International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

Cloud Based: Video on Demand Mobile Social TV

Srishti Aslen Marandi[1], Panimozhi K[2]

[1]M.Tech 4th sem, Department of Computer Science and Engineering, AMC Engineering College, Bangalore-560083
[2]Department of Computer Science and Engineering, AMC Engineering College, Bangalore – 560083

***Abstract*—Today smartphones combines the features of a mobile phone with those of another popular consumer device, such as a personal digital assistant , a media player, a digital camera, and/or a GPS navigation unit. Modern smartphones include all of these features plus the features of a laptop, including web browsing, Wi-Fi, and third party apps and accessories ,multiple microprocessor core and gigabyte RAMs. The most popular smartphones today are powered by Google's Android and Apple's IOS mobile operating systems and the wide deployment of 3G broadband cellular networks. The combination of cloud computing and mobile networks to bring benefits for mobile users, network operators, as well as cloud computing providers The design of mobile social TV system, CloudMoV(Cloud Mobile Social TV), which can effectively utilize the cloud computing to offer a living-room experience of video watching by mobile users with spontaneous social interactions. In cloud mobile social TV, mobile users can import a live or on-demand video to watch from any video streaming site and invite their friends to watch the video concurrently, and chat with their friends while enjoying the video. It therefore blends viewing experience and social awareness among friends on the move.**

***Keywords*-Computers and Information processing, Mobile computing, Communications technology, Mobile TV.**

I. INTRODUCTION

There has been an rapid reinvention in the smartphones[30] by combining the features of a mobile phone with those of another popular consumer device, such as a personal digital assistant (PDA), a media player, a digital camera, and/or a GPS navigation unit. Modern smartphones include all of those features plus the features of a laptop, including web browsing, Wi-Fi, and third party applications and accessories, multiple microprocessor core and gigabyte RAMs. The most popular smartphones today are powered by Google's Android and Apple's iOS and the wide deployment of 3G broadband cellular networks. Many mobile social or media applications have emerged, truly gaining mass acceptance and are still impeded by the limitations of the current mobile and wireless technologies, among which battery lifetime and unstable connection bandwidth are the most difficult ones.

It is natural to resort to cloud computing, the newly-emerged computing paradigm for low-cost, and scalable resource supply, to support power-efficient mobile data communication with virtually infinite hardware and software resources. The cloud can offload the computation and other tasks involved in a mobile application and may significantly reduce battery consumption at the mobile devices. The big challenge is how to effectively exploit cloud services to facilitate mobile applications. There have been a few studies on designing mobile cloud computing systems [1][2][3], but none dealt in particular with stringent delay requirements for spontaneous social interactivity among mobile users. Mobile/cloud computing [20] is the combination of cloud computing and mobile networks to bring benefits for mobile users, network operators, as well as cloud computing providers. The design of mobile social TV system, which can effectively utilize the cloud computing paradigm to offer a living-room experience of video watching by mobile users with spontaneous social interactions. In Cloud mobile social TV, mobile users can import a live or on-demand video to watch from any video streaming site, invite their friends to watch the video concurrently, and chat with their friends while enjoying the video. It therefore blends viewing experience and social awareness among friends on the go. As opposed to traditional TV watching, mobile social TV is well suited to today's life style, where family and friends may be separated geographically but hope to share a co-viewing experience. While social TV enabled by set-top boxes over the traditional TV systems is already available [4][5], it remains a challenge to achieve mobile social TV, where the concurrently viewing experience with friends is enabled on mobile devices. The design Cloud mobile social TV to seamlessly utilize resource support and rich functionalities offered by both an IaaS (Infrastructure-as-a-Service) cloud and a PaaS (Platform-as-a- Service)[20] cloud. The mobile users can import a live or on-demand video to watch from any video streaming site and invite their friends to watch the video concurrently, and chat with their friends while enjoying the video. It therefore blends viewing experience and social awareness among friends on the move. This design achieves the following goals.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

A. Encoding Flexibility

Now a day's we have different sized displays, customized hardwares for different mobiles. Each different mobile will have different codecs. Traditional systems would adopt few encoding formats for video processing but the problem lies with generous content providers that will not support all different mobile coding formats. Our model uses the generic streams of video for different devices at real time, by offloading the transcoding tasks to an IaaS cloud. In particular, we employ a surrogate (Virtual machine) for each user, which is a virtual machine (VM) in the IaaS cloud. The surrogate downloads the video on behalf of the user and transcodes it into the desired formats of the mobile device.

