

Implementation of Remote Object Monitoring through Video Streaming Based on Bit Caching Algorithm Over Android Platform

Deepak Maheshwari

M.Tech Student, Computer Science and Engg.
Meerut Institute of Engg. & Technology, UPTU
Meerut, India

Dr. Ajay Kumar Singh

Professor (Ph.D.), Computer Science and Engg.
Meerut Institute of Engg. & Technology, UPTU
Meerut, India

Abstract- Video Streaming Video streaming over wireless links is a challenging issue due to the stringent Quality-of-Service (QoS) requirements of video traffic, the limited wireless channel bandwidth and the broadcast nature of wireless medium. For handling the bit delay, firstly we proposed a cache architecture which is implemented over dual buffer. This cache architecture provides us a solution to cache the bits and send that on network, in mean while dual buffer did with its process. So data transmitted through cache blocks and that block fetch data from dual buffer till buffer is not empty. This approach will reduce the fragment delay.

At last we developed an extensive mobile application which include that cache approach and several memory management features. Mobile is used to capture the video at remote station. And processed in MJPG format. It also allows user to view the live image on the Mobile screen while recording. The video is then streamed over Wi-Fi network. The client used in the system is the VLC Media Player which receives the streaming video over Wi-Fi network. This system can be used to monitor the activity of object like disabled person, patients in remote locations, baby, pet monitor and surveillance of employees.

Keywords- Video Streaming, performance analysis, Android Application Development, dual buffer, Cache Algorithm Android Systems Development, Android Market, Web Camera, Mobile Camera, Wi-Fi, IP Network.

I. INTRODUCTION

Live streaming refers to content delivered live over the Internet, requires a form of source media, an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content. Video streaming over wireless networks is of increasing demand, and it becomes possible thanks to the emerging high-speed, short-range wireless communication technologies such as IEEE 802.11n, Ultra-Wideband (UWB) and Millie Meter Wave (MMW). However, supporting wireless video

streaming is a challenging issue due to the stringent Quality-of-Service (QoS) requirements of video traffic, the limited wireless

Channel bandwidth and the broadcast nature of wireless medium. How to utilize the premium wireless resources and support the QoS for video streaming is a challenging issue. Most of the existing wireless standards define either contention-based or contention-free medium access control (MAC) protocols, and some newer ones support both of them in each media access cycle, such as the WiMedia standard (e.g., EMCA- 368) for UWB networks. Contention-based MAC protocols have been widely back off procedures, which may lead to a considerable degradation of video quality. Reservation-based (i.e., contention-free) protocols are often preferred for video streaming, which guarantee the channel access opportunity to satisfy the delay requirement. However, due to the burstiness of video traffic, if reservations are made at the peak data rate to minimize queuing delay, the channel utilization is low; if made at the average data rate, considerable queuing delay and even packet losses can occur during traffic bursts. Therefore, neither reservation nor contention alone is an effective approach for supporting video streaming in wireless networks.

In [1], a hybrid approach for video streaming over wireless networks has been proposed, which uses both contention and reservation-based channel access for transmitting packets from each video flow, by reserving well below the peak data rate and handling traffic bursts using contention-based channel access. By properly selecting the number of time slots reserved for each video stream, it is possible to efficiently utilize the reserved slots and reduce the competition level during the contention periods. To use both reservation and contention access in ad hoc networks, how to define the conflict avoidance strategy in the presence of reservation periods and how to design the buffering architecture are open issues.

