e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

DESIGN AND DEVELOPMENT OF AN OPTIMIZED P2P BASED ONLINE VIDEO STREAMING ON RASPBERRY PI 3

Imran Khan*1, Nasir Ahmad*2

*1,2Computer Systems Engineering, University Of Engineering & Technology, Peshawar, Khyber Pakhtunkhwa, Pakistan.

ABSTRACT

Streaming technologies are becoming more and more popular with the inventions of new internet technologies. Streaming technology provides multimedia data to clients on the Internet. Transmission of multimedia content is achieved in real-time over the internet using current Streaming Technologies. Few of the existing video streaming technologies include CDN, VoD, Web based distribution, P2P systems, and Live streaming. Currently peer to peer live streaming is running or PCs or mobile devices which is slow as well as energy consuming. Energy is a big issue because we must keep on the system all the time. It's ideal to have a dedicated machine for your Live Streaming, so you can use it anywhere any time. But it is energy intensive to leave a full rig powered up and online that often. For solving these issues, we are entering the Raspberry PI. Raspberry PI is an inexpensive hardware platform that generates little heat, draws little power, and can run silently 24 hours a day without having to think about it. By implementing peer to peer live streaming on Raspberry PI we will have a dedicated small chip which can be integrated in devices easily. We can implement this technology in cars and many other places where we can't use PCs and keep them on all the time, then Raspberry PI is a better option because of its low cost and small size. In this paper P2P live Streaming will be implemented on Raspberry PI 3. Peer-to-peer (P2P) live video streaming systems distribute live video streams among very large set of users. P2P technology makes possible the direct data exchange and interaction among participating peers by eliminating a central server, which forms the basis for decentralized distribution environment and brings robustness in case of system failure. All nodes contribute to the resources in a P2P system. Due to vast features of Raspberry PI 3 like speed, we will implement live streaming on Raspberry PI to take advantages of P2P with an energy efficient system. Raspberry PI 3 provides Quad-core 64-bit ARM 1.2 GHz CPU, GPU 400MHz, Video Core IV multimedia, 1GB RAM with 40 GPIO ports which will make it easy to capture videos.

Keywords: P2P, FFMPEG, H264, Raspberry Pi 3, ARM.

I. INTRODUCTION

Open-source multimedia software such as FFMPEG and P2P streaming libraries like GRAPES will be utilized to make optimized system for diverse and heterogeneous networking conditions of undeveloped countries. The development of application-level software will be the main contribution of this paper which when installed will incorporate these libraries into a consolidated package which will allow users to access real time multimedia content. This multimedia content is captured by a camera which is interfaced with Raspberry PI. FFMPEG will De mux the raw file of multimedia content and then encode audio and video file separately and then raspberry PI will transmit these multimedia content using P2P streaming [4] framework (using GRAPES library). In P2P system, each peer act as a "server" and "client" both at the same time. Multimedia stream to be transmitted is divided into small pieces of chunks which are then send to partner peers frame by frame. This process of splitting encoded media stream into chunks is known as "chunkisation". At the receiver raspberry PI the received audio and video file is decoded and multiplexed to a single file by FFMPEG. Any user will be able to receive and play the contents over internet. The exponential growth in internet traffic is due to recent emergence of high speed networks. The amount of media related traffic has specially increased with the advancement in modern digital multimedia capture and their production. According to measurements, in 2014 the video traffic is the biggest internet traffic generator which constitute of about more than 57% of the internet traffic [6]. This trend greatly challenges the content providers as well as internet service providers to provide the users with best possible experience of video streaming. The greatest problem associated with this traditional client server systems to facilitate large group of users is that in these systems the delivery of the content rely on dedicated infrastructure i.e. server, due to which the server load increases which in turns

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

decreases the speed and the quality of the server to the end users. Peer to peer system were introduced due to this disadvantage of client server systems. The main objective of P2P streaming systems is to utilize the bandwidth of the clients in the network thereby decreasing the load on the server and hence improving the overall performance. Peer to peer computing or networking is a distributed application architecture that partition tasks and workloads between participating peers. Peers are equally privileged, equipotent participants in the application. These form a peer-to-peer networks of nodes.

