ARM 11 Based Real-Time Video Streaming Server Using RTSP

[1]Shahin S Rupalwala, [2]Ms. Nabila Shaikh
Student, Assistant professor
Department of Electronics & Communication, L.J Institute of Engineering & Technology, Ahmedabad, India

Abstract— **This paper represents a design of embedded video streaming server. It makes use of ARM11 processor & embedded Linux operating system as platform & develops a real-time streaming server through implementation of RTSP protocol. We designed a wireless video delivery system using Wi-Fi connection as connection mechanism for both real-time streaming and On-demand video playback. It ensures the delivery of media in proper manner by streaming the content from the server to the client. It uses the combination of web service technology & streaming media technology to successfully deliver media according to user requirements. The system performance is tested and relevant results are presented.**

Keywords— **Media management system,RTSP.Wi-Fi,Streaming media technology,Embedded system.**

I. INTRODUCTION

Network streaming applications has gain its importance with the development of Internet and multimedia technologies. Streaming media means the user needn't to download the data to the local machine. In recent times, most portable computers and smart phones have some type of inbuilt Wi-Fi network interface cards. Such development of Wi-Fi networks allows the user to convert the video distribution system to wireless networks. In this paper, we describe a real –time multimedia system based on embedded processor. The objective is to provide mobile users with Wi-Fi enabled devices, with an ability to watch live programs. The multimedia can be streamed in real-time to wireless handheld devices, for e.g., it is used to distribute the digitally captured On-Campus live events such as lectures, rules and regulations of university to new attendees and other events of general interest in real-time.

In addition to the primary objective of distributing live videos, the server also provides the Video-on-demand feature, where the user can start video when he wants, make pauses, go forward and back in the video at his desire.

Since the ability to access information at one's convenience is a desired feature of any multimedia distribution system, video-on-demand is of course the best in video streaming and is a dream of every user. In this paper, a new player based on ARM11 processor is developed, that uses RTSP protocol to stream video. This player is designed to solve some shortcomings of other media player, such as poor performance, delay, cost, inefficiency, complexity etc. Test plans has been carried out to check the performance of system and results states that it fully meets the needs of users and it proves that, it is an effective application for embedded system.

II. DEFINITION OF RELATED CONCEPTS

[A] ARM 11 processor(Raspberry pi)

The initial ARM11 core (ARM1136) was released to licensees in October 2002. In terms of instruction set, the ARM11 builds on the preceding ARM9 generation. It incorporates all ARM926EJ-S features and adds the ARMv6 instructions for media support (SIMD) and accelerating IRQ response.

Microarchitecture improvements in ARM11 cores include: SIMD instructions which can double MPEG-4 and audio digital signal processing algorithm speed. **C**ache is physically addressed, solving many cache aliasing problems and reducing context switch overhead. Unaligned and mixed-endian data access is supported. Reduced heat production and lower overheating risk. Redesigned pipeline, supporting faster clock speeds (target up to 1 GHz)

The arm chips have the general characteristics of the RISC systems; moreover some special advanced technologies are adopted in the processor by engineers to ensure the high performance with the minimized chip size and reduced power consumption. Raspberry pi board is used as the development platform in this paper.

The Raspberry Pi is based on the Broadcom BCM2835 system on a chip (SoC), which includes anARM1176JZF-S 700 MHz processor, Video Core IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded (Model B & Model B+) to 512 MB. The system has Secure Digital (SD) or Micro SD (Model B+) sockets for boot media and persistent storage. The Raspberry Pi chip, operating at 700 MHz by default, will not become hot enough to need a heat sink or special cooling. The SoC is stacked underneath the RAM chip, so only its edge is visible. Though the Model A does not have an 8P8C ("RJ45") Ethernet port, it can connect to a network by using an external user-supplied USB Ethernet or Wi-Fi adapter. On the model B the Ethernet port is provided by a built-in USB Ethernet adapter.