In this research, we first deeply study the video streaming transmission over wireless or wired network. In this we studied peer-to-peer video streaming and peer-to-peer video-on-demand in detail. P2P video streaming used two technology for transmission of data i.e. Single-tree Streaming and Multi-tree Streaming.[7] These technologies are good to transmit the data but they have some extensive draw backs. We elaborate those drawbacks and reasons. With the help of this study, we able to find issues like end-to-end bandwidth, upload bandwidth and fragment delays and segment size.[5]

We then study the single buffer, dual buffer architecture of video bit reservation, and proposed a cache buffer algorithm applied on dual buffer architecture and further implement this algorithm in our application. For the single-buffer one, there is only one logical buffer for the video traffic. When the reserved period becomes available for this video flow, packets accumulated in the buffer will be transmitted as many as possible. After its reserved time, the remaining packets can be transmitted by competing with other flows during contention periods.

One major issue with the single-buffer architecture is in a reservation period. For dual buffer architecture Video traffic will first fill up the R-Buffer, which will be transmitted in the upcoming reservation period. When the R-Buffer is full, packets are stored in the C-Buffer, and they will contend for the channel during contention periods.[5,6] In this case, the utilization of the reservation periods can be maximized, so the collisions during contention periods can be minimized.

As we can see that in both architectures time delay is happened at the time of reallocation of bit. It consumes the time when bits are reallocated itself in a specific order, due to this streaming is delayed. So we proposed a cache architecture which is implemented over dual buffer. This cache architecture provides us a solution to cache the bits and send that on network, in mean while dual buffer did with its process. So data transmitted through cache blocks and that block fetch data from dual buffer till buffer is not empty. This approach will reduce the fragment delay.

At last we developed an extensive mobile application which include that cache approach and several memory management features. Mobile is used to capture the video at remote station. And processed in MJPG format. It also allows user to view the live image on the Mobile screen while recording. The video is then streamed over Wi-Fi network. The client used in the system is the VLC Media Player which receives the streaming video over Wi-Fi network. This system can be used to monitor the activity of object like disabled person, patients in remote locations, baby, pet monitor and surveillance of employees.

VLC Media Player is used for playback of streamed video file in the system implemented. It has been used in the system as it supports several data streaming format including MJPG supported by Android Environment.

II. DUAL-BUFFER ARCHITECTURE AS EXISTING SYSTEM

The hybrid MAC approach can be accompanied with different buffering structures. We investigate two buffering architectures for video streaming. The single-buffer and the dual-buffer architectures are illustrated in Fig. 1 a and b, respectively. For the single-buffer one, there is only one logical buffer for the video traffic. When the reserved period becomes available for this video flow, packets accumulated in the buffer will be transmitted as many as possible. After its reserved time, the remaining packets can be transmitted by competing with other flows during contention periods.[5,10]

One major issue with the single-buffer architecture is in a reservation period. Video traffic will first fill up the R-Buffer, which will be transmitted in the upcoming reservation period. When the R-Buffer is full, packets are stored in the C-Buffer, and

they will contend for the channel during contention periods. In this case, the utilization of the reservation periods can be maximized, so the collisions during contention periods can be minimized. Note that, dual-buffer will introduce packet reordering, as packets in the R-Buffer have to wait for the next reservation period, and those in the C-Buffer may experience random queuing delay. We need to quantify the system performance to adjust the system parameters carefully to ensure the QoS for video traffic that, due to the asynchronous nature between video traffic and reservation schedule, it is possible that when a reserved time begins, there is insufficient video packets to send, and thus the reserved period is under-utilized and the contention periods become more crowded. Using a dual-buffer architecture can solve the problem. As shown in Fig. 3.3b, the size of the R-Buffer is set according to the maximum number of packets that can be transmitted.[5.13]

