and communication protocol. In a SIP session, there are SIP user agents and SIP servers present. User agents can be an IP phone, mobile phones etc and are endpoints in a SIP session. SIP servers allow other user agents to be located. SIP has 14 different request methods. Most common among these methods include

INVITE, ACK, CANCEL, BYE and REGISTER. A SIP URI is used to indicate a user's SIP number. In a SIP call session, there are 4 important steps that occur for a successful connection between 2 users. [18.]

1. Registration process happens when a user instigates their user agent for example an IP phone. Request method REGISTER is issued and the phone has to register to a SIP server. After it has been registered, it is now discoverable by other user agents. [18.]

2. The second step is the call establishment process. This step explains when one user agent tries to connect to another and a number of different request methods are issued. The SIP INVITE request starts the connection process. The message includes the receiver agent's SIP URI. It is sent to the SIP server where it identifies the intended receiver. The second message, SIP Response 100 (Trying), is sent to the receiver user agent by the server for confirmation of previous INVITE request. After the INVITE request is received successfully to the receiver user, the message SIP Response 180 (Ringing) shows up which informs the sender that the receiver user agent is alerting the user of the call. SIP Response 200 (OK) message is sent back to the sender when the receiver accepts the call invitation. The final step in establishment process is the ACK request sent by the sender to receiver after which the VoIP call is commenced by the receiver. [18.]

3. The actual VoIP call data between both users is being sent by transport protocols such as RTP. VoIP call can contain audio alone or audio and video together. [18.]

4. The last step in the session is to end the communication between the two users. This is achieved by the BYE request message. Any user can issue the SIP BYE request to terminate the VoIP call. When one user issues a BYE request, a response 200 OK message is sent to the other user. [18.]

2.3.5 ONVIF

ONVIF (Open Network Video Interface Forum) is an open standard and global forum which was founded by Bosch Security Systems, Sony and Axis communications in

2008. The main objective of ONVIF is providing and assisting of interfaces that support interoperability characteristics of physical IP-based security systems to effectively be able to operate with other products or systems in implementation or direct access. ONVIF provides such standardized interfaces that allow different organizations and brands to create IP-based security products that are able to work with each other. ONVIF is an open forum which is open to all organizations and manufacturers. ONVIF provides more than 12000 conformant products. In order for a product to be compliant with ONVIF, it must have at least one of the six ONVIF profiles. These profiles help identify the conformant products and if they are compatible with other products. However, there are many manufacturers and products that falsely claim to be ONVIF conformant. Each profile has distinct features that have to be present in an ONVIF conformant device and client. A conformant client that has Profile T will work with a conformant device with Profile T. There are 6 ONVIF profiles A, C, G, Q, S and T. Profiles A and C are used for access control whereas Profiles G, Q, S and T are employed in Ip video networking products. [17.]

- Profile A is used for control configuration. Devices with Profile A can fetch information and configure credentials and access. A client with Profile A will provide such credentials and configured access rules. [17.]

- Profile C is also used for access control. Like Profile A, devices and clients with Profile C can support door control access, site information, alarm management etc. [17.]

- Profile G is made for video systems and is mainly responsible for recording control and storage. A Profile G video device can record video over an IP network or the conformant device itself. A Profile G client will control the recorded video data from the Profile G device. [17.]

- Profile Q is used for setup and discovery of other conformant devices. A video networking device bearing Profile Q will be easily discovered by a Profile Q client. The client can also control the conformant device over a network. [17.]

- Profile S is mainly made for basic video streaming in IP based video devices and cameras. A Profile S video device can transmit video to a Profile S client over a network. A Profile S client is able to configure and control the video data

being sent from the conformant device. Profile S also includes PTZ controls, relay outputs and audio in for certain devices and clients that carry such features. [17.]