Generic USB keyboards and mice are compatible with the Raspberry Pi. The video controller is capable of the following video resolutions: 640 × 350 EGA; 640 × 480 VGA; 800 × 600 SVGA; 1024 × 768 XGA; 1280×720 720p HDTV; 1280 × 768 WXGA

[B] Streaming protocol

The Real-Time Streaming Protocol (RTSP) establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not typically deliver the continuous streams itself, although interleaving of the continuous media stream with the control stream is possible.
The protocol supports the following operations: Retrieval of media from media server, Invitation of a media server to a conference, Addition of media to an existing presentation.
RTSP has the following properties: Extendable, easy to parse, secure , transport-independent, multi-server capable, control of recording devices, Separation of stream control and conference initiation, Suitable for professional applications, Proxy and firewall friendly, HTTP-friendly etc.
Each presentation and media stream may be identified by an RTSP URL. RTSP is identified by an RTSP URL, which points to the media server handling that particular media stream and names the stream stored on that server. Several media streams can be located on different servers; for example, audio and video streams can be split across servers for load sharing. The description also enumerates which transport methods the server is capable of.
RTSP has some overlap in functionality with HTTP. It also may interact with HTTP in that the initial contact with streaming content is often to be made through a web page. The current protocol specification aims to allow different hand-off points between a web server and the media server implementing RTSP. It also enables On-demand delivery of real -time data.
Some of the RTSP commands commonly used are as follows:
OPTIONS
DESCRIBE
SETUP
PLAY
PAUSE
RECORD
TEARDOWN
An example of such a client request using OPTIONS is as follows (assuming port number of 554):
OPTIONSrtsp://198.164.81.55/sample.1RTSP/1.0
In response to the client's request, the server will response with ASCII based status code similar to HTTP. Examples of the status codes with their meanings are as follows:
200: OK
301: Redirection
405: Method not allowed
451: Parameter not understood
462: Destination unreachable

III. HARDWARE DESIGN

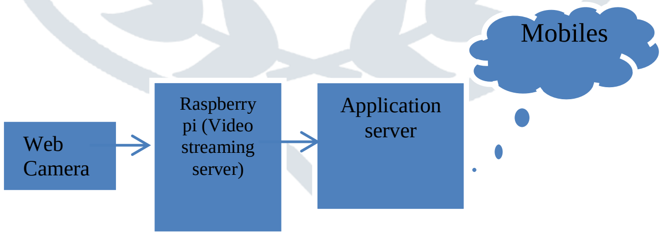

Figure 1. Block diagram of the hardware system design

Video streaming system consists of four blocks: Web Camera, Video streaming server, application server, and client side devices.
Web Camera: It is used to record videos to be stored in server. There is a plug with in ARM used as extension to connect the camera. This web camera continuously monitor the room and send the video.
Video streaming server: It is used for compression and processing of videos. Streaming Media server allows you to send media to clients across the internet through the streaming media protocol .Raspberry pi used as a development platform for streaming media server. It enables live streaming of video content from a single server to a large number of clients. Video streaming server is implemented and installed on a raspberry pi with ARM11 processor and linux as the operating system.
Application Server: It is used to send and receive multimedia. Through web server ,it forwards the data packets between computer network making connection through Ethernet cable on ARM11 Board(Raspberry pi). Using
Wi-Fi we can stream the video through mobile or PC.

IV. SOFTWARE DESIGN

Programming was done strictly within the C language. The system selected Linux operating system as software platform, use embedded Linux 2.6 kernel. The software controls the operation of the entire system, in structuring the camera, compression and processing of video , and transmitting it to clients using transport level and application protocols. There are mainly three function modules, that is, Video capture module, Video Compression module, and Video Streaming module. The flow chart system of software operation shown in Figure 2. [1]

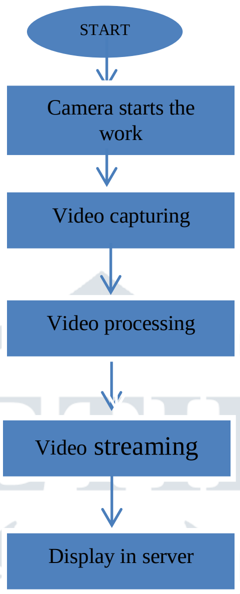

Figure 2. Flow chart of software operation [1]

Video capture module

Video Capture Based Video4Linux (referred to as "V4L") is a Linux kernel on the video device driver, which is for video equipment, application programming interface functions to provide a system. V4L USB camera using the programming on the need to use Linux system. Programming was done strictly within the C language.

Video Processing

The increasing demand to incorporate video data into telecommunications services, the corporate environment, the entertainment industry, and even at home has made digital video technology a necessity. A problem, however, is that still image and digital video data rates are very large, typically in the range of 150Mbits/sec. Data rates of this magnitude would consume a lot of the bandwidth, storage and computing resources **i**n the typical personal computer. For this reason, Video Compression standards have been developed to eliminate picture redundancy, allowing video information to be transmitted and stored in a compact and efficient manner.