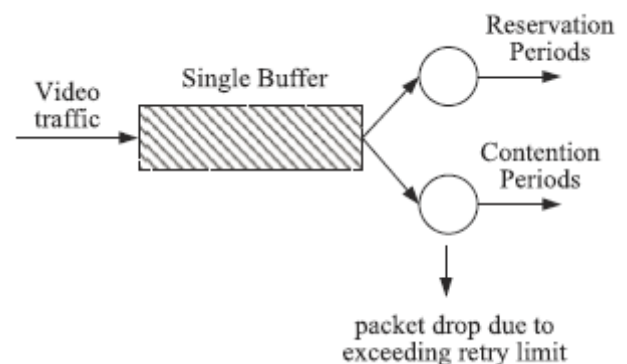


Fig. 1(a) Single Buffer Architecture

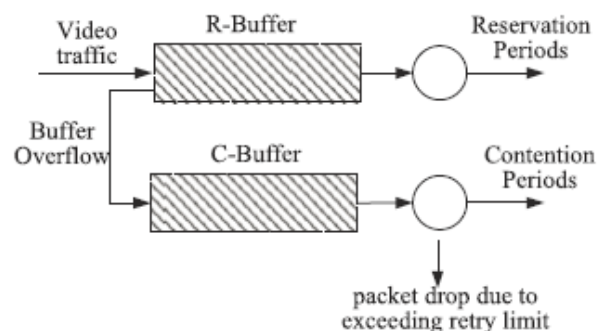


Fig. 1(a) Dual Buffer Architecture

III. PROPOSED CACHE ALGORITHM

As we discussed in proposed system dual buffer architecture is resolved the problem of video traffic delay and video traffic contention. Now in my proposed system I am looking to enhance the speed of packet transfer with help of custom Caching. We all know, Caching is the process which is used to speed up the system. At the time of video packet transmission from R-Buffer and C-Buffer produce some delay in rearranging and rescheduling of bits. In our Approach we will develop the small buffer of cache

and they will attach as a mediator between receiver and dual Buffers .This approach will reduce packet drop problem and delay.

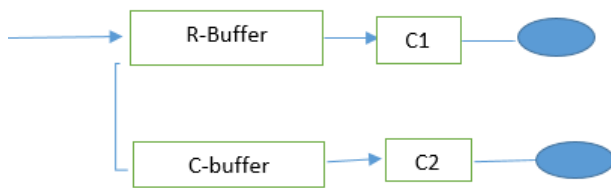


Fig. 2 Dual Buffer with Cache

In the fig. 2 there is two cache buffer (C1, C2) attached for caching the packets for transmission the bits. C1 and C2 will preserve the bits for transmission when R-buffer and C-buffer rearrange the order and all.

A. Cache Algorithm

- 1) Start
- 2) Get Input from Media Codec
- 3) Set Value in dynamic Buffer (Byte Buffer[] input Buffers)
- 4) Set the MediaCodec dequeuebuffer of size 500000 to buffer Index variable.
- 5) if (buffer Index >= 0)
- 6) Clear the input buffers (input Buffers[buffer Index].clear())
- 7) Call Convertor for convert byte to raw data.
- 8) Set the input buffer start index to size of buffer.
- 9) End

IV. ANDROID

Android is a new generation of smart mobile phone platform launched by Google. Android provides the support for various applications. These application make use of Wi-Fi & Broadband features on a mobile phone as a solution for video streaming activity. This research makes use of an Android based mobile phone as the embedded platform, connecting to an IP network using its built-in Wi-Fi interface. The phone's camera performs the video capturing function, whose output is encoded into MPEG4 format. And it is streamed over the Wi-Fi network using the LIVE555 media server. Then the video can be observed at remote stations by adopting the inbuilt media player. And it also gives a study of Android framework and how it functions.[11,12,13,16]

Android's source code is released by Google under open source licenses, although most Android devices ultimately ship with a combination of open source and proprietary software, including proprietary software developed and licensed by Google. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007, along with

the founding of the Open Handset Alliance—a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices[1,17].