- Profile T is made for advanced video streaming which covers H.264/H.265 standards, motion alarm systems, imaging settings, metadata streaming and bi directional audio. Video devices of the H.264/H.265 standard that support imaging settings and motion detection are Profile T devices that also consist of on-screen display. Profile T also support PTZ controls, motion configuration and bi directional audio for devices. [17.]

2.3.6 Bonjour

Bonjour is a protocol created by Apple Inc that allows discovery and searching of other network devices. Hence it can allow communication between devices connected on the network. It allows for network devices to be found without the need for configuration of DNS servers and IP addresses entries. What was originally designed to help locate Apple devices and products within a single LAN now is also used between Windows and Apple to help share devices such as printers. Many IT tech firm giants now employ Bonjour as gateways and protocols that allow different networks to operate. [19.] Upon logging in to the IP camera's account being used in this project, it is found that the Bonjour name assigned to the camera is NC450 2.0-1ab651. When Bonjour is installed on Windows and after completing the setup, the respective Bonjour name can be detected on the Bonjour search list without entering any IP address.

2.3.7 Security

When the products and devices being used are only manufactured so they can protect your privacy and monitor your safety, the important question always arise that are IP cameras really secure? Security is an important factor taken into consideration nowadays when a consumer is using products just for the sole reason of video monitoring and surveillance. IP cameras with different capabilities are employed either in private households or on a larger scale in industries and corporations. It is important to understand that everything which is connected to the network is prone to fall under the wrong hands of cyber criminals who can infiltrate into online systems if security is not too

strong. IP cameras like other network devices come with their fair share of security they can provide. An IP camera, be it wired or wireless wifi camera, connects to servers online. It is assigned its own IP address. If the cameras are not set up correctly, it is very easy for it to fall under the wrong hands. Weak login credentials is very simple to crack, it can be done simply with a key logger. Moreover a known IP address can be used to login to a camera's administration user interface online. There even exists several search engines that exposes vulnerable camera devices being connected online. Therefore it is important to question an IP camera's security and the measures taken in order to ensure maximum security. A lot of IP cameras are designed with features that support two factor authentication. This ensures an extra layer of security. Hackers who are able to find out passwords will not be able to get through two factor authentication. The camera which is being used in this project does not support such features. If the camera supports WPA2 protocol, it will ensure that the data being transferred over the network is done securely through the wireless router.

3 Practical Implementation of Live Video Streaming

3.1 Raspberry Pi

The Raspberry Pi used in this project is the Pi 3B+. It is preinstalled with a wifi chip for wireless network connectivity. It is powered by a quad core 64 -bit 1.4 GHz processor. It has a faster Ethernet and PoE (Power over Ethernet) support. The Pi was chosen as to serve as the main IoT kit for the live video surveillance system. Its main objective is to run a program using a programmed script that shows live video feed on a display. It is run under Debian Raspbian, Linux OS therefore all the packages and different software libraries being affiliated with this project are purely Linux based. After being set up and booted for the first time, the SSH (Secure Shell) and VNC (Virtual Network Computing) were enabled in the Pi configuration settings. This allows the Pi to be controlled remotely and be used as a headless system eradicating the use of external peripherals. In order to use VNC, VNC viewer was installed on Windows. The IP address of the Pi also has to be determined so VNC can connect to the Pi on Windows. Typing *hostname –I* in the terminal window of the Pi reveals the IP address the Pi is assigned. The Pi is now ready to be used remotely for the next steps.