B. Battery Efficiency

A major analysis conducted by Carroll et al. [6] defines the significant portion of power consumption is due to network modules (both Wifi and 3G) and the display of mobile devices which affects the usages from other hardware modules including CPU , memory, etc. We in our system focus at energy. Saving which is consumed by different network modules of smart phones through an efficient data transmission mechanism design. We focus on 3G wireless networking as it is getting more widely used and challenging in our design than Wi-Fi based transmissions. Based on cellular network traces from real-world 3G carriers, we investigate the key 3G configuration parameters such as the power state. We use burst transmission mechanism for streaming from the surrogate to the mobile devices. The burst transmission mechanism helps in making decisions on multiple burst sizes and opportunistic transitions among high/low power consumption modes at the devices which eventually results in eventually better energy consumption.

C. Spontaneous Social Interactivity

Multiple mechanisms to provide spontaneous social Co-viewing experience. We are enabling the synchronization mechanism to guarantee that the people who join in video program may watch the same portion (if it's opted/chose by people engaged in video program) and share immediate reactions and comments. We have a synchronized playback which is inherently a feature of traditional TV, the current Internet video services rarely offer such a service. We adopt textual chat messages rather than voice in our current design, believing that text chats are less distractive to viewers and easier to read/write and manage by any user.

These mechanisms are seamlessly integrated with functionalities provided by a typical PaaS cloud, via an efficient design of data storage i.e by including the database in cloud and handling concurrent messages dynamically. We exploit a PaaS cloud for social interaction support due to its provision of robust underlying platforms (other than simply hardware resources provided by an IaaS cloud). The above mechanism provides the transparent, automatic scaling of users applications onto the cloud.

D. Portability

In our system both front end user interaction modules and backend server side programming are implemented '100% pure java', which enables the main feature of jave i.e "Write Once, Run Anywhere"(WORA). the only exception is the transcoding module, which is implemented using ANSI C for performance reasons and uses no platform-dependent or proprietary APIs. The end user module can run on any mobile devices supporting HTML5, including Android phones etc. To showcase its performance, we deploy the system on cloud for performance test. Our system can be easily deployed in to different clouds with ease and can be used by the mobile client users at anytime and anywhere.

II. Related Work

There has been a number of mobile TV systems that have sprung up in recent years, which advances in mobile devices both hardware and software . Some early systems [8], [9] bring the "living-room" experience to small screens on the move. But they focus more on barrier clearance in order to realize the convergence of the television network and the mobile network, than exploring the demand of "social" interactions among mobile users. There is another trend in which efforts are dedicated to extending social elements to television systems [4], [5], [10]. Coppens *et al.* [4] try to add rich social interactions to TV but their design is limited to traditional broadcast program channels. Oehllberg *et al.* [5] conduct a series of experiments on human social activities while watching different kinds of programs. Though inspiring, these designs are not that suitable for being applied directly in a mobile environment. *Chuah et al.* [11] extend the social experiences of viewing traditional broadcast programs to mobile devices, but have yet to deliver a well-integrated framework. Schatz *et al.* [12], [13] have designed a mobile social TV system, which is customized for DVB-H networks and Symbian devices as opposed to a wider audience.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

Compared to these prior work and systems, we target at a design for a generic, portable mobile social TV framework, featuring co-viewing experiences among friends over geographical separations through mobile devices. Our framework is open to all Internet-based video programs, either live or on-demand, and supports a wide range of devices with HTML5 compatible browsers installed, without any other mandatory component on the devices. For any application targeted at mobile devices, reducing power consumption is perennially one of the major concerns and challenges. In addition, we employ a surrogate for each mobile user in the cloud rather than relying on a dedicated cluster, which can be more easily implemented in practice. Liu *et al.* [14] build a mobile-based social interaction framework on top of the Google App Engine and offer an iOS implementation. We set out to design a portable, generic, and robust framework to enable real-time streaming and social interaction concurrently, which is not bound to any specific cloud platform. Although our prototype is implemented on only two public clouds, *i.e.*, Rackspace and Google App Engine, it can be easily ported to other cloud systems as long as the targeted cloud platforms conform to the unified standard. A recent work by Zhang *et al.* [15] investigates the media caching management problem under HTTP adaptive bit rate streaming over a wireless network environment, which can complement our work when video streams are required to the video converter into multiple bit rates.