Android is popular with technology companies which require a ready-made, low-cost and customizable operating system for high-tech devices. Android's open nature has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users or bring Android to devices which were officially released running other operating systems. The operating system's success has made it a target for patent litigation as part of the so-called "smartphone wars" between technology companies.[1]

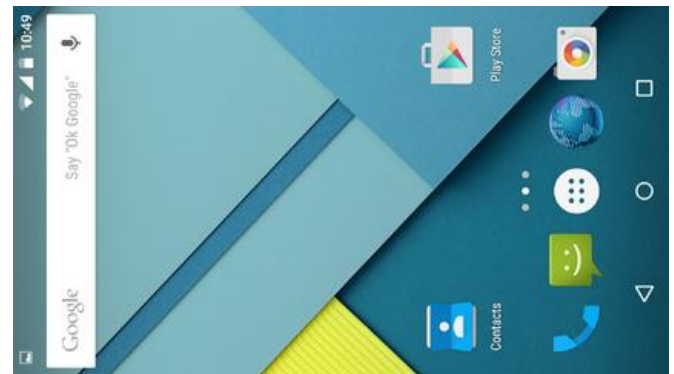


Fig. 3 Android Lollipop

V. APPLICATION FRAMEWORK

Since Android provides an open development platform, it provides the developers' ability to build very rich and advanced applications. Developers are released to take advantage of device hardware, approach location information, executes background services, fix alarms, and contribute notifications to the status bar and so on. Software developers have entire access to the same fabric framework APIs applied by the core application program. This application architecture is planned to alter the reuse of the components. This mechanism grants components to be substituted by the user. The following application is a set of systems and services,

- 1) It has an innovative set of views which is used to build an application including grids, list, buttons, text boxes and even a web browser.
- 2) The content providers enable the application to access data from the other applications or to share their own data.
- 3) The resource Manager provides access to noncore resources such as graphics, strings and layout files.
- 4) The Notification Manager enables all applications which are used to explore custom alert in the status bar.
- 5) The Activity Manager manages the development of the application and allows for a common navigation back stack.

VI. LIBRARIES

Android admits a set of C/C++ libraries are used by several elements of the Android system. The capacities application fabric framework. The core libraries are,

- 1) System C library – BSD derived execution from the authoritative C system library (libc), Adapt for embedded Linux-based devices.
- 2) Media Libraries – Suitivated on Packet Video’s Open CORE, the libraries affirm Playback and showing lots of approved audio and video formats and also still image files, Includes MPEG-4, MP3, AMR, AAC, H.264, PNG and JPG.
- 3) Surface Manager – It accesses the display subsystem and seamlessly complex 2D and 3DGraphic levels from heterogeneous applications.
- 4) LibWebCore – It is a modern web-browser engine which has the Android browser and Web view.
- 5) SGL – The fundamental 2D graphics engine
- 6) 3D libraries – The implementation established on the OpenGL ES 1.0 API.
- 7) Free Type– Vector font rendering and bitmap.

VII. PRELIMINARIES

For creating an application on android phones needs the starter packages and tools that are needed to develop the app. This application is done on the system wise and then it can be installed in the android mobiles. So for the normal PC we want to install starter packages, they are,

- 1) Android SDK starter package
- 2) Android Studio(IntelliJ)
- 3) Android Development Tool (ADT)
- 4) Android NDK
- 5) JDK version6

VIII. ENVIRONMENT SETUP

Development of Android Apps can be conveniently performed using the Android Studio IDE, which is an open-source software development tool that supports multiple languages, including Java. To use it for Android App development, additional extension tools are needed. These extension tools are packaged in the Android Development Tools (ADT) plug-in, which is available from the Google Android Developer Website. In additional, an Android SDK starter package that contains the various software development tool chains and the libraries are needed by the Android Studio to compile and package the application into the Android Package file (apk). Android ADT 19 also includes an Android emulator such that programs can be debugged without using the actual Android phone. Android

Studio provide a bundled environment where all the ADT and AVD already installed. It provide gradle build.

