

Creating A Video Streamer App On Android Mobile Phones

M.Adimoolam¹, M.Gunashanthi², K.Hemachandran³

Information Technology, Christ college of Engineering and Technology, Puducherry.

Abstract

Android is a new generation of smart mobile phone platform launched by Google. Android provides the support for various applications. These applications make use of Wi-Fi & Broadband features on a mobile phone as a solution for video streaming activity. This paper 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.

1. Introduction

1.1. Android Operation System

Android is a Linux-based operating system for mobile devices such as Smartphone's and tablet computers. It was first developed by the Open Handset Alliance which was lead by Google Corporation. And then Google purchases the primary developer of the software, i.e. by Android Inc, in 2005 [1]. The introduction of the Android distribution in 2007 was legally announced with the Open Handset Alliance, an association of 86 hardware and software, and telecommunication companies dedicate to promoting open standards for mobile agent. Google issue the Android Code as an Open source, beneath the Apache License. AOSP - the Android Open Source Project is associated with the maintenance and promotes the growth of Android [1,3].

The Android Software Development Kit (Android SDK) allows for all required tools to develop Android applications. This also includes a compiler, device emulator and a debugger, also has its own virtual machine running Android programs. Android grants background processing, which provides a rich user interface library, affirms 2-D and 3-D graphics by accessing the OpenGL libraries, approach to the file

system and allows for an embedded SQLite database [2].

And this paper gives the objectives of Android Architecture and creating an application using LIVE555 media server. This application (app) can be done using the Android SDK and Eclipse IDE. This paper also includes the Dalvik virtual machine and implementation of video streaming app on android based mobile phones [3] [4].

And it also gives how to develop an app using Android NKD and eclipse classic IDE. And the video can be streamed over Wi-Fi network and the video can be viewed using inbuilt media streamer [3] [4] [5].

2. Android architecture

Android architecture can be broadly separated into 5 components as shown in Figure1. Sitting on top of the Kernel is the Libraries component that contains the various C/C++ Libraries that can be used by the Applications [1] [3]. On top of the Libraries is the Application Framework which provides the APIs and services that is used by the core applications, and can also be used by custom applications (commonly known as Apps). The Android Apps, located at the top Applications layer, are written using the Java programming language [6].

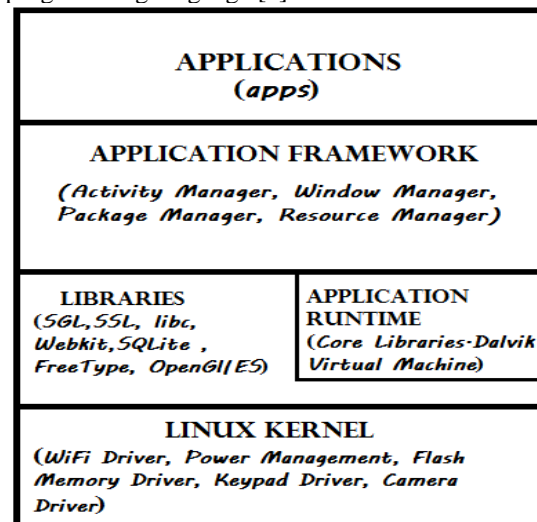


Fig 1: Android Architecture

2.1 Application

Android applications are framed up of one or more application elements such as, activities, services, content providers and broadcast receiver. Each element executes at different functions in the overall application conduct and each one can be actuated separately.

The manifest file must hold all components in the application and ought to also declare all application necessity, such as the minimal version of android needed and any hardware contour necessitated [7]. Non-code application resources such as string, image, layout files, etc. and should admits alternatives for dissimilar device configuration such as different thread for different languages [7,8].

2.2 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 [3].

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 [8] [9].

The following application is a set of systems and services,

- 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.
- The content providers enable the application to access data from the other applications or to share their own data.
- The resource Manager provides access to non-code resources such as graphics, strings and layout files.
- The Notification Manager enables all applications which are used to explore custom alert in the status bar.
- The Activity Manager manages the development of the application and allows for a common navigation backstack.

2.3 Libraries

Android admits a set of C/C++ libraries are used by several elements of the Android system. The capacities are disclosed to developers through with the Android

application fabric framework. The core libraries are [6,8],

- **System C library** – BSD derived execution from the authoritative C system library (libc), adapt for embedded Linux-based devices.
- **Media Libraries** – Suitivated on PacketVideo's OpenCORE, 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.
- **Surface Manager** – It accesses the display subsystem and seamlessly complex 2D and 3D graphic levels from heterogeneous applications.
- **LibWebCore** – It is a modern web-browser engine which has the Android browser and a web view
- **SGL** – The fundamental 2D graphics engine
- **3D libraries** – The implementation established on the OpenGL ES 1.0 API.
- **FreeType** – Vector font rendering and bitmap.
- **SQLite** – The lightweight and mighty relational database are available to all developing applications.