In peer to peer system, each node in the network acts both as a client and the server. There is no centralized server. Here, the content delivery does not rely on the single server rather the upload capacity of each peer on the network is taken advantage of and used for video transmission which drastically decreases the load on the server thus making the system more scalable and resulting in better performance. The low cost incurred in the deployment of peer to peer systems where a wellequipped delivery infrastructure is not needed makes it especially appealing to the content providers. Another advantage of P2P system which contributes to the popularity is its self-scaling property i.e. each time a peer consumes the resources of the network by streaming, its upload capacity is also used to add to the overall system resources [7].

II. METHODOLOGY

To develop scalable multimedia streaming system based on P2P, open source libraries like FFMPEG and GRAPES will be used to achieve the functionalities.

Video/Audio Source

This module will be responsible for preprocessing operations on the input of the system. To perform these operations, the module will need information like max chunk size, number of frames in payload, streamer IP port and audio/video encoder parameters. The resulting AV chunk is then sent to streaming engine via UDP stream.

Streaming Engine

This module will be responsible for retrieving, storing and sharing of the media chunks with the partner peers. It will also be responsible for sending a copy of each chunk that it receives to the media player. The streaming engine needs parameters like chunk buffer size, topology configuration and sent/received message. The streaming engine exists on both sender and receiver ends and perform differently in each case. The streaming engine on the sender end retrieves the AV chunk from the AV source while the same streaming engine on the receiver end receive chunks from the peers and send those to the media player on the channel specified by the player.

Media Player

Subheading should be 10pt Times new Roman, Once the streaming engine has received enough media chunks, it will assemble those chunks into a media file and delivers it to the player. The player will be responsible for playback of this media file. The player receives the AV stream via UDP stream. It also needs max chunk size, streamer IP port and channel configuration. A block diagram of Peer-to-Peer Modules is shown below in Figure 1.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

