

AMES-Cloud A Framework of Adaptive Mobile Video Streaming and Efficient Social Video Streaming

¹m.Renuka, ²g.Ravi Kumar, ³t.Venu Gopal, ⁴g.Nagendra, ⁵v.Sridhar

^{1,2,5}Assistant professor, ^{3,4}Associate professor^{1,2,3,4,5}Department of Electronics and communication Engineering^{1,2,3,4,5}Vidya Jyothi Institute of Technology, Aziz Nagar, Hyderabad, Telangana, India

Abstract - While demands on video traffic over mobile networks have been souring, the wireless link capacity cannot keep up with the traffic demand. The gap between the traffic demand and the link capacity, along with time-varying link conditions, results in poor service quality of video streaming over mobile networks such as long buffering time and intermittent disruptions. Leveraging the cloud computing technology, we propose a new mobile video streaming framework, dubbed AMES-Cloud, which has two main parts: AMoV (adaptive mobile video streaming) and ESoV (efficient social video sharing). AMoV and ESoV construct a private agent to provide video streaming services efficiently for each mobile user. For a given user, AMoV lets her private agent adaptively adjust her streaming flow with a scalable video coding technique based on the feedback of link quality. Likewise, ESoV monitors the social network interactions among mobile users, and their private agents try to prefetch video content in advance. We implement a prototype of the AMES-Cloud framework to demonstrate its performance. It is shown that the private agents in the clouds can effectively provide the adaptive streaming, and perform video sharing based on the social network analysis.

Keywords: AMES-Cloud framework, mes, movs,java.

1. INTRODUCTION

Cloud computing promises lower costs, rapid scaling, easier maintenance, and services that are available anywhere, anytime. A key challenge in moving to the cloud is to ensure and build confidence that user data is handled securely in the cloud. A recent Microsoft survey [10] found that "...58% of the public and 86% of business leaders are excited about the possibilities of cloud computing. But, more than 90% of them are worried about security, availability, and privacy of their data as it rests in the cloud."

There is tension between user data protection and rich computation in the cloud. Users want to maintain control of their data, but also want to benefit from rich services provided by application developers using that data. At present, there is little platform-level support and standardization for verifiable data protection in the cloud. On the other hand, user data protection while enabling rich computation is challenging. It requires specialized expertise and a lot of resources to build, which may not be readily available to most application developers.

We argue that it is highly valuable to build in data protection solutions at the platform layer: The platform can be a great place to achieve economy of scale for security, by amortizing the cost of maintaining expertise and building sophisticated security solutions across different applications and their developers.

2. PURPOSE OF THE PAPER

2.1 Target Applications:

Over the past decade, increasingly more traffic is accounted by video streaming and downloading. In particular, video streaming services over mobile networks have become prevalent over the past few years. While the video streaming is not so challenging in wired networks, mobile networks have been suffering from video traffic transmissions over scarce bandwidth of wireless links. Despite network operators' desperate efforts to enhance the wireless link bandwidth soaring video traffic demands from mobile users are rapidly overwhelming the wireless link capacity. While receiving video streaming traffic via 3G/4G mobile networks, mobile users often suffer from long buffering time and intermittent disruptions due to the limited bandwidth and link condition fluctuation caused by multi-path fading and user mobility. Thus, it is crucial to improve the service quality of mobile video streaming while using the networking and computing resources efficiently. Recently there have been many studies on how to improve the service quality of mobile video streaming on two aspects:

2.2 Scalability:

Mobile video streaming services should support a wide spectrum of mobile devices; they have different video resolutions, different computing powers, different wireless links (like 3G and LTE) and so on. Also, the available link capacity of a mobile device may vary over time and space depending on its signal

strength, other users traffic in the same cell, and link condition variation. Storing multiple versions (with different bit rates) of the same video content may incur high overhead in terms of storage and communication. To address this issue, the Scalable Video Coding (SVC) technique (Annex G extension) of the H.264 AVC video compression standard defines a base layer (BL) with multiple enhance layers (ELs). These substreams can be encoded by exploiting three scalability features: (i) spatial scalability by layering image resolution (screen pixels), (ii) temporal scalability by layering the frame rate, and (iii) quality scalability by layering the image compression. By the SVC, a video can be decoded/played at the lowest quality if only the BL is delivered. However, the more ELs can be delivered, the better quality of the video stream is achieved.

