Architectural and Qos Issues in Mobile Cloud Computing Environment for Real-Time Video Streaming

Ashish Kumar chakraverti[1]
Research Scholar: Department of computer science and Engineering
IKG Punjab Technical University
Kapurthala, Jalandhar, Punjab, India

Dr. Vijay Dhir[2]
Director: Engineering
(SSIET) St. Soldier Group of Institution
Jalandhar-Amritsar Bye Pass, NH-1, Behind NIT, Jalandhar Punjab, India

Dr. Sugandha Chakraverti[3]
Former Faculty: Department of computer science and Engineering
IEC College of Engineering and Technology
Greater Noida Uttar Pradesh, India

***Abstract*—this paper state the issue related to mobile cloud computing for real-time video streaming. Recently the media streaming particularly video streaming becoming very popular among the users of hand held devices. Due to exponential growth in mobile device users, mobile network is facing large traffic of video signals. Due to this growth of users, video streaming especially real-time video streaming is now an interesting area of research. To increase the streaming speed and quality of real-time video a lot of work and progress is being done in hardware architecture of mobile devices. We have multi-core processors and enough memory space in modern mobile devices. The concept of GPU is also introduced in mobile devices to overcome the problem of speed in video data processing. But very less work is done to utilize the capability of these devices through parallel architecture development for real-time video streaming process. So it is the matter of discussion and research that how the computational and memory power of these devices can be utilized through parallel computations to speed up the process of video data up to ultra high speed. Mobile cloud computing environment can play an important role in this field. So In this paper, the review of architectural and QoS issues in mobile cloud computing environment for real time video steaming are being presented. There are many real-time video streaming techniques for mobile devices, some possible parallel architecture and QoS (Quality of Services) issue to leverage the architecture capability of the mobile cloud computing environment for these real-time video streaming is being discussed in this paper.**

***Keywords*—QoS; Video streaming; Mobile cloud computing; A mobile device; GPU; Parallel algorithms**

I. Introduction

In few next year's RTVS (real-time video streaming) in mobile devices will be a common and generalized application. RTVS in mobile devices may be improved by using MCC (mobile cloud computing) environment. MCC is the area in which mobile computing and cloud computing concepts and benefits are integrated to get better performance in mobile devices. Due to exponential growth in handheld device users and betterment in telecommunication services RTVS in handheld devices (Especially in Smartphone) is going to be a very common and usual application. Smartphone and tablets have cellular, wifi and Bluetooth interfaces facilities. Task offloading from Smartphone to the cloud is a promising strategy to enhance the computing capability of Smartphone and prolong their battery life [14]. Utilizing multiple links parallel can improve video streaming in several aspects [15]. Now day's Smartphone are also equipped with multiprocessor systems; that can also be utilized for parallelization of streaming process.

In this progress, some parallel algorithms for processing of video streaming in different steps have been proposed. One of the techniques is motion estimation. Accurate motion estimation between frames is crucial for drastically reducing data redundancy in video coding. However, advanced motion estimation methods are computationally intensive and their execution in real-time usually requires parallel implementation [4]. Similarly, many techniques are used for RTVS process enhancement. But rare work is done via MCC in this field. Since RTVS play an effective role in education, public safety, health care, real estate industry, etc. so there is a need to have study about the improvement possibilities in this field.

This paper is aimed to address the issue related to RTVS over MCC environment. The issues discussed in this paper are the standardization of the architecture of MCC, operations over MCC, QoS of MCC, operations in MCC and economics of MCC services. The recent progress of digital media has stimulated the criterion, storage, and distribution of data requiring efficient technologies to increase the usability of these data. Video summarization methods generate concise summaries of video contents and enable faster browsing, indexing and accessing of the large video collection. However, these methods often perform slowly with large and high-quality video data. One way to reduce this long time of execution is to develop a parallel algorithm, using the advantage of the modern computer architectures that allow high parallelism [21].

MCC has some feasible solution to the inherited limitation of mobile computing. These limitations include battery lifetime, processing power and storage capacity. By using MCC, the processing and the storage of intensive mobile device job will take place in the cloud system, and the result will be given to the mobile device. This method of computation needs less power and time to compute intensive jobs like real-time videos streaming [28].

The remaining part of this paper has five more section. In section II, this paper elaborate MCC architecture which may support the RTVS. In section III, paper explores different video streaming algorithm and technique either based on simple or MCC architecture. This paper also discusses possible future work, findings, and improvements given in many recent research articles and papers in section IV. In section V, this article discusses the social and legal issue related to RTVS and how it is beneficial for health sector, education, social connectivity and security. Finally, this paper concluded the discussion in section VI.

II. Mobile Cloud Computing Architecture for Real-Time Video Streaming

MCC is born to leverage power full computing and storage resource in the cloud to provide abundant services in