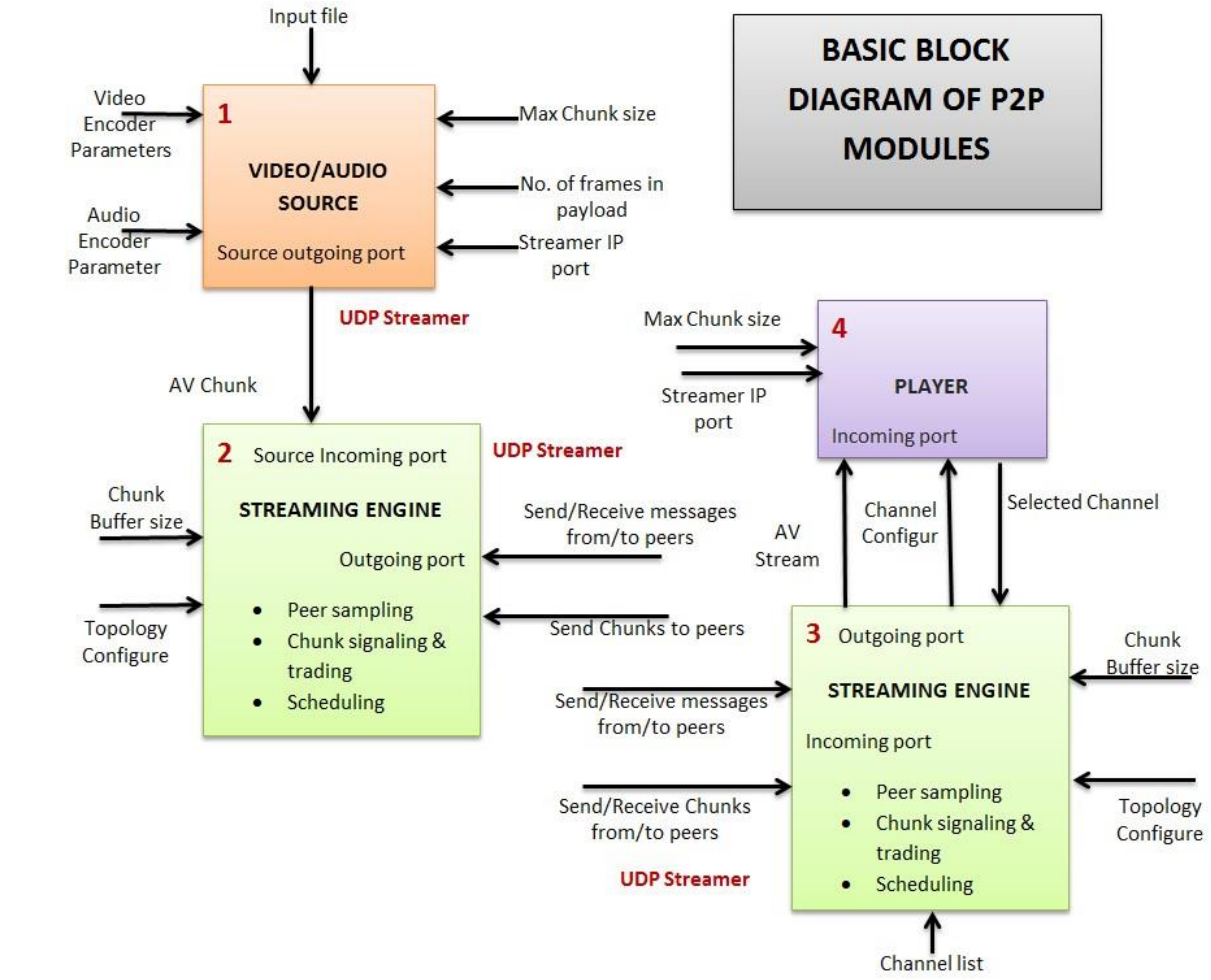

Figure 1: Block Diagram of P2P Modules

III. VIDEO/AUDIO SOURCE

Model and Material which are used is presented in this section. Table and model should be in prescribed format.

De Multiplexer

De Multiplexer takes single multiplexed stream of media (video and audio) and separate the streams which were combined into one. Audio and Video streams are separated from each other and forwarded for the further processing. The process where we separate the audio and [9] video from into the individual streams is called de-multiplexing. The AV source module takes the video file as an input. Initially, the input file is multiplexed. Therefore, a de-multiplexer component is placed in order to divide the input into video and audio streams separately. The libavformat library provides a generic framework for multiplexing and demultiplexing (muxing and demuxing) audio, video and subtitle streams. It encompasses multiple muxers and demuxers for multimedia container formats.

Video/Audio Decoder

After DE multiplexing the input file, the audio and video streams are decoded using the decoders. The decoders decodes the video stream into YUV frames with defined width and height while the audio stream into RAW PCM along with specific sample rate. There are wide range of decoders which are used by the developers. Some of them are AV_CODEC_ID_MP2, AV_CODEC_ID_MP3, and AV_CODEC_ID_H263 etc. Video decoder initializes queues for video decoder's input and output. Inputs are taken from demux_live_queue, outputs in video_decoder_2_video_ encoder queue. Audio decoder initializes queues for audio decoder's input and output. Inputs take from demux_2_audio_decoder queue, outputs in audio_decoder_2_audio_encoder queue. For the decoding purpose it is necessary to set few of the properties like sample rate (rate at which carrying of audio