3.2 TP-Link NC450

The IP camera being used in this project is a wireless wifi camera from TP-Link Technologies. It is a PTZ (Pan Tilt Zoom) camera that holds night vision capabilities along with sound and motion detection. Camera can be easily mounted on the roof or walls. It can rotate vertically up to 150 degrees and rotate 360 degrees horizontally. The camera is installed with a 1 Megapixel CMOS sensor which is used for imaging and recording at a maximum resolution of 1280x720p at max 30 fps. The camera also supports 2 way-audio which mean the user who is using camera for surveillance can communicate through the camera's microphone with others in the area being monitored. Camera has a micro SD slot which can store up till 32 GB of recorded and saved content. TP-Link has its own cloud application which allows the camera to be controlled remotely via a smart phone. The camera can only connect to 2.4 GHz wireless networks. TP-Link NC450 uses H.264 video compression standard and claims to have ONVIF compliance. Although upon confirming whether it is an ONVIF conformant device, the device was looked up on the official ONVIF conformant search and there were no results showing TP-Link products being conformant. Therefore it is not truly ONVIF conformant. This camera which claims to support ONVIF has ONVIF port number 3702. The RTSP port for this camera is port 554. The Bonjour name for this camera is NC450 2.0-1ab651. Figure 9 below shows the camera used in this project.

Figure 9: TP-Link NC450 wifi camera used in this project

3.3 Motion Eye OS

Motion Eye OS is software by Linux distributions. It is solely used as a video surveillance system for Linux based single board computers, most particularly Raspberry Pis.

As the name suggests it is itself an OS that replaces the current Linux OS on the Pi. It has an online web based interface that is operational in any browser. It has support for most IP cameras including TP-Link NC450. Motion Eye is capable of motion detection and still images. Alerts can be enabled and the media content can be viewed via FTP server. This method of viewing live video feed was also attempted using Motion Eye OS. The steps leading to the results are covered shortly.

3.4 Scripting and RTSP Stream Testing

The main objective of this project is to achieve the desired live streamed video feed using a programmed script that the Raspberry Pi will run. For the programming aspect, Python is chosen to be the most suitable environment to work in. The first step is to ensure that Python 3 is installed in the terminal window and it is updated to latest version. OpenCV, an open source software library, is used for this video surveillance system. The API for python is downloaded and installed from the documentation tab in OpenCV website. Thonny IDE environment is used for programming and writing the script. After installing OpenCV and with the help of imported libraries, the scripting for this program is under operation. RTSP is the key role included in the script. Upon researching the camera streaming info, the RTSP stream link for TP-Link NC450 which works for RTSP streaming is *rtsp://<username:password@<camera ip>:554/h264_hd.sdp*. TP-Link confirmed that the port number 554 will allow port forwarding for this camera. After modifying the imported library, it is ready to run. Listing 4 below displays the main code for streaming live video feed.

```
import cv2

cap = cv2.VideoCapture('rtsp://admin:password@192.168.1.108:554/h264_hd.sdp')

while(cap.isOpened()):
    ret, frame = cap.read()
    if ret==True:
        cv2.imshow('frame',frame)
        if cv2.waitKey(1)  == 27:
            break
    else:
        break
```

Listing 4: Main program code for displaying live video stream

The wait key line in the code is included for termination of the session. The number 27 corresponds to the ASCII value for <ESC>. When ESC is pressed, the video session will terminate instantly. However, after testing the RTSP streams multiple times, there

are still errors in generating the live stream. The main reason for this is because the H.264 format is not being encoded and decoded. This is where FFmpeg comes into play. It has to be installed on the Pi terminal and the script was finally able to recognize the camera's media format. Another method of displaying live video feed in this program was incorporated and that was through a RTSP client such as VLC media player. For that purpose, VLC had to be installed on terminal. After creating a code, the program was successful in running and displaying the live video feed in VLC. Listing 5 shows the program code for running live feed on VLC player.

```
import numpy as np
import cv2
import vlc

player =
vlc.MediaPlayer('rtsp://admin:password@192.168.1.108:554/h264_hd.sdp')
player.play()
```

Listing 5: Program code for displaying video feed in VLC

4 Results

4.1 Displaying Live Video Feed Using Python Script

The results obtained were surprisingly desirable. The main program code for generating RTSP stream was successful after FFmpeg was installed in the Pi. The program automatically opens the live video stream in a separate window frame after being run. It takes a couple of seconds for camera to initialize before the stream becomes smoother and frames become stable. For the first code, the video generated stream has a video latency of approximately less than or equal to 1 second. An important observation regarding video latency was made. When the display is viewed on the laptop display via VNC, the latency is greater than 1 second and image quality is low. The frames drop in comparison to when the same stream is being viewed in a HDMI display. Latency is much lower compared to when video feed is viewed in a VNC connected laptop display. The figure 10 below shows the obtained result after running the main code.