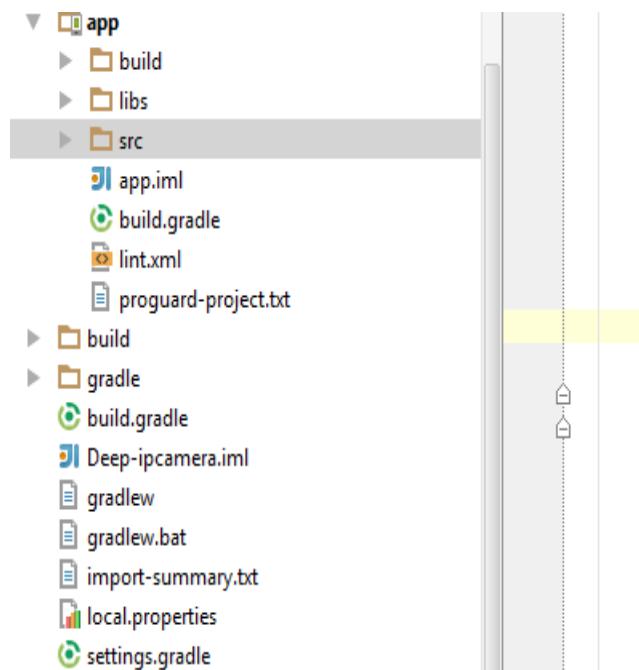


Fig. 4 Enviornment Setup

IX. PROJECT ARCHITECTURE

Figure 5 shows the First level Architecture of the project. As we discussed our project will work on latest platform of Android framework

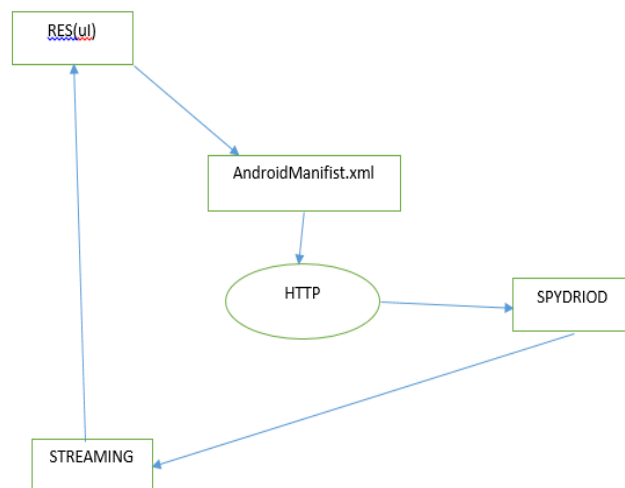


Fig. 5 Project Architecture

A. Res(UI):

Firstly the user will interact with user interface, so the request will come from res (ui) where user can see and request from streaming through your front and default camera. We will make the user interface very user friendly where user can select which video streaming mod needs to choose i.e. H.263 or H.264 and

accordingly bit rates .The user interface is developed by android user interface controller.

B. *AndroidManifest.xml:*

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code.

C. *Http:*

As name mentioned the functionality of this attribute. In this attribute we are going to handle all the http processes like client request and response, ssl for security perspective, Internationalization. So when we will integrate our application with online wowza streaming server then in that case online security and client-server request and response needs to be implement. International protocol integration are also present in this module.

D. *Spydroid Library:*

In this module, first we are implementing api of RTSP and RTTP server and develop a request handler which will handle the request and do specific streaming. Second handling the user interfaces .In that module we are implementing all the utility classes.

E. *Streaming:*

This module have important implementation of this application. In that module, I am developing Streaming classes and proposed cache algorithm.

X. RESULTS AND ANALYSIS

Combining the Video Streaming application running on an Android mobile SAMSUNG Galaxy GT S5570 running Android 4.2 jelly bean, an MJPG video stream system using the Mobile camera is successfully implemented, with the operation of the system as captured in the photo shown in Figure 6. The video of the real time clock captured by mobile Camera is streamed by Android to VLC Media Player through the Wi-Fi interface through LAN router, which is then playback in real time by a VLC Media Player.