2.4 Android Runtime

The android framework provides a set of basic libraries that afford most of the core functionality present in the libraries of the Java programming language.

The Android Application run on its own process, along with the instance of the Dalvik virtual machine. The Dalvik virtual machine creates an environment that a device can run at any multiple VMs effectively [10]. The Dalvik VM executes files and store file as Dalvik Executable (.dex) format, which is used for the optimization for minimum memory footprint. The Dalvik VM is register based and runs Java class files. The Java class files which are compiled by Java compiler is converted into .dex file by using Dalvik VM. IT also includes "dx" tool.

The Dalvik Virtual Machine awaits on the Linux Kernel for elementary functionality such as low-level memory management and threading.

2.5 Linux Kernel

Android is extended and relies on Linux version 2.6 for for basic system services such as memory management, process, security, driver model, management and network stack. The kernel present in Android is acting as an abstraction layer between hardware and software stack [11].

3. 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,

- Android SDK starter package
- Eclipse IDE (Indigo)
- Android Development Tool (ADT)
- Cygwin
- Android NDK
- JDK version6

First we want to install Android SDK

First Install Eclipse, in which the Indigo or classic version is recommended. And install the Android SDK starter package with the latest version. Add the Android Development Tools (ADT) plug-in in the Eclipse IDE. The Eclipse is downloaded from the below website [12].

<http://www.eclipse.org/downloads>

The “Eclipse Classic” version is highly recommended. Otherwise JAVA or RCP version of Eclipse package is recommended [13].

Then the Android offers a custom plug-in for the Eclipse IDE, which is called as Android Development Tools (ADT) that is designed to integrate the environment to build Android applications.

It then the capability of the Eclipse let to adopt the Android platform quickly. This creates the Android project application UI, which implements the Android project using the Android SDK and APK in order to distribute the application. For developing the Eclipse with ADT to do Android project easily [11] [13].

4. Existing system

The existing system uses the older version of android OS i.e. version1.5 mobile phones. It does not support for multimedia and video transferring. And they didn't develop an application; instead they root the video streamer into the OS itself.

If there occur error in building video streamer into Android phones may affect the OS directly and it cause damage the whole system has broken. The in build video streamer a raise problem as maximum time delay and it does not support for USB port. Another consequence of using the file buffer is that the system will stop operation once the file size exceeds the maximum limit that the Android platform can support [1] [3].

5. Proposed system

In the proposed system we use Android OS of version 2.2 . And we are creating an application to overcome the time delay and damage of the Android OS.

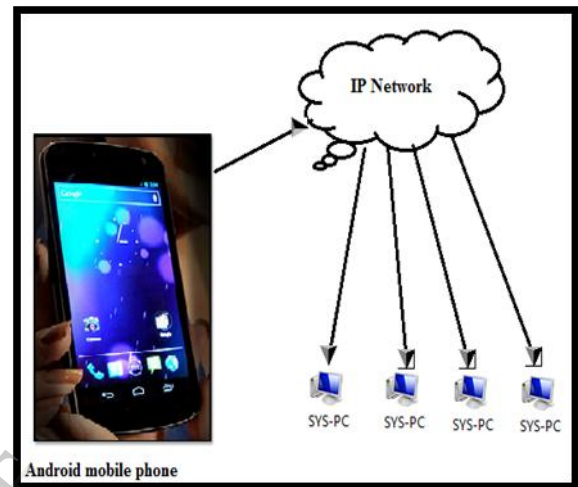


Fig: 2 System set up

This system setup deploy that a camera on the Android Mobile phone takes a real time image and streamed using LIVE555 media server [17] and send through the Wi-Fi network by accessing the IP address of the system. And it can be viewed in the local Pc's using VLC player. The video format is in MPEG4 format [15] [16].

6. Environment Setup

Development of Android Apps can be conveniently performed using the Eclipse 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 Eclipse to compile and package the application into the Android Package file (apk) [13] [15].

Android ADT also includes an Android emulator such that programs can be debugged without using the actual Android phone. The Eclipse IDE gives a basic environment to develop an application, In that ADT is

installed and the SDK starter package. Then create AVD by clicking the AVD icon in the eclipse then create a new AVD by giving the related hardware and software components needed to develop the application. Then the new AVD is created [14].