Embedded Web Streaming Server

Once RGB web camera is connected through master USB interface to arm board make minicom-s settings in the terminal window, during the settings we run the application related shell script in terminal which will execute application in board resulting video streaming on web browser using RTSP protocol, entering a static IP address by user in any wireless device which is in local network can view the remote location. The server periodically obtain videos from camera through the private network, such videos are transmitted from camera to the server. They are accumulated as MPEG video temporally into the internal buffer of the server.

V. PERFORMANCE EVALUATION

A. System setup

The basic application of this project is to stream lectures in a college campus to prevent loss of classes by delivering video lectures at the scheduled timings. Streaming videos stored on server directly to Android can be achieved using ES file explorer. In ES File Explorer, slide over to the LAN tab, and press the '+' icon to specify your Video directory. For this method to work, however, you first need to make the videos stored on your server accessible across your network any device on your network (within the same Workgroup) can access your videos directory. Using the same Wi-Fi connection of server, mobile clients can watch videos stored on server. The basic information required to request media from server is to have an IP address of server.

B. Evaluation of client video reception performance

In order to better evaluate the performance of our proposed video streaming server, we established a realistic view to measure the performance with a real-time video stream and an Android mobile phone. A smart phone with Android OS is used as the client. The smart phone is integrated with Wi-Fi wireless interface.

We implemented a video streaming application which can request the video contents from the web server through RTSP URL. The commands are sent as message through a RTSP URL.

Based on this experiment, we checked out the delay in the transmission of data packets. There is minimal delay compared to other Video streaming techniques. The video quality of playback of video images is also good and control functions of the playback of video images are quite effective.

VI. CONCLUSION

This paper describes the implementation of a video network streaming system on the Raspberry Pi board, an embedded board based on ARM11 processor running on Linux operating system. The system can also be used for video-on-demand feature. We deal with delivering continuous live video streamed to a group of clients, using RTSP URL. This server effectively realizes the real-time transmission, playing and playback of video images with RTSP protocol. The paly, pause, stop and other functions of streaming media are implemented in this paper. This system is ideally suited for institutions and organizations to create their own media delivery system. At last we designed test plans to validate the performance of media player and the test results show that it completely satisfies the embedded system user’s demand and performs well.

REFERENCES

[1] Weitao Xu, Dongfeng Yaun, Jiali Xu, “Implementation of a network streaming media player based on ARM processor”, April 2012

[2] Dian Chu, Chun-hua Jiang, “The design and implementation of video surveillance system based on H.264, SIP, RTP/RTCP and RTSP’’, April 2013.

[3] Prof. Nitin Talhar, K.S Thakre, “Real-time and object-based video streaming techniques with applications to communication system”, ISCCC 2009.

[4] Hongyun Wang, Jiawei Li, “Design of an embedded streaming media server in video monitoring”,March 2013.

[5] Xiaoling Qiu, Happing liu , Deshi Li, Son Zhang, “Optimizing HTTP-based adaptive video streaming for wireless access networks”, March 2010.

[6] Meng Guo, Mostafa Ammar, “Scalable live video streaming to cooperative clients uses time shifting and video patching”, July 2004.

[7] Pradipta De, Srikant Sharma, Andrew Shuvalov, “WI Vision: A wireless video system for real-time distributon.

[8] Ying Li, “Content –aware distortion fair video streaming in congested networks” October 2009.

[9] Will hrudey, Ljiljana Trajcovic, “Streaming Video content over IEEE 802.16/Wimax broadband access”.

[10] Joohno choi, “Efficient video-on-demand streaming for broadband access networks”, January 2010

[11] Emrin Zerman, “Real-time 3D video Streaming: A mobile approach”, in press.

[12] Wen boyin, Xing ruonan,” Design of Video-on-demand system based on streaming media technology”, June 2013.

[13] Graeme stewart, David renshaw, martyn riley, “A low-cost, FPGA based video streaming server”, April 2007.

[14] Abdul Wahid Ansari, Mannika garg, “ARM based real-time video, streaming using XBee for perimeter control in defense application, 2014.

[15] Valeria Manuel, florin smaranda, “Control system for video advertising based on Raspberry Pi”, in press.

[16] N vun, M ansary, “Implementation of an Embedde H.264 live video streaming system”, March 2010.

[17] Alexandros G, Fragkiadakis, “Video streaming performance in wireless hostile environments”, 2011.

[18] Kavitha mamindla, Dr V. Padmaja, “Embedded real time video monitoring system using ARM”, IOSR July 2013.

[19] “Real-time streaming protocol 2.0(RTSP)draft-ietf-mmusic-rfc232bis-36.2013-09-11”