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

samples takes place), sample format (number of bits for each sample), channel layout (channel parameter to the input data), frame rate per second (number of frames per second in a video), width and height of the video (frames width and height). Then the process of the [8] decoding involves initialization of ffmpeg using av_register_ all() function. Then opening file using the av_open_input_file(include parameters) and getting the information using avformat_find_stream_info(p1, p2). Then finding the video and audio streams and decoders for them. Find video stream and open it. Searching for audio stream is similar to video stream. Just use CODEC_TYPE_AUDIO instead of CODEC_TYPE_VIDEO. Now getting information

about streams like resolution, length and fps. Decode the information, process frames and close file.

Video/Audio Encoder

Encoding is the process of the converting data into particular format which can be later on transmitted over the network in efficient manner in terms of performance and storage of the data. In multimedia encoding of digital media represents the compressions of the data into small and effective streaming media coding format. The process of encoding is proven efficient in terms of bandwidth usage while transmission of streams. The tools which encodes the media streams are called encoders. [7] The video encoder takes the YUW frames and encodes those using the H.264 encoding scheme to allow scalable video transmission. The inputs required by the video encoder includes video codec, video resolution and video bitrate. Similarly, the audio encoder takes RAW PCM generated by decoder and encodes those by using mp3Lame.This component requires audio codec, audio sample rate and audio bit rate as its input parameters. The functionality of both these components is supported by FFMPEG library.