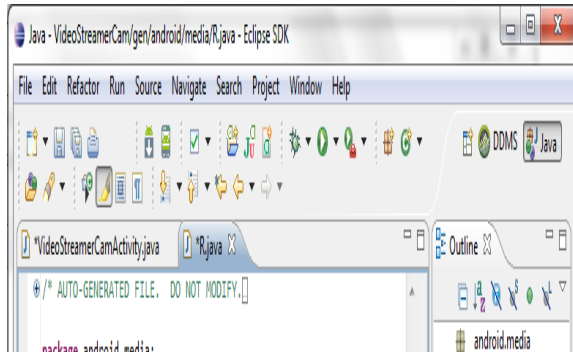


Fig 3: Eclipse IDE

The fig 3 shows the environmental set up for the android app development. In this environment only we can create the android project. Here we use Eclipse classic IDE for better development.

And then the new android project is developed by creating a new project name, activity and the platform. Then the source code is attached in the eclipse. And create an Android project as [13],

File→new→android project

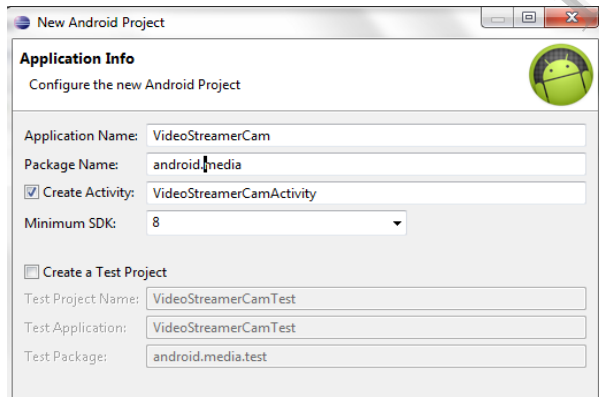


Fig 4: Creating a new Android project

In that android project give project name, package name and activity, and then choose a target SDK package, new project is developed in the workspace. Then in the application info give the activity and SDK minimum package [7] [8]. And give finish, the android package is created in the workspace.

Usually the application is written in the Java programming language only, but LIVE 555 media

server is in the C++ code for that we want to convert that as Java native code by using the Android Native Development Kit (NDK) [14]. The NDK converts the C/C++ code into native code using the Java interface method. But the NDK is only supported in Linux and not in windows, so that we are installing Cygwin which gives a Linux environment in windows [18].

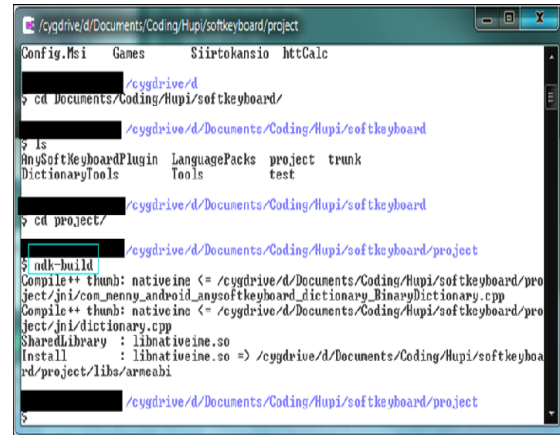


Fig 5: Cygwin

The LIVE555 class file and android media recorder source code are integrated in the eclipse, The dx tool converts the Java class file into .dex (dalvik executable) file [19].

In the project package the “res” contains the structural value of the Android platform in the directory “assets” which is used to accumulate any kind of data within. In the Java we can access the data through the Assets Manager can be accessed by the method getAssets().

The .dex file is converted into .apk file using aapt. Then the application is created using Android Assets Packing tool (aapt) which is included in the AVD it. And it also creates an application in AVD [17] [18] [19].

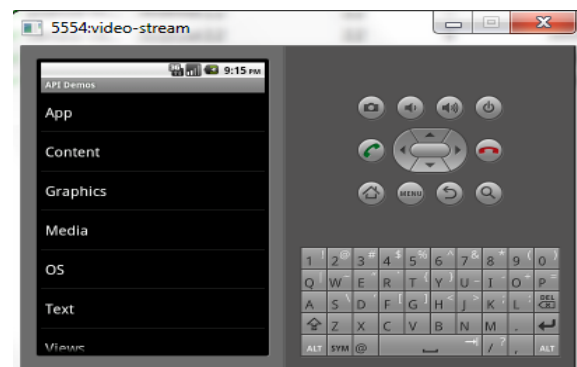


Fig 6: Emulator

The Android SDK which as an emulator includes a mobile device that runs on the computer. The emulator is a virtual mobile device that runs on the computer. The emulator lets to develop and test the Android applications without using the physical device. The Android Virtual Device (AVD) is an Emulator that has a model to actual device by defining the specification. Hardware and software options are emulated by the Android Emulator.