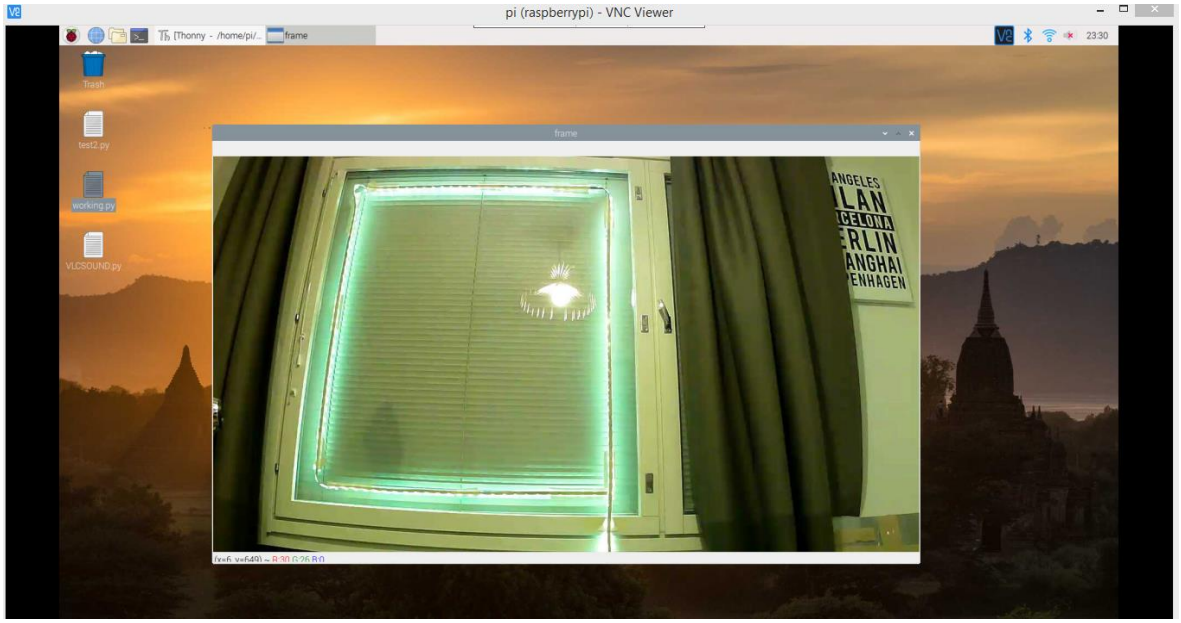


Figure 10: Snap taken of live video feed display using RTSP stream link

Figure 11 below shows the live video feed being streamed on VLC when the second code is run.