Finally, we are aware of the lack of a richly-featured cloud based mobile social TV system in real life. The only system coming close to ours is Live Stream [16] on the iOS platform. Conversely, the prototype we implement is browser-based and platform independent, it supports both live channels, VoD channels and even personal channels hosted by any user, with wider usage ranges and flexible extensibility. The framework can be readily applied to other cloud-assisted mobile media applications as well.

III. Arcitecture And Design

As cloud based mobile social TV system provides the two major functionalities for mobile users: (1) Universal streaming. A user can stream a live demand video from any video sources he chooses, such as a TV program provider or an Internet video streaming site, with tailored encoding formats and rates for the device each time. (2) Co-viewing with social exchanges. A user can invite multiple friends to watch the same video, and exchange text messages while watching. The group of friends watching the same video is referred to as a *session*. The mobile user who initiates a session is the *host* of the session.

We present the architecture of *Cloud Mobile Social TV* and the detailed designs of the different modules in the following.

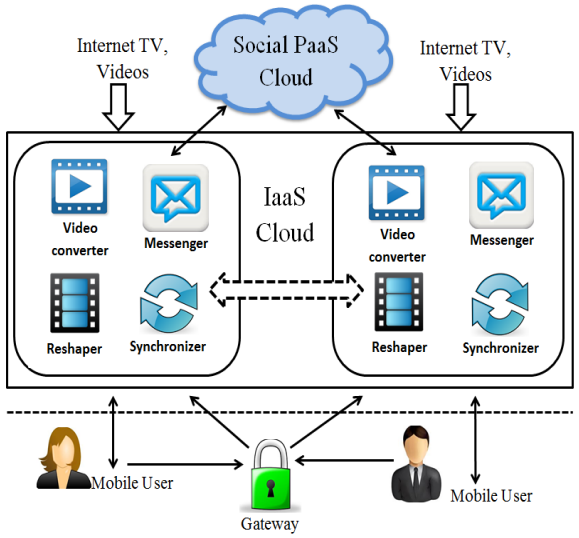

Fig. 1. The architecture of cloud mobile social TV

A. Key Modules

Fig. 1 gives an overview of the architecture of *Cloud mobile social TV* A *surrogate* (*i.e.*, a virtual machine (VM) instance), is created for each client end mobile user in an IaaS cloud infrastructure. The Virtual machine acts as a proxy between the mobile device and the video sources in the cloud which provides multiple transcoding service. Transcoding helps in segmenting the video streaming traffic. We have also used the burst transmission to the end user. Virtual Machines plays a key role in handling the social messages exchanged between the ends users. The VM exchange social messages via a back-end PaaS cloud by which we can have huge scalability which results in better robustness of the system. There is a Third party auditor module in Cloud *server* that keeps track of participating users and their VM surrogates. The design of cloud based mobile social TV can be divided into the following functional modules.

1) Video Converter: Its function part which resides in each virtual machine for each end user, the main functionality of the video converter is to decide how to encode the video stream from the video source in the appropriate format, dimension, and bit rate dynamically. It is necessary to encapsulate the video in to proper transport stream before delivering the video to the user.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

In our implementation we export the video in to MPEG-2 transport streams, which is default standard being used to transferring the digital video and audio stream over a lossy medium.

2) Reshaper: After segmenting the encode video we need to reshape the video packet (segments) in to proper order. The job of the reshaper is to receive the video from the video converter and to split the video in to multiple packets as video segments and then send each video segment in a burst transmission way as requested by the user. Burst transmission is decided on the burst size, *i.e.*, the amount of data in each burst, is carefully decided according to the 3G technologies implemented by the corresponding carrier. Despite as a proof of concept we are implementing this module in static size , Later it can be easily changed to any 3G transmission burst size as per 3G carrier.

3) Social Cloud: We are using the social cloud which is built on top of PaaS cloud services with data store. This is to get the better performance over the different specific proprietary platforms. As per proof of concept our system can be readily placed on cloud services like we are using free cloud services like Rackspace.