To create an AVD by using the graphical AVD manager, this can be done by the Eclipse IDE by clicking Windows-> Android SDK and AVD Manager. The AVD manager can also start from the command line by using the Android tool in the tools directory of the Android SKD [20].

The AVD can be created by using AVD manager, which can be launched by Eclipse by, Windows-> Android SDK and AVD manager. We can also start the AVD manager from command line using the android tool in the tools directory.

Android Debugger Bridge (adb) is used to install the application in the mobile wise and run the application in mobile wise.

7. Implementation and Result

The video streamer application is created and installed in the Android mobile using adb tool. First the finds the nearby Wi-Fi connection using Wi-Fi finder. And then the mobile gives request the remote PC by giving the IP address of the system. And the system accepts the request and give the mobile ID in the request , then the real time image can be streamed over the Wi-Fi network and it can be viewed in the remote station using VLC media player.

The experiment shows that the video of the clock running on the first PC monitor (on the right in the photo) is captured by the Android phone. The video is then streamed through the phone Wi-Fi link to a LAN router, which is then playback in real time by a VLC player running on the second PC [21].

Using a real-time running clock as the image source, it is noticed that there is no observable throughput delay in the streaming process under normal network condition. However, though not clearly noticeable in the photo, there is playback frame delay between the source and the playback image. The delay will depend on the instance the VLC media player is started. This is due to the use of the file as a buffer between the two software modules running on the Android phone. While the video recorder module will start recording the video into the file upon launching by the user, the LIVE555 will only start streaming the video upon receiving the request from the VLC player.

But the VLC player must start early to avoid time delay [16] [17].

7. Conclusion

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.

9. Reference

- [1] <http://developer.android.com/guide/topics/resource/index.html>
- [2] <http://developer.android.com/reference/packages.html>
- [3] "A study on present and future of Google's Android", Android by 2012.
- [4] <http://www.live555.com/mediaServer/>
- [5] Wireless Network Setup Guide for GovWiFi – Android 2.2
- [6] Android OS: A robust, free, open-source operating system for mobile devices, Paul Michael Kilgo.
- [7] AT&T Developer Program, "Developing Applications for Android" 2007.
- [8] <http://developer.android.com/about/index.html>
- [9] "Android™ Professional Application Development", Reto Meier, 2009 by Wiley Publishing, Inc., Indianapolis, Indiana.
- [10] Dan Bornstein, "Dalvik VM internal", Goggle Presentation 2008-05-29.
- [11] "Developing Android™ applications for ARM® Cortex™-A8 cores", Darren Etheridge, Android Multimedia Engineering Manager, DSP and ARM, Texas Instruments.

- [12] Lars Vogel, "Android Development with Eclipse – Tutorial".
- [13] "Eclipse.org CDT and Cygwin: A Tutorial on Installation and Functionality", Christopher T. S. Allen, Department of Computer Science and Statistics, University of Rhode Island – Undergraduate
- [14] "Android Beginners: NDK Setup Step by Step", mindtherobot 2011.
- [15] N. Vun and M Ansary, "Implementation of an Embedded H.264 Live Video Streaming System", IEEE ISCE 2010.
- [16] N Vun, Y H Ooi, "The Implementation of an Android Phone Based Video Streamer", 2010 IEEE International Conference on Green Computing and Communications.
- [17] Enhancement of Live555 Media Server to support MPEG4 streaming Aravind V, Archan Pratap Mishra, Ch. Arjun Kumar Reddy, Sneha S, Sumit Kishore, Technical Report IIITB-OS-2010-4b, April 2010.
- [18] <http://www.softwaregeek.com/android-media-server/freeware/p1.html>
- [19] <https://www.freelancer.ph/work/live555-media-server-wrapper/>
- [20] Kumar, , Guru Gobind Singh, Indraprastha University, "Development and Research Implementation of Remote Object Monitoring Through Video Streaming Based on Android Mobile", Sumit International Journal of Internet Computing (IJIC), ISSN No: 2231 – 6965, Volume-1, Issue-2, 2011.
- [21] "A Study of Android Application Security" William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri, Systems and Internet Infrastructure Security Laboratory Pennsylvania State University.