2.3 Adaptability:

Traditional video streaming techniques designed by considering relatively stable traffic links between servers and users, perform poorly in mobile environments. Thus the fluctuating wireless link status should be properly dealt with to provide "tolerable" video streaming services. To address this issue, we have to adjust the video bit rate adapting to the currently time-varying available link bandwidth of each mobile user. Such adaptive streaming techniques can effectively reduce packet losses and bandwidth waste. Scalable video coding and adaptive streaming techniques can be jointly combined to accomplish effectively the best possible quality of video streaming services. That is, we can dynamically adjust the number of SVC layers depending on the current link status. However most of the proposals seeking to jointly utilize the video scalability and adaptability rely on the active control on the server side. That is, every mobile user needs to individually report the transmission status (e.g., packet loss, delay and signal quality) periodically to the server, which predicts the available bandwidth for each user. Thus the problem is that the server should take over the substantial processing overhead, as the number of users increases.

Cloud computing techniques are poised to flexibly provide scalable resources to content/service providers, and process offloading to mobile users. Thus, cloud data centers can easily provision for large-scale real-time video services as investigated in. Several studies on mobile cloud computing technologies have proposed to generate personalized intelligent agents for servicing mobile users, e.g., Cloudlet and Stratus. This is because, in the cloud, multiple agent instances (or threads) can be maintained dynamically and efficiently depending on the time-varying user demands.

Recently social network services (SNSs) have been increasingly popular. There have been proposals to improve the quality of content delivery using SNSs [23] [24]. In SNSs, users may share, comment or re-post videos among friends and members in the same group, which implies a user may watch a video that her friends have recommended. Users in SNSs can also follow famous and popular users based on their interests (e.g., an official facebook or twitter account that shares the newest pop music videos), which is likely to be watched by its followers. In this regard, we are further motivated to exploit the relationship among mobile users from their SNS activities in order to prefetch in advance the beginning part of the video or even the whole video to the members of a group who have not seen the video yet. It can be done by a background job supported by the agent (of a member) in the cloud; once the user clicks to watch the video, it can instantly start playing.

3. LITERATURE SURVEY:

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

3.1 Adaptive Mobile Video Streaming (AMoV):

In this paper, we design an adaptive video streaming and prefetching framework for mobile users with the above objectives in mind, dubbed AMES-Cloud. AMES-Cloud constructs a private agent for each mobile user in cloud computing environments, which is used by its two main parts: (i) AMoV (adaptive mobile video streaming), and (ii) ESov (efficient social video sharing). The contributions of this paper can be summarized as follows:

AMoV offers the best possible streaming experiences by adaptively controlling the streaming bit rate depending on the fluctuation of the link quality. AMoV adjusts the bit rate for each user leveraging the scalable video coding. The private agent of a user keeps track of the feedback information on the link status. Private agents of users are dynamically initiated and optimized in the cloud computing platform. Also the real-time SVC coding is done on the cloud computing side efficiently.

3.2 Efficient Social Video Sharing(ESoV):

AMES-Cloud supports distributing video streams efficiently by facilitating a 2-tier structure: the first tier is a content delivery network, and the second tier is a data center. With this structure, video sharing can be optimized within the cloud. Unnecessary redundant downloads of popular videos can be prevented.

Based on the analysis of the SNS activities of mobile users, ESoV seeks to provide a user with instant playing of video clips by prefetching the video clips in advance from her private agent to the local storage of her device. The strength of the social links between users and the history of various social activities can probabilistically determine how much and which video will be prefetched.

4. SYSTEM ANALYSIS

4.1 Existing System:

Cloud computing promises lower costs, rapid scaling, easier maintenance, and service availability anywhere, anytime, a key challenge is how to ensure and build confidence that the cloud can handle user data securely. A recent Microsoft survey found that "58 percent of the public and 86 percent of business leaders are excited about the possibilities of cloud computing. But more than 90 percent of them are worried about security, availability, and privacy of their data as it rests in the cloud.