4) Messenger: In the end user side the social cloud implementation has Messenger module which resides in the IAAS of the cloud. The main functionality of the messenger is to query the social cloud for the social data like messages, comments etc. The Messages are pre-processed in light-weighted format (plain text files), at a much lower frequency. The messages are transferred asynchronous over the lower traffic. On the other hand the messenger sends the user's messages like(chat messages) to the other users via the data store of the social cloud.

5) Synchronizer: The synchronizer on a surrogate is important to synchronize the video as per previously viewed playback segment within a time window of other users in the same session (if the user chooses to synchronize with others). To achieve this, the synchronizer periodically retrieves the current playback progress of the session host and instructs its mobile user to adjust its playback position. In this way, friends can enjoy the "sitting together" viewing experience.

6) Mobile Client: The mobile client is not required to install any specific client software in order to use *cloud mobile social TV*, as long as it has an HTML5 compatible browser and supports HTTP Live Streaming protocol[24].

7) Gateway: The gateway provides secure authentication services for users to log in to the *cloud mobile social TV*. The gateway stores users' credentials in a permanent table of a MySQL database it has installed. After a user successfully logs into the system.

C. Pipelined Video Processing

This module of the surrogate is indeed needed to have a feel of realtime live streaming of video and to have on-demand streaming of stored contents of video in the system. The key designs in cloud based mobile social TV are as follows.

B. Burst Transmissions

1) 3G Power States: Different from Wi-Fi which is more similar to the LANed Internet access, 3G cellular services suffer from the limited radio resources, and therefore each user equipment(UE) needs to be regulated by a Radio Resource Control (RRC) state machine [19]. Different 3G carriers may customize and deploy complex states in their respective cellular networks. A 3G carrier may commonly transfer a UE from a high-power state to a low power state (state demotion), for releasing radio channels allocated to this UE to other users. For example, if a UE working at a high-power state does not incur any data traffic for a pre-configured period of time (measured by a critical inactivity timer), the state of the UE will be transferred to a low-power one; when the volume of data traffic rises, the UE "wakes up" from a low-power state and moves to a high-power one. Timeouts of the critical inactivity timers for state transitions are properly set by the carrier to guarantee performance in both delay and energy consumption, since extra delay and energy consumption are potentially incurred for acquiring new radio channels when the UE transits from a low-power state to a high-power one later (state promotion).

2) Transmission Mechanism: Using the HTTP live streaming protocol [18], the mobile device sends out requests for the next segment of the video stream from time to time. The surrogate divides the video into segments, and sends each segment in a burst transmission to the mobile device, upon such a request.

3) Burst Size: To decide the burst size, i.e., the size of the segment transmitted in one burst, we need to take into consideration characteristics of mobile streaming and energy consumption during state transitions.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

For video streaming using a fixed device without power concerns, it is desirable to download as much of the connection bandwidth allows; however, for streaming over a cellular network, we should a video as what avoid downloading more than what is being watched for one main reason: users may switch among channels from time to time and those prefetched contents are probably never watched, leading to a waste of the battery power and the cellular data fee due to their download. Hence, the bursty size should be kept small, to minimize battery consumption and traffic charges and this in return helps saving the battery lifetime.

IV. Prototype Implementation

Following the design guidelines in Section III, since our implementation is done on Java platform, we can deploy our system in Google App Engine (GAE) [As a matter of choice] and Rackspace (freely available cloud service) which are most commonly used PaaS and IaaS platforms respectively. GAE, as a PaaS cloud, provides rich services on top of Google's data centers and enables rapid deployment of Java-based and Python-based applications. Hence, GAE is an ideal platform for implementing our social cloud, which dynamically handles large volumes of messages. On the other hand, GAE imposes many constraints on application deployment, *example* , lack of support for multi-threading, file storage, etc.,

Rackspace is a representative IaaS cloud, offering raw hardware resources including CPU, storage, and networks to users. Rackspace has two main service-level segments: Managed and Intensive. Both service levels receive support via e-mail, telephone, live chat, and ticket systems, but they are designed to fit the needs of different businesses. The Managed support level consists of "on-demand" support where proactive services are provided, but the customer can contact Rackspace when they need additional assistance. The Intensive support level consists of "proactive" support where many proactive services are provided, and customers receive additional consultations about their server configuration. Highly customized implementations generally fall under this level of support.