Payload Header

Payload is the actual data sent and header identifies the source and destination of the packet. Once the audio and video streams are encoded, their respective payload headers are added to them. These headers helps in the muxing of these separate audio video streams into a single video file at receiver's end. The payload header has all the necessary information of the streams. The number of frames in the payload needs to be specified.

Chunkizer

Before sending to partner peers, video/audio streams are divided into small pieces of chunks. The process of splitting encoded media into small chunks is referred to as

"Chunkisation". The resulting frames from the payload header are divided into chunks of predefined size using the GRAPES chunkizer. After dividing the streams into chunks, these are sent to the streaming engine as UDP data. A block diagram of Video/Audio source module is shown below in figure 2.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

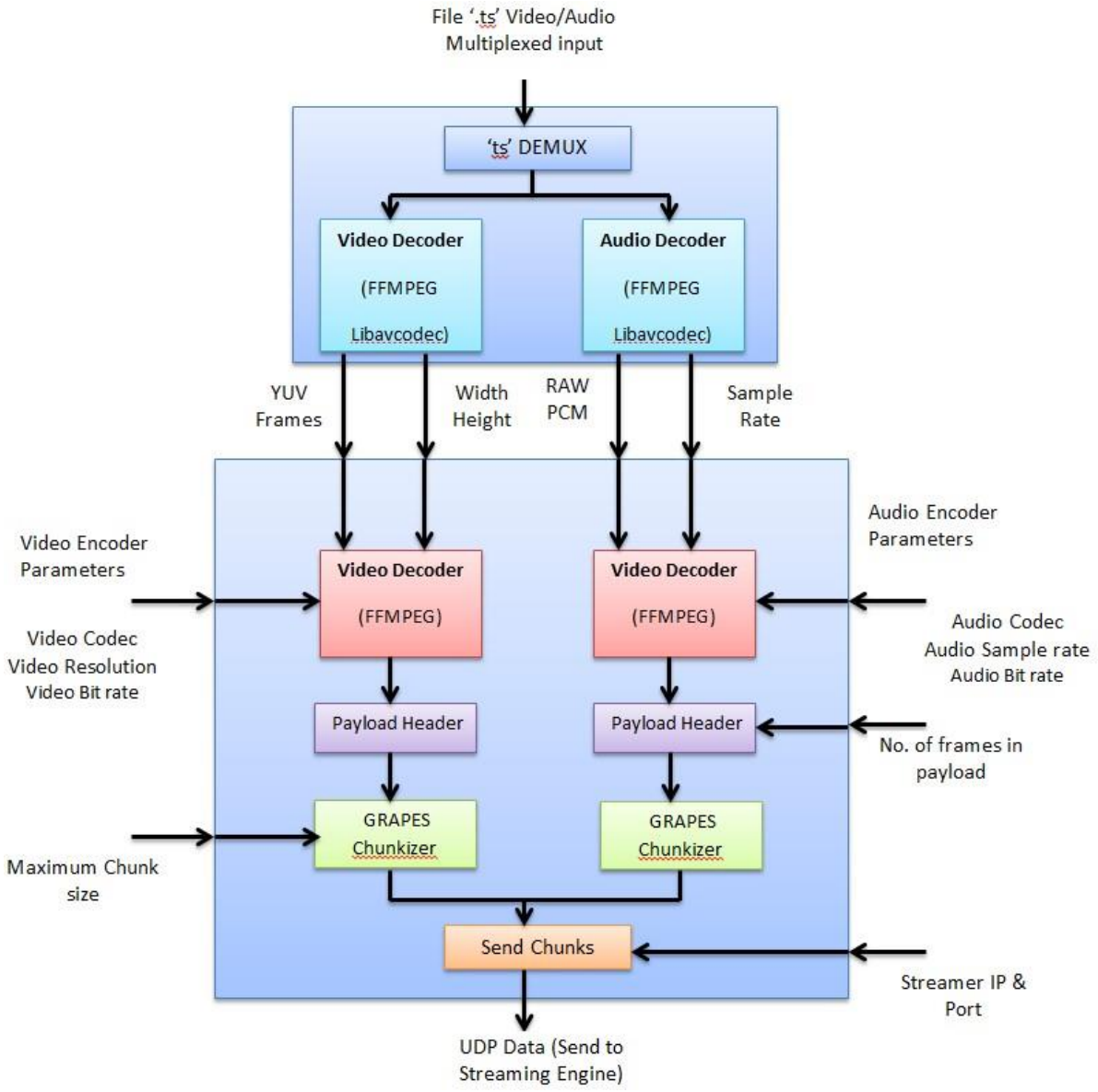

Figure 2: Block Diagram of Video/Audio Source

Streaming Engine (VA Source Connection)

Messages containing information about other peers are received in Streaming Engine.

Peer Sampler

The peer sampler consists of different configuration, and has to give information regarding adding new peers or removing existing peers from the engine.

Chunk Signaling

The second part of the message received is regarding chunk signaling. The Chunk Signaling mechanism is responsible for requesting new chunks IDs, offering new chunks IDs and accepting the chunk IDs. It also requests ad sends buffer maps to other peers. The message contains information of the peers currently available i.e. their Chunk ID, buffer map and chink buffer size.

Scheduling

The existing peers needs some kind of order in which they must receive or send chunks i.e. a form of an organized system in order to ensure the complete and reliable information in the chunks. It also helps in reducing over-head. Whenever a new peer joins the network this module updates the existing list of peers. This system comes after the scheduling section, and after this the peer information table gets updated. This module

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

specifically deals with the sending of chunks to other peers. As can be seen in figure 3, in order to store the chunks, a buffer stores the chunks in order. The Add Chunk Buffer module is responsible for receiving the chunks from the Audio/Video Source Module through UDP. These chunks are then stored in the Chunk Buffer for further processing, which forwards each chunk to Scheduling module. Block Diagram Of Streaming Engine is shown below in figure 3.