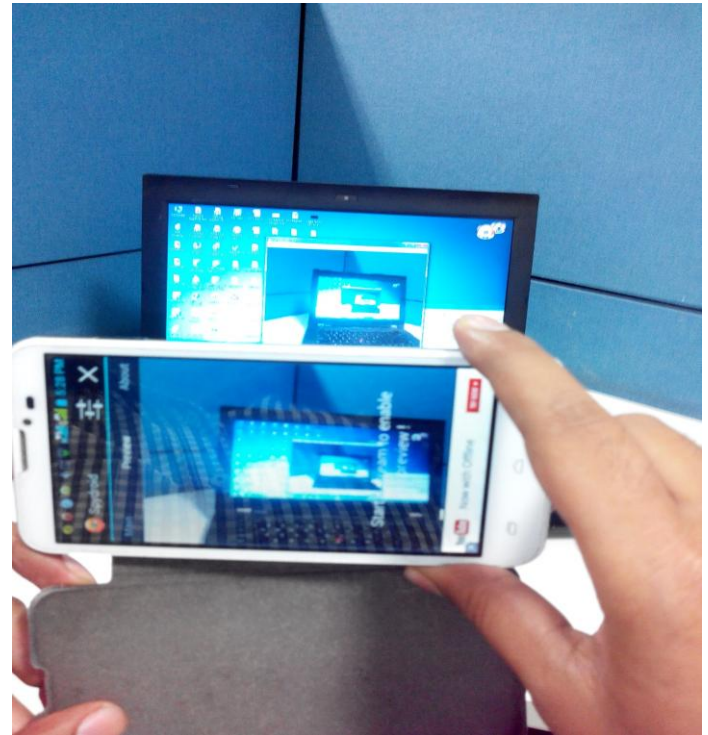


Fig. 6 Live Video Streaming



Fig. 7 Emulator View

Figure 7 shows the application front page view on android studio Emulator.