A. Client Use of Cloud Mobile TV

Android is used for programming for the client mobile devices. Our Cloud based Mobile social TV is installed with HTML5 compatible browsers can use *Cloud based mobile TV* services, as long as the HTTP Live Streaming (HLS) [24] protocol is supported, for achieving this have used Used the Http servelet objects for the interface between the data owner and cloud system.

The user first connects to the login page of *application*, after the user successfully log in through the gateway (Third Party Auditor), User is assigned a VM surrogate from the VM pool (Multi-threading) user is automatically redirected to the assigned VM surrogate, and welcomed by a portal page.

The user can enter the filename of the video which downloads the stream on the user's behalf, converted video and sends properly encoded segments to the user. From the surrogate to the mobile device, the video stream delivered using HLS is always divided into multiple segments, with a playlist file giving the indices. The client starts to play the video as soon as the first segment is received. When watching a video, the user can check for their friends' messages and invite them to join in watching the video. Users in the same session can exchange opinions and comments on the "Chat" tab where new chat messages can be entered and the chat history of the session is shown.

B. VM Surrogates

All the VM surrogates are provisioned from Rackspace web services and tracked by the gateway. We have also installed a Tomcat web server (version 6.5) to serve as a Servlet container and a file server on each Surrogate and process the video stream by video converting and segmentation. For example, in our experiments, since we are working in better speed of internet we have excluded the different streaming part dynamically, but we have the proposed system to implement high-quality stream to have "480 x 272" resolution with 24 frames per second, while the low-quality one has a "240 x 136" resolution with 10 frames per second. The transcoded stream is further exported to an MPEG-2 transporting stream (.ts), which is segmented for burst transmission to the user.

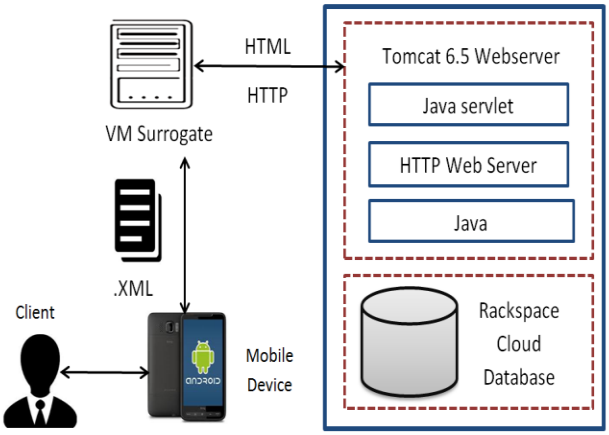

Fig. 2. Social message exchange via Google App Engine

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014)

C. Data Models in the Social Cloud

Google App Engine is mainly used as the back-end data store keep online presence status, social messages (invitation and chat messages) in all the sessions shown in Fig. 6. With Jetty as the underlying Servlet container, most Java-based applications can be easily migrated to GAE, under limited usage constraints, where no platform-specific APIs are enforced for the deployment.

GAE provides both can be easily migrated to other PaaS clouds as well. If the user wishes to synchronize his playback progress with that of the session host, his VM surrogate synchronizes with the session host to maintain the playback "current time" value (HTML5 property).

The social cloud maintains a "Logs" entry for each existing session in *Cloud based mobile system TV* with the session ID as the primary key and an array list as the value, which corresponds to individual messages in this session.

When a user in a session posts a comment, this message is first sent to his VM surrogate, which further injects the message into the social cloud via another Servlet listener. The message is stored as a "Message" entry in the social cloud, with the message content as the value, and an auto-generated integer as the key. this message can then be viewed by the client. the user can also reply to the messages that has been received, hence this leads to a chat or and interaction wich is socially among the users using the cloud mobile TV.

V. CONCLUSION

The mobile TV might become an emerging trend, which is mobile social TV based on cloud computing .The resource and rich functionalities of cloud computing services in combination with the mobile networks. Introducing a portable mobile social TV framework, cloud mobile social TV that makes use of both an IaaS cloud and a PaaS cloud with the video streaming on the mobile devices. The framework provides efficient video converting services for most platforms under various network conditions and supports for co-viewing experiences through timely chat exchanges among the viewing users. By employing one surrogate for each mobile user, achieve ultimate scalability of the system, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel Internet regardless of environments and platforms .The power states in commercial 3G cellular networks, this propose an energy-efficient burst transmission mechanism that can effectively increase the battery lifetime of user devices.