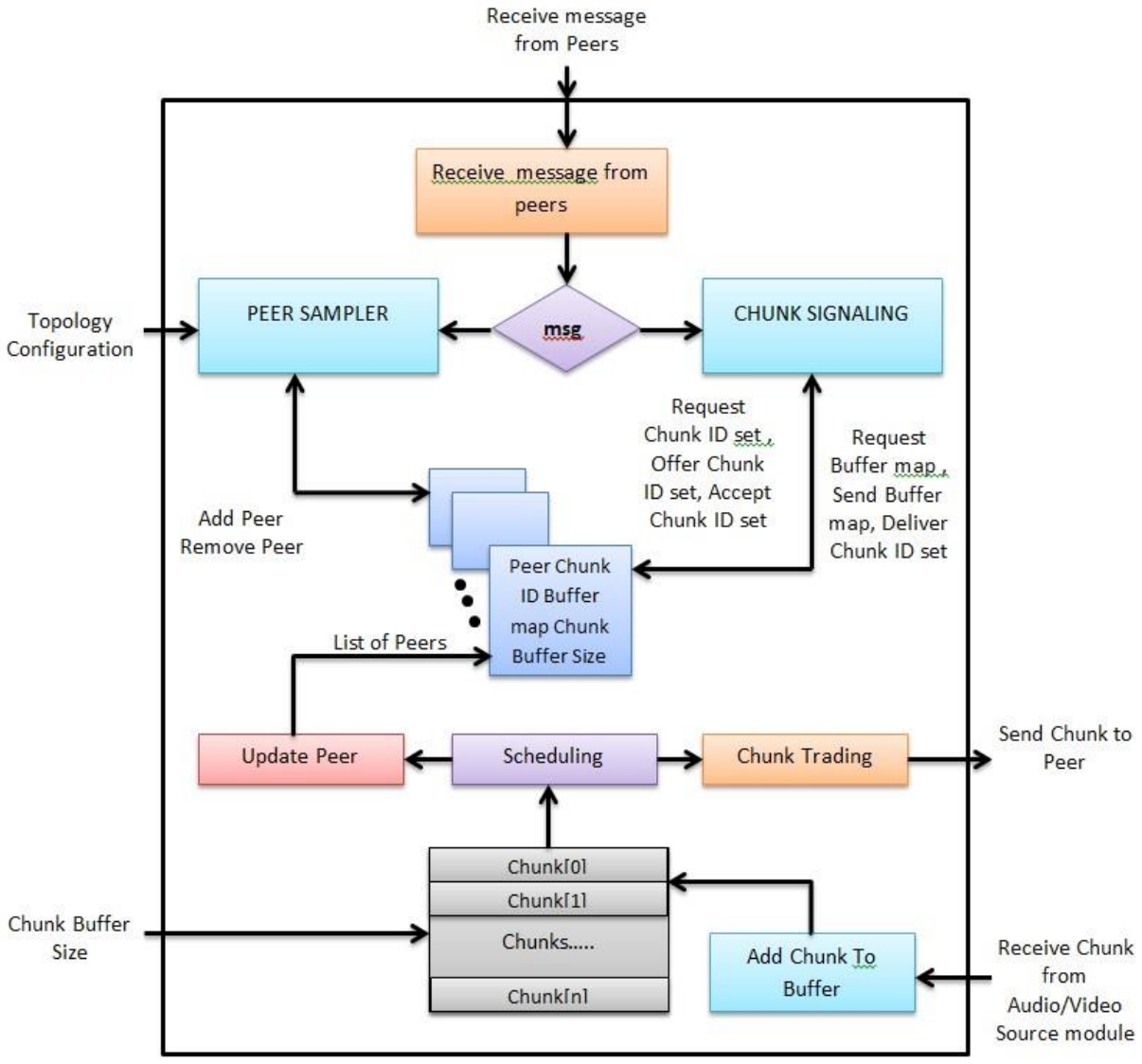

Figure 3: Block Diagram of Streaming Engine

IMPLEMENTATION ON RASPBERRY PI 3

Till now there is no dedicated system developed for P2P based live Streaming. So for now P2P live streaming is running on PCs or mobile devices which is slow as well as energy consuming. Energy is a big issue because we have to keep on the system all the time. It's ideal to have a dedicated machine for your Live Streaming, so you can use it anywhere any time. But it is energy intensive to leave a full rig powered up and online that often. For solving these issues we are entering the Raspberry PI. Raspberry PI is an inexpensive hardware platform that generates little heat, draws little power, and can run silently 24 hours a day. By implementing P2P live streaming on Raspberry PI we will have a dedicated small chip which can be integrated in devices easily. For implementing P2P based live streaming on raspberry PI 3 we will need some Dependencies. These are packages required for compiling. First of all you need to make a folder in your HOME directory which will contain all the source code and binaries. mkdir ffmpeg_sources. This part is to install some necessary third-party libraries. Each part will contain step to install these libraries. These are the special command to install these libraries on

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:11/November-2022 Impact Factor- 6.752 www.irjmets.com

32-bit architecture. An assembler which is used by some libraries, NASM x86-64 assembler designed for portability and modularity. It supports a wide range of object file formats in Win32 and Win64. This library work as an assembler and disassembler for the x86 architecture. It can be used to write 16 bit, 32 bit and 64-bit programs. It is a full rewrite of NASM.it can generally use with interchangeably NASM and supports the x86 and x64 architectures. it is a free and open-source codec library and a command-line utility which is developed by VideoLAN for encoding video streams into the H.264/MPEG-4 or MP4 AVCformat. It is a software package encoder that converts audio into MP3 file format.it is the world world first free open source library for encoding Audio streams. SDL is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware. It is used by video playback software, emulators, and popular games. SDL officially supports Windows, Mac OS X, Linux, iOS, and Android. It also support other open source platform. SDL is written in C, works natively with C++. Before compiling SDL on Raspbian or Ubuntu mate you to install these libraries. It is usually a very strong and free open source library used for executing numerous conversion operations on audio and video files. It is available for different OS like windows, Linux and also Mac. FFMPEG is a free of charge application paper which produces libraries and also programs intended to provide multimedia codecs. FFMPEG involves libavcodec, a great audio/video codec library used by a several different papers, libavformat,an audio/video jar mux and also de multiplex library, as well as the FFMPEG command line program intended for trans-coding multimedia files.