A. Graph Analysis:

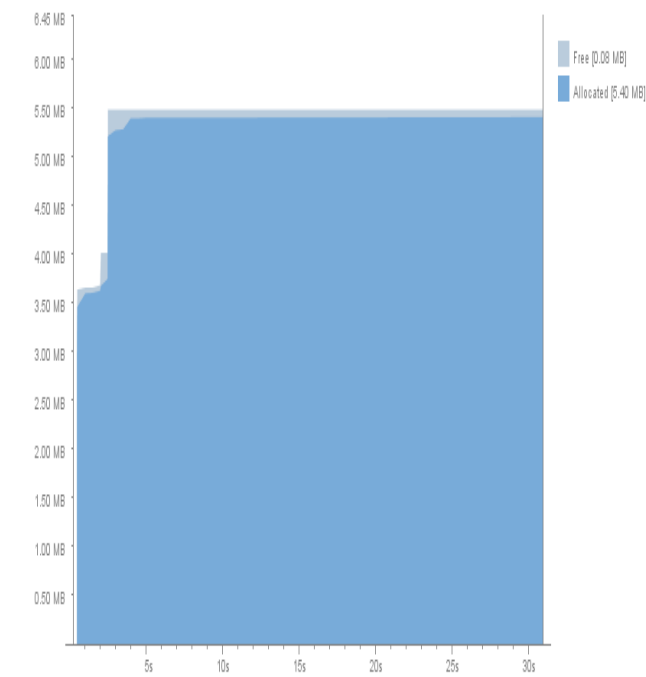


Fig. 8(a) Application memory management

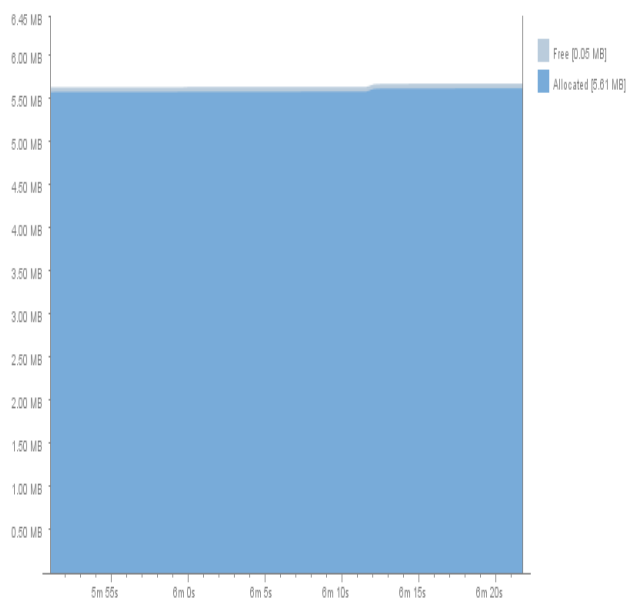


Fig. 8(b) Application memory management

Figure 8(a) and Figure 8(b) shows the application memory management graph. Figure 8(a) represents the graph of application in 0-30 sec, which consumes the memory from 3.50 MB-5.40 MB. Figure 8(b) shows the results of the application after 5 min run and we can see that the memory consumption of application

5.40-5.61 MB. There is only .20 MB size increment in approximate 6 Minutes.

So according to results analysis, our running application is very memory efficient.

XI. CONCLUSION

This work covers almost every terminology and techniques of all Video Streaming area as an introduction and then we cover Video Streaming as IP Camera area as a deep. The primary focus of the work of this project is to show how Video Streaming system works, what is Video Streaming and give basic knowledge of the enhancement of Video Streaming delay through our proposed Cache Algorithm, and the subsequent memory management of Application. We tried to show the step by step process of video Streaming and develop the application on Android Platform.

In the whole project we faced lots of problems regarding finding a good research papers and memory analysis tools. But it's our good fortune that we found latest development tool Android Studio which is able to give the result of memory consumption in the form of graphs, through which we are able to optimize our application and research. As we know Android and wireless video streaming is new hot filed of research in India. That was the new and challenging topic for me to research, but we take it as a challenge and at last we have achieved our proposed and designed task. After studying so much research papers we found the problems to develop a android mobile video streaming application like bit delays of images and memory management. So we were looking to find the way to optimize the delay and memory. At last after doing deep research we got it and develop a good mobile application and optimize the delay with our algorithm.

It describes the successful implementation of a video streamer system using an Android phone as the video capturing device, by integrating a cross-compiled LIVE555 media server as the streaming server. The MPEG4 encoded video captured stream through the phone Wi-Fi connection upon request by a media player (e.g. VLC media player) running on another station connected to the same network. Android technology becomes increasingly popular in the consumer electronics market; there is great motivation to leverage on this open source technology to explore new ideas and concepts, using its essentially free development tools and large collection of applications produced by the open source community. With the largest installed base, it will also be of tremendous incentive to use the Android platform for embedded applications, through modifying and extending the functionalities available in the open source resources. Furthermore, with rich features Android based smart phones and tablets become increasingly available at affordable cost, they provide very convenient hardware platforms for the developers to implement and test their concepts and ideas, with much less effort and at much lower development cost than otherwise.

Use of this application in society as Closed-circuit television (Real time video streaming), also known as video surveillance, is the use of video cameras to transmit a signal to a specific place, on a limited set of monitors. It differs from broadcast television in that the signal is not openly transmitted, though it may employ point to point (P2P), point to multipoint, or

mesh wireless links. Though almost all video cameras fit this definition, the term is most often applied to those used for surveillance in areas that may need monitoring such as banks, casinos, airports, military installations, and convenience stores. Video telephony is seldom called "video streaming" but the use of video in distance education, where it is an important tool, is often so called.