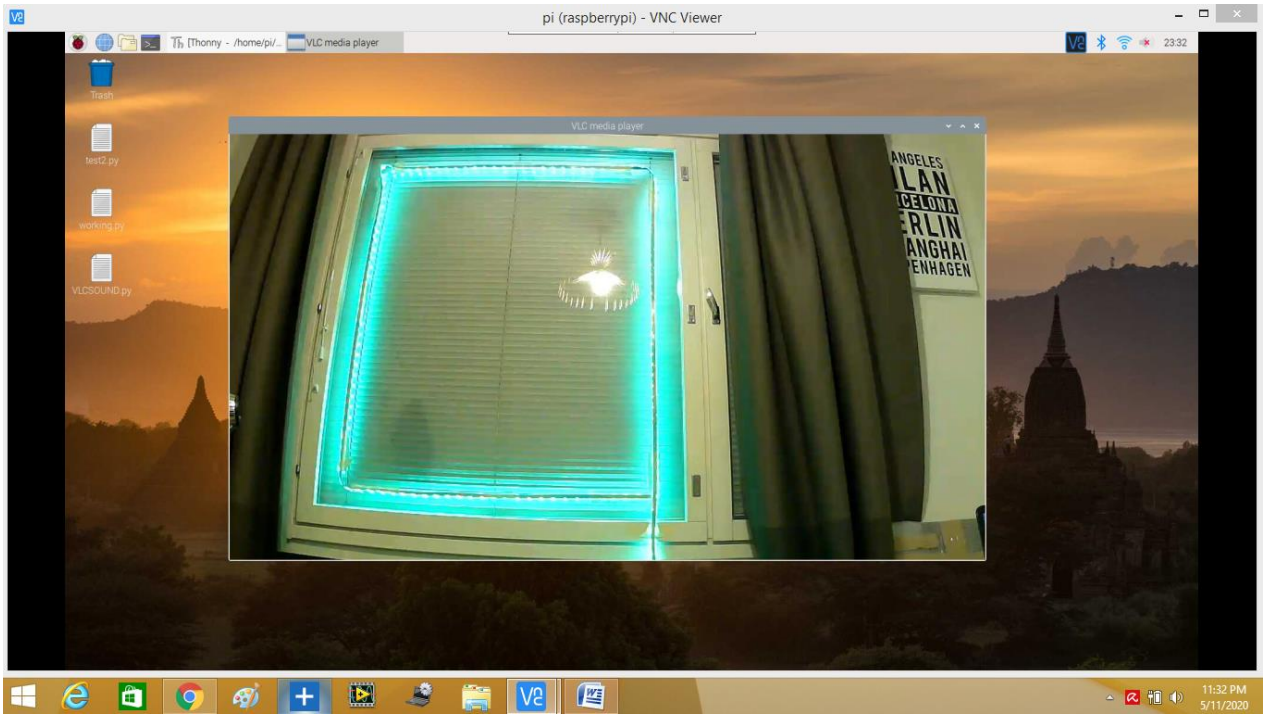


Figure 11: Image snapped of second code showing video feed on VLC

In comparison to the previous results, the latency is greater when video is streamed in VLC player than the previous code. Video latency is approximately about 2 seconds and image quality is low too. Frames drop even lower in comparison. There is same

delay in audio too in VLC. However, when video feed from latter script is viewed through HDMI display, it is much smoother than video feed being viewed in the laptop via VNC.

4.2 Displaying Live Video Feed on Motion Eye OS

Video feed display is also achieved using Motion Eye OS for Raspberry Pi. For that purpose, Motion Eye OS is installed on the SD card to replace Raspbian. Upon installation, the Pi running as a video surveillance system has to be connected to a wired network for first boot. The Pi displays the ip it has been assigned during the first boot process. The IP address is entered in a web browser and login credentials have to be entered. Motion Eye successfully opens on the browser and the camera has to be added manually. When the RTSP stream link and the camera login credentials are entered, the camera is detected and starts live streaming. Camera streaming settings and other preferences can be changed in the settings tab. When the resolution for this camera is set at 1280x720p and frame rate at 30 fps, resultant video latency is less than a second. Image quality is surprisingly good. The live video stream in Motion Eye OS can be seen in figure 12 below.