IV. RESULTS AND DISCUSSION

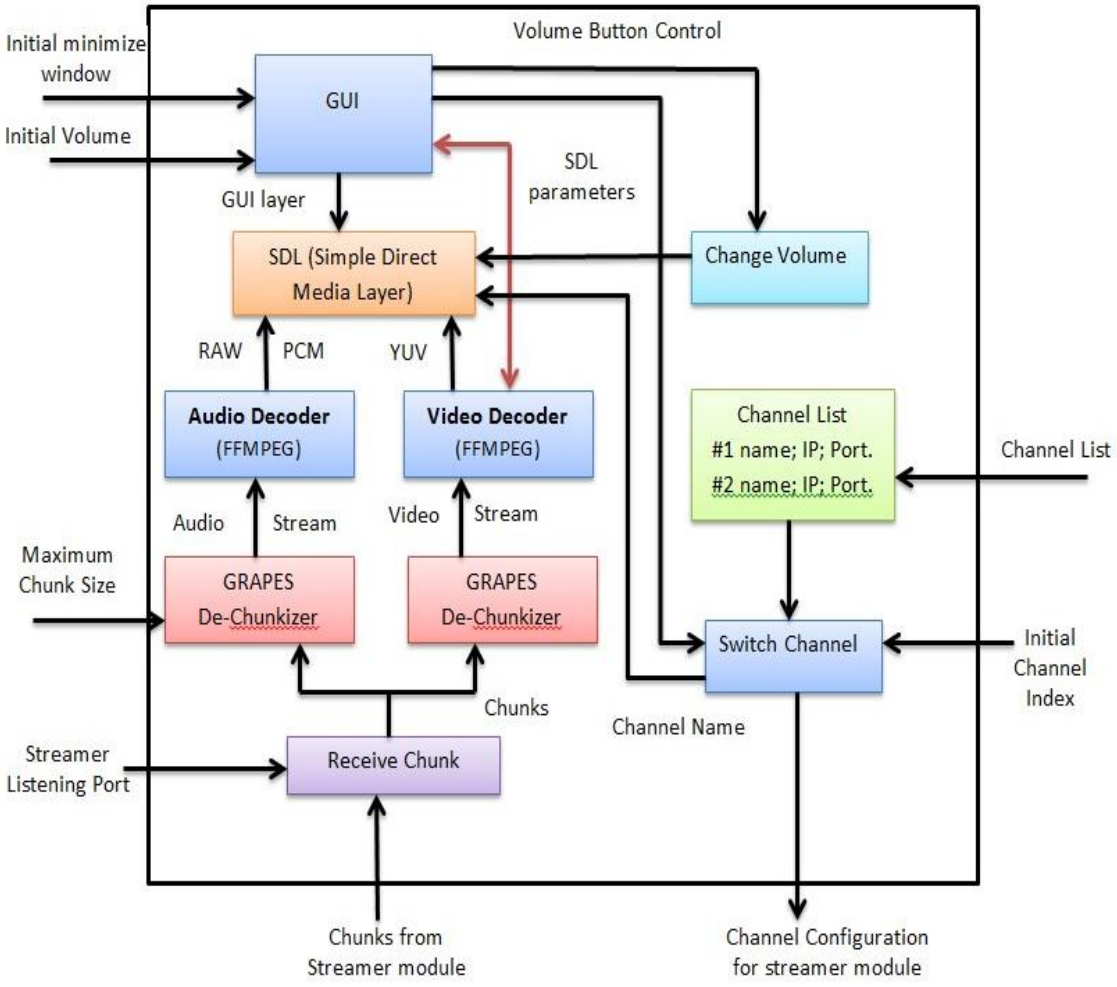

Figure 4: Block Diagram of Media Player

The demand of streaming technology is increasing day by day due to development of more and more internet technologies and facilities. Our paper Objective was to make an optimized P2P based multimedia streaming technology and implement it on raspberry PI which is credit card sized, low cost computer, capable of doing