In industrial plants, video streaming equipment may be used to observe parts of a process from a central control room, for example when the environment is not suitable for humans. Video streaming systems may operate continuously or only as required to monitor a particular event. A more advanced form of video streaming, utilizing digital video recorders (DVRs), provides recording for possibly many years, with a variety of quality and performance options and extra features (such as motion detection and email alerts). More recently, decentralized IP cameras, some equipped with megapixel sensors, support recording directly to network-attached storage devices, or internal flash for completely stand-alone operation. Surveillance of the public using video streaming is particularly common in many areas around the world. In recent years, the use of body worn video cameras has been introduced as a new form of surveillance.

XII. REFERENCES

- [1] <http://developer.android.com/guide/basics/whatis-android.html>
- [2] <http://www.eclipse.org>
- [3] <http://developer.android.com/sdk/eclipse-adt.html>
- [4] VLC Media Player, <http://www.videolan.org/vlc/>
- [5] Resource management for video streaming in ad hoc networks Ruonan Zhang a, Lin Cai b, Japing Pan c, Xuemin (Sherman) Shen d.
- [6] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, *IEEE J. Sel. Areas Commun.* 18 (3) (2000) 535–547.
- [7] X. Wang, G. Min, L. Guan, Performance modeling of IEEE 802.11 DCF using equilibrium point analysis, in: *Proc. IEEE AINA'06*, 2006.
- [8] X. Ling, K.H. Liu, Y. Cheng, X. Shen, J.W. Mark, A novel performance model for distributed prioritized MAC protocols, in: *Proc. IEEE Globecom'07*, Washington, DC, USA, November 2007.
- [9] L.X. Cai, X. Shen, L. Cai, J. Mark, Y. Xiao, Voice capacity analysis of WLAN with un-balanced traffic, *IEEE Trans. Veh. Technol.* 55 (3) (2006) 752–761.
- [10] Y. Xiao, Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs, *IEEE Trans. Wireless Commun.* 4 (4) (2005) 1506–1515.
- [11] C. Hu, H. Kim, J. Hou, D. Chi, S.S. Nandagopalan, Provisioning quality controlled medium access in UltraWideBand-operated WPANs, in: *Proc. IEEE INFOCOM'06*, Barcelona, Spain, April 2006.
- [12] K.H. Liu, X. Shen, R. Zhang, L. Cai, Performance analysis of distributed reservation protocol for UWB based WPAN, *IEEE Trans. Veh. Technol.* 58 (2) (2009) 902–913.
- [14] Resource management for video streaming in ad hoc networks Ruonan Zhang a, Lin Cai b, Jianping Pan c, Xuemin (Sherman) Shen d.
- [15] Annapureddy, S., Guha, S., Gkantsidis, C., Gunawardena, D., & Rodriguez, P. R. (2007). Is high-quality vod feasible using P2P swarming? *Proceedings of the 16th International Conference on World Wide Web*, 903-912.
- [16] B. Cohen. Incentives build robustness in bittorrent. <http://bitconjurer.org/BitTorrent/bittorrentecon.pdf> May 2003.
- [17] Chu, Y., ao, S., Seshan, S., & Zhang, H. (2002). "A case for end system multicast". *Selected Areas in Communications, IEEE Journal on*, 20(8), 1456-1471.
- [18] EMULE . Emule Homepage. <http://www.emule-project.net/>.
- [19] Guo, Y., Suh, K., Kurose, J., & Towsley, D. (2003). "P2Cast: Peer-to-peer patching scheme for VoD service". *Proceedings of the 12th International Conference on World Wide Web*, 301-309.
- [20] Hua, K. A., Cai, Y., & Sheu, S. (1998). "Patching: A multicast technique for true video-on-demand services". *Proceedings of the Sixth ACM International Conference on Multimedia*, 191-200.
- [21] Internet Protocol Multicast, ch 43 of *Internetworking Technology Handbook*, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ipmulti.pdf