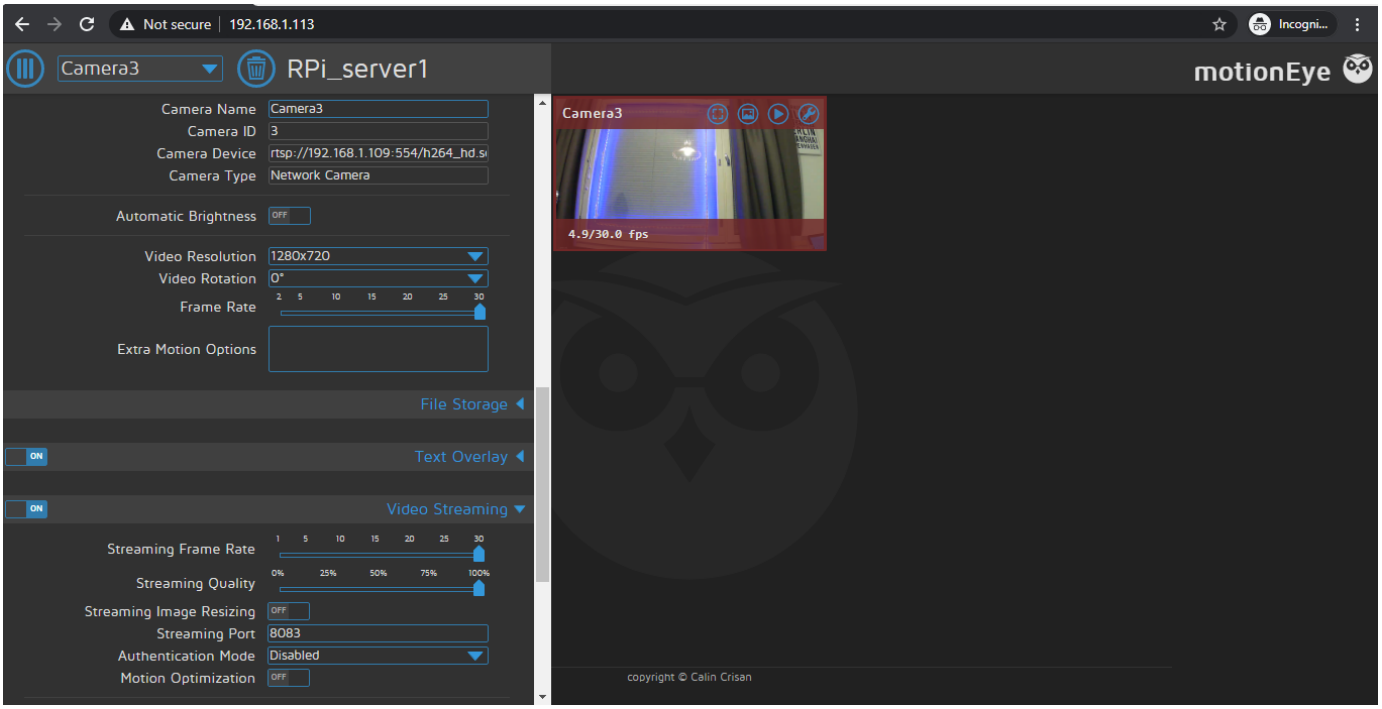


Figure 12: A screenshot of MotionEye showing live stream on right column

5 Conclusion

The main aim of this project was to create a program for Raspberry Pi that will show live video streaming with the help of an IP camera on a display. Apart from the main objective, another analysis was to determine whether the less powerful Pi can manage video streaming capabilities without too much video latency. The goal was to create a system that allows users to monitor environment or specific areas they wish to do so remotely using an IP camera. Practical implementation of such a system can range from surveillance of infants or pets at a private household to larger commercial businesses and companies. The fully functional video surveillance system was achieved as planned in several ways. The core of this project revolves around a Pi 3B+ and the TP-Link NC450 wifi camera. Communication between the two is established wirelessly using RTSP streaming protocol. The camera using a H.264 video codec standard has to be encoded and decoded in order for the media data to be transmitted. FFmpeg along with OpenCV allows the media format to be compressed. Once the camera description file has right encodings, RTSP client can issue SETUP request to the camera and later on other method requests to establish a real-time session between the two. The media content is then transmitted in RTP/RTCP data packets.