almost anything just like a desktop computer or laptop when interfaced with a keyboard, mouse, PI mini camera and sound card. The basic purpose of a streaming technology is to provide access of multimedia streams over the internet to the users anytime anywhere. P2P streaming technology is a streaming technology that provide real time access of a multimedia streams to a large set of users. In P2P streaming system, central server is eliminated and multimedia stream or resources are exchanged between participating neighbouring peers directly without involving a server. Each peer or node in P2P streaming technology have equal responsibilities, they do not rely on a central server for managing resources rather than that all of the nodes equaly contributes and are self organized thus making the system optimized and brings robustness in case of system failure. The multimedia streams are transmitted in real time .The video stream to be transmitted is demuxed to separate the audio and video file then, each file is decoded separately. Audio and video files are divided into small pieces known as chunks and each chunk is transmitted by a raspberry PI. These chunks when received at the receiver raspberry PI are encoded, muxed and played immediately rather than waiting for the whole file to be received first. The paper was implemented on raspberry PI 3 in order to make it a low cost and low energy consuming streaming technology.

V. CONCLUSION

The paper can be extended by using a quad-copter or flying drone to make live video stream and transmits the multimedia stream in real time. Flying drone can be made by a raspberry PI because of its small size and high computational power. It has a great application in surveillance system and security. In today's busy world every common user wants to watch, listen multimedia stream without any kind of delay. With the passage of time, many technical innovations and advancements were made in computer technology. Today's modern computers are capable of doing critical tasks efficiently and accurately within very small time. In peer to peer streaming technology the data is transmitted in real time that is as soon as the data is transmitted in the form of chunks, these chunks are encoded, muxed and played simultaneously as they arrive at the receiver end without waiting for the other whole data to be arrived. Peer to peer streaming technology are also implemented on computers(x86). In our project this technology is implemented on Raspberry PI 3 which is inexpensive device with affordable price. Minimum cost of an ordinary laptop or a computer is upto 30,000 however raspberry PI which is a small sized complete programmable computer costs 6000. Because of low cost, raspberry PI can be purchased by common user. Raspberry PI consumes very small amount of energy as compare to Desktop Computers. The low price of the PI does not means less computational power. It comes with features like HD video playback and audio quality and support for 3D games. In short it is an all-rounder. Raspberry PI due to its small size and high computational power can frequently be used in robotics as embedded system, providing Artificial intelligence in Human Computer Interaction. It can be used to make any old television a smart TV and get complete functionalities which every traditional smart TV provides, or maybe even more. Fully functional tablet or mobile phone can be made from raspberry PI by attaching the screen and other required peripherals. A dedicated and lean torrent machine can be made with raspberry PI. A bit torrent server, dedicated specifically for torrent business.

VI. REFERENCES

[1] D. Quanfeng and L. Zhenghe, "File Sharing Strategy Based on WebRTC," in 2016 13th Web Information Systems and Applications Conference (WISA), 2016.

[2] X. Tian, C. Zhao, H. Liu and J. Xu, "Video On-Demand Service via Wireless Broadcasting," IEEE Transactions on Mobile Computing, vol. PP, pp. 1-1, 2016.

[3] R. Eskola and J. K. Nurminen, "Performance evaluation of WebRTC data channels," in 2015 IEEE Symposium on Computers and Communication (ISCC), 2015.

[4] Zhijie Shen et.al.," Peer-to-Peer Media Streaming: Insights and New Developments"

[5] Chin Yong Goh et.al.," A Comparative Study of Tree-based and Mesh-based Overlay P2P Media Streaming".

[6] D. B. Z. Despotovic and G. M. S. Bergamaschi, "Agents and Peer-to-Peer Computing".

[7] C. Kiraly, L. Abeni and R. L. Cigno, "Effects of P2P Streaming on Video Quality," in 2010 IEEE International Conference on Communications, 2010.

[8] V. Kalogeraki, D. Gunopulos and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in Proceedings of the eleventh international conference on Information and knowledge management, 2002.

[9] C. Hall and A. Carzaniga, "Uniform sampling for directed P2P networks," EuroPar 2009 Parallel Processing, pp. 511-522, 2009.

[10] E. Meshkova, J. Riihijärvi, M. Petrova and P. Mähönen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," Computer networks, vol. 52, pp. 2097-2128, 2018.

[11] X. Li and J. Wu, "Searching techniques in peer-to-peer networks," Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks, pp. 613-642, 2017.

[12] Jagdish A. Patel , Aringale Shubhangi , Shweta Joshi , Aarti Pawar , Namrata Bari5 “Raspberry PI based smart Home” International Journal of Engineering Science and Computing, March 2017