4.2 Proposed System:

We propose an adaptive mobile video streaming and sharing framework, called AMES-Cloud, which efficiently stores videos in the clouds (VC), and utilizes cloud computing to construct private agent (subVC) for each mobile user to try to offer "non-terminating" video streaming adapting to the fluctuation of link quality based on the Scalable Video Coding technique. Also AMES-Cloud can further seek to provide "nonbuffering" experience of video streaming by background pushing functions among the VB, subVBs and localVB of mobile users. We evaluated the AMES-Cloud by prototype implementation and shows that the cloud computing n implementation, designing of methods to achieve changeover and evaluation of changeover methods.

technique brings significant improvement on the adaptivity of the mobile streaming. We ignored the cost of encoding workload in the cloud while implementing the prototype.

Advantages of Proposed System:

- Security is achieved.
- Denil and delay in service is avoided.

5. SYSTEM REQUIREMENTS

5.1 Hardware:

Processor	:Pentium –III
Speed	: 1.1 GHz
RAM	: 256 MB (min)
Hard Disk	: 20 GB
Floppy Drive	: 1.44 MB
Key Board	: Standard Windows Keyboard
Mouse	: Two or Three Button
Monitor	:SVGA

5.2 Software:

Operating System	:
Windows	:95/98/2000/XP
Application Server	:Tomcat5.0/6.X
Front End	:HTML, Java, Jsp
Scripts	: JavaScript.
Server side Script	: Java Server
Pages.	
Database	: Mysql
Database Connectivity	: JDBC.

6. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints o

7. OUTPUT SCREENS



Fig : 9.1 Home Page

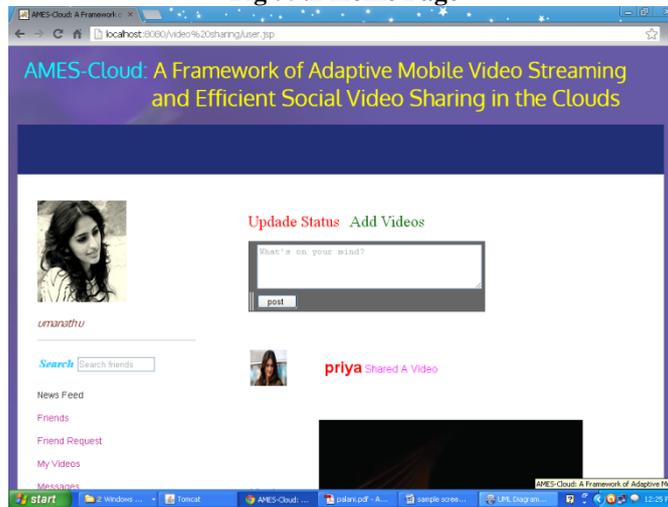


Fig : 9.2 User Login

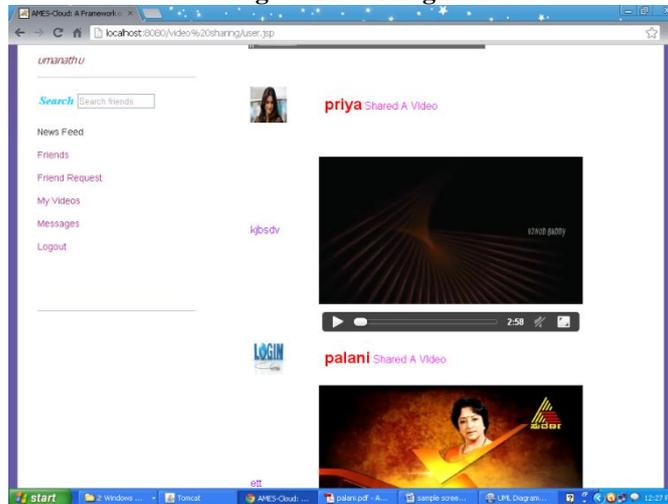


Fig : 9.3

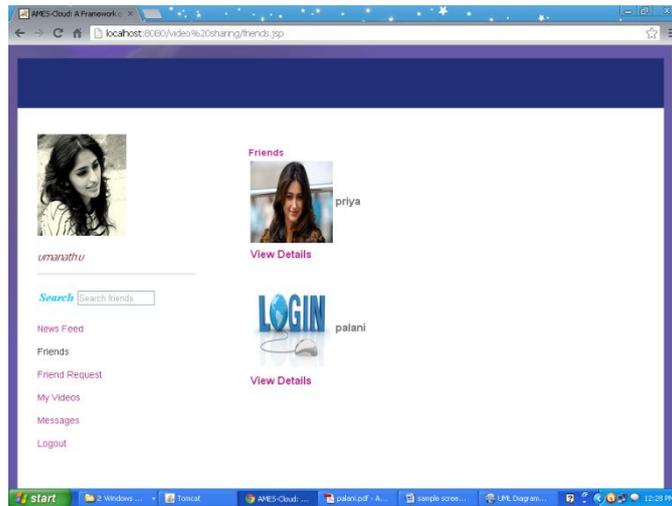


Fig : 9.4

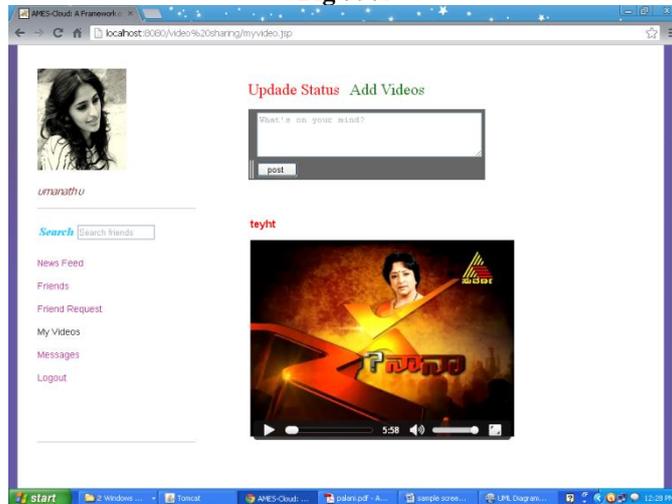


Fig : 9.5

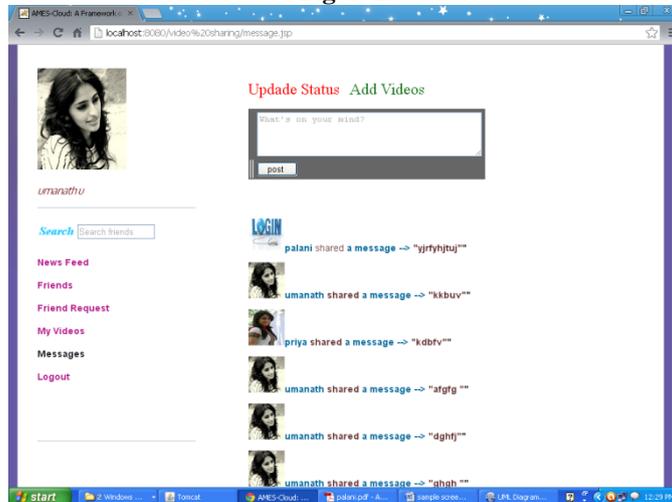


Fig : 9.6



Fig : 9.7

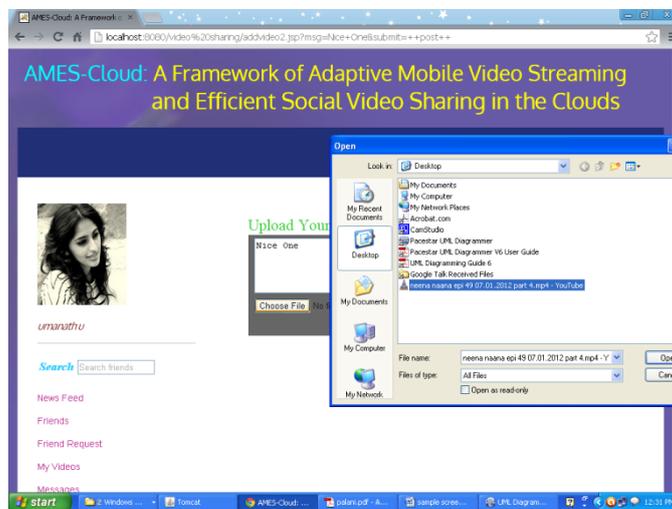


Fig : 9.8



Fig : 9.9

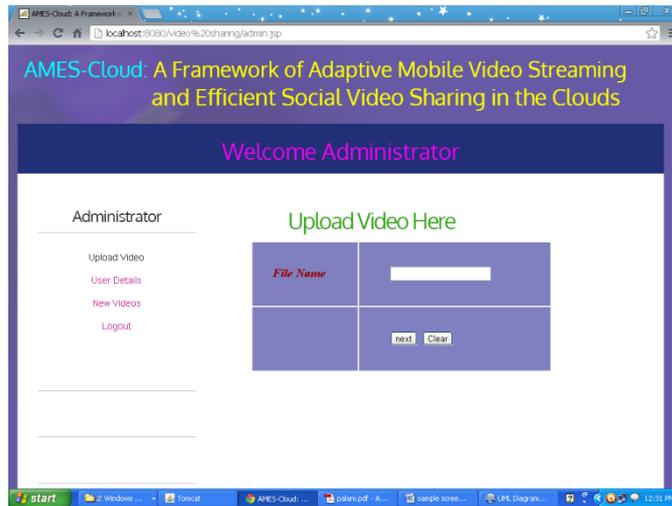


Fig : 9.10

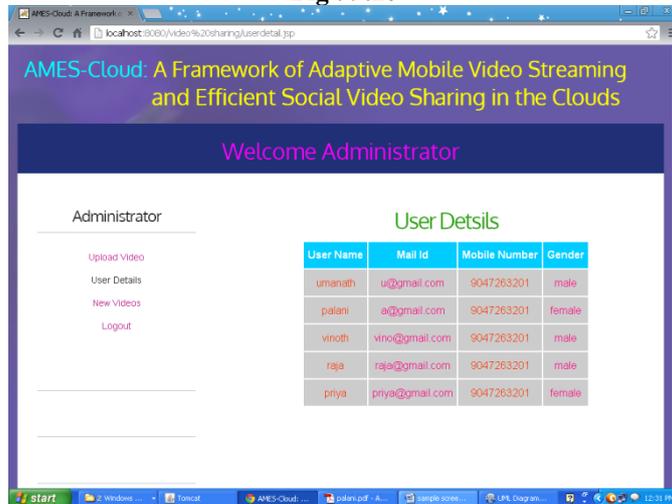


Fig : 9.11

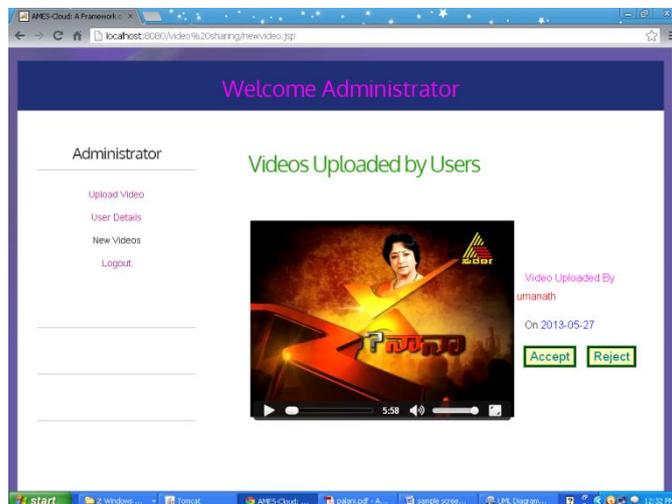


Fig : 9.12

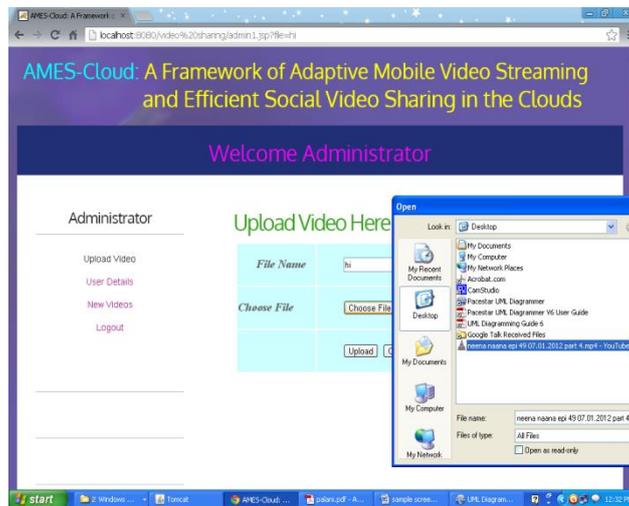


Fig : 9.13

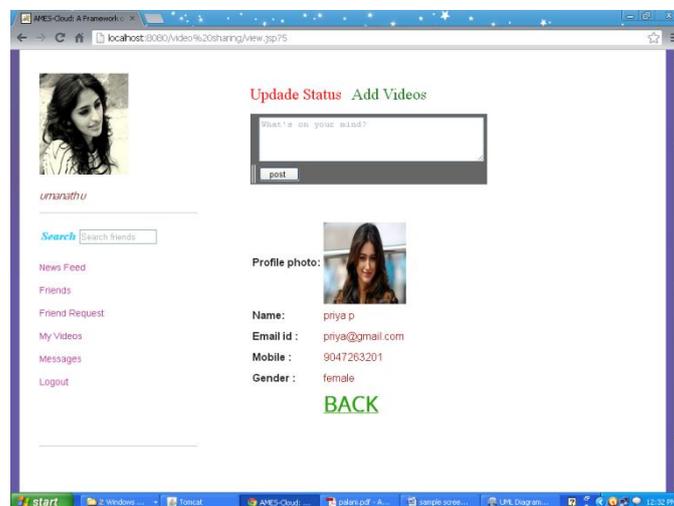


Fig : 9.14

10. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement

10.1 Test Plan:

A test-plan is basically a list of test cases that need to be run on the system. Few test cases for some components like report generation from database, can be tested independently where as others require the