The Pi plays a role in acting as an IoT system that runs the program written in python. The results obtained were as expected and revealed that the Pi can be used as a cost effective easy to use video surveillance system. With the final program, video latency resulted to be less than 1 second which is quite acceptable in terms of efficiency and reliability of this system. When a HDMI display is employed for viewing the live footage, latency is further deduced. It can be concluded that the existent video latency can be a result of the Pi's performance capabilities and processing power. However, the operational Pi in conjunction with the TP-Link NC450 can raise doubts when it comes to security. The camera which does not support two- factor authentication can be prone to cyber breaches. The overall reliability of this particular project can be questioned in terms of security. These questions can perhaps be solved with more secure and better replacements to the NC450.

References

1. Understanding Linux [online]
   URL: https://en.wikipedia.org/wiki/Linux
   Accessed on November 14, 2019

2. Comparing the three leading Oss [online]
   URL: https://www.zdnet.com/article/windows-mac-or-linux-we-compare-the-pros-and-cons-of-these-computing-platforms/
   Accessed on November 28, 2019

3. The three Oss Info graphics [online]
   URL: https://www.educba.com/linux-vs-mac-vs-windows/
   Accessed on December 7, 2019

4. What is Raspbian? [online]
   URL: https://en.wikipedia.org/wiki/Raspbian
   Accessed on January 28, 2020

5. Describing V4L [online]
   URL: https://en.wikipedia.org/wiki/Video4Linux
   Accessed on February 14, 2020

6. Python language [online]
   URL: https://en.wikipedia.org/wiki/Python_(programming_language)
   Accessed on March 13, 2020

7. OpenCV [online]
   URL: https://opencv.org/
   Accessed on March 19, 2020

8. Ffmpeg [online]
   URL: https://en.wikipedia.org/wiki/FFmpeg
   Accessd on April 4, 2020

9. MPEG-2 [online]
URL: https://en.wikipedia.org/wiki/MPEG-2
Accessed on April 15, 2020

10. H.263 [online]
URL: https://en.wikipedia.org/wiki/H.263
Accessed on April 20, 2020

11. H.264 [online]
URL: https://en.wikipedia.org/wiki/Advanced_Video_Coding
Accessed on April 26, 2020

12. H.265 [online]
URL: https://en.wikipedia.org/wiki/High_Efficiency_Video_Coding
Acessed on April 27, 2020

13. RTSP[online]
URL: https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol
Accessed on May 3, 2020

14. RTSP memo [online]
URL: https://tools.ietf.org/html/rfc2326
Accessed on May 3, 2020

15. Ffmpeg with V4L2 [online]
URL: https://ffmpeg.org/ffmpeg-devices.html#video4linux2_002c-v4l2
Accessed on 6 May, 2020

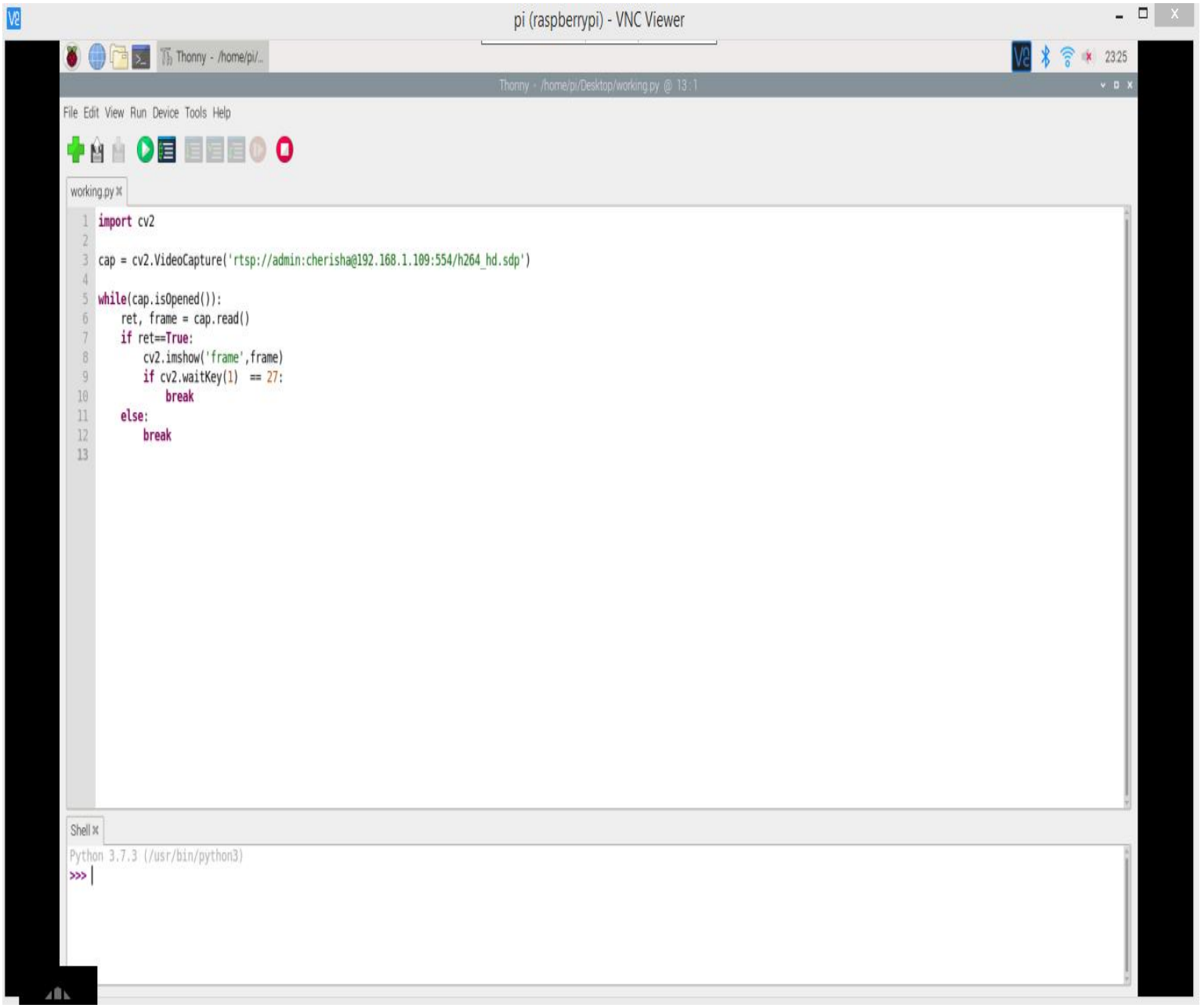
16. H.263/H.264 devices [online]
URL:
https://tietokonekauppa.fi/products/1147532/Viihde_eletroniikka/Puhelimet/IP_puhelimet/YEA_VP530/Yealink_Video_phone_7_SIP_H_264_H_263_PoE
Acessed on 7 May, 2020

17. ONVIF [online]
URL: https://www.onvif.org/
Acessed on May 7, 2020

18. SIP [online]
URL: https://en.wikipedia.org/wiki/Session_Initiation_Protocol
Accessed on May 7, 2020

19. What is Bonjour? [online]
URL: https://searchmobilecomputing.techtarget.com/definition/Apple-Bonjour
Accessed on 9 May, 2020

First Code


pi (raspberrypi) - VNC Viewer
Thonny - /home/pi/...
23:25
Thonny - /home/pi/Desktop/working.py @ 13 : 1
File Edit View Run Device Tools Help
working.py
import cv2
cap = cv2.VideoCapture('rtsp://admin:cherisha@192.168.1.109:554/h264_hd.sdp')
while(cap.isOpened()):
ret, frame = cap.read()
if ret==True:
cv2.imshow('frame',frame)
if cv2.waitKey(1) == 27:
break
else:
break
Shell
Python 3.7.3 (/usr/bin/python3)
>>>

Second Script (VLC)