The experiments also compares the HTTP Live Streaming protocol (it works by breaking the overall stream into a sequence of small HTTP-based file downloads, each download loading one short chunk of an overall potentially unbounded transport stream) implementation on mobile devices [24] as compared to the burst transmission mechanism (any relatively high-bandwidth transmission over a short period ,for example, a download might use 2 Mbit/s on average, while having "peaks" bursting up to, say, 2.4 Mbit/s)which increases of battery lifetime. Much more, however, can be done to enhance cloud mobile social TV to have product-level performance. This prototype, do not enable sharing of encoded streams (in the same format/bit rate) among surrogates of different users.

In the future work, such sharing can be enabled and carried out in a peer-to-peer fashion, e.g., the surrogate of a newly joined user may fetch the transcoded streams directly from other surrogates, if encoded in the format/bit rate that the new user wants.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based Cloudlets in mobile computing," IEEE Pervasive Comput., vol. 8, pp. 14–23, 2009.

[2] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in Proc. IEEE INFOCOM, 2012.

[3] Z. Huang, C. Mei, L. E. Li, and T. Woo, "Cloudstream: Delivering high-quality streaming videos through a cloud-based SVC proxy," in Proc. INFOCOM'11, 2011, pp. 201–205.W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[4] T. Coppens, L. Trappeniners, and M. Godon, "AmigoTV: Towards a social TV experience," in Proc. EuroITV, 2004.

[5] N. Ducheneaut, R. J. Moore, L. Oehlberg, J. D. Thornton, and E.Nickell, "Social TV: Designing for distributed, sociable television viewing," Int. J. Human-Comput. Interaction, vol. 24, no. 2, pp.136–154, 2008.

[6] A. Carroll and G. Heiser, "An analysis of power consumption in as smartphone," in Proc. USENIXATC, 2010.

[7] What is 100% Pure Java. [Online]. Available: http://www.javacoffeebreak. com/faq/faq0006.html.

[8] J. Santos, D. Gomes, S. Sargento, R. L. Aguiar, N. Baker, M. Zafar, and A. Ikram, "Multicast/broadcast network convergence in next generation mobile networks," Comput. Netw., vol. 52, pp. 228–247, Jan. 2008.

[9] DVB-H. [Online]. Available: http://www.dvb-h.org/.

[10] K. Chorianopoulos and G. Lekakos, "Introduction to social TV: Enhancing the shared experience with interactive TV," Int. J. Human- Comput. Interaction, vol. 24, no. 2, pp. 113–120, 2008.

[11] M. Chuah, "Reality instant messaging: Injecting a dose of reality into online chat," in CHI '03 Extended Abstracts on Human Factors in Computing Syst., 2003, ser. CHI EA '03, pp. 926–927.

[12] R. Schatz, S. Wagner, S. Egger, and N. Jordan, "Mobile TV becomes social – Integrating content with communications," in Proc. ITI, 2007.926–927.

[13] Z. Liu, Y. Feng, and B. Li, "Socialize spontaneously with mobile applications,"in Proc. IEEE INFOCOM, 2012.

[14] R. W. Lucky, "Automatic equalization for digital communication," Bell Syst. Tech. J., vol. 44, no. 4, pp. 547–588, Apr. 1965.

[15] W. Zhang, Y.Wen, Z. Chen, and A. Khisti, "Qoe-driven cache management for http adaptive bit rate (abr) streaming over wireless networks," in Proc. IEEE Globecom, 2012.

[16] Livestream[Online].Available:http://itunes.apple.com/us/app/livestream/id379623629?mt=8/.

[17] NoSQL.Data.Base.[Online].Available:http://nosqldatabase.org/.

[18] HTTP.Live.Streaming.[Online].Available:http://tools.ietf.org/html/draft-pantos-http-live-streaming-01.

[19] http://en.wikipedia.org/wiki/cloud computing.

[20] http://en.wikipedia.org/wiki/Smartphones.