whole system to be ready for their execution. For better results each component is unit tested as and when it is ready before integrating the components. It is ensured that the test cases cover all the aspects of the system and meet all the requirements stated in the RS document.

11. CONCLUSION

In this paper, we discussed our proposal of an adaptive mobile video streaming and sharing framework, called AMES-Cloud, which efficiently stores videos in the clouds (VC), and utilizes cloud computing to construct private agent (subVC) for each mobile user to try to offer "non-terminating" video streaming adapting to the fluctuation of link quality based on the Scalable Video Coding technique. Also AMES-Cloud can further seek to provide "nonbuffering" experience of video streaming by

background pushing functions among the VB, subVBs and localVB of mobile users. We evaluated the AMES-Cloud by prototype implementation and shows that the cloud computing technique brings significant improvement on the adaptivity of the mobile streaming. The focus of this paper is to verify how cloud computing can improve the transmission adaptability and prefetching for mobile users. We ignored the cost of encoding workload in the cloud while implementing the prototype.

12. FUTURE SCOPE

As one important future work, we will carry out large-scale implementation and with serious consideration on energy and price cost. In the future, we will also try to improve the SNS-based prefetching, and security issues in the AMES-Cloud.

REFERENCES

- [1]Hello, Android, E. Burnette, the Pragmatic Programmers (2009).
- [2]Professional Android 2 Application Development, R. Meier, Wiley (2010).
- [3]Beginning Android 2, M. Murphy, Apress (2010).
- [4]Android Wireless Application Development, S. Conder and L. Darcey, Addison-Wesley (2010).
- [5]Android Application Development in 24 Hours, L. Darcey and S. Conder, Sams (2010).
- [6]The Android Developer's Cookbook, J. Steele, N. To, Addison-Wesley (2011).
- [7]Mark L. Murphy ,”Managing and Accessing local dabase, 455-504 ”The Busy Coader’s Guide to Android Development
- [8]Mark L. Murphy ,”Accessign Location-based Services, 563-569 ”The Busy Coader’s Guide to Android Development.
- [9]Mark L. Murphy ,”Playing media”, 165-187 ”The Busy Coader’s Guide to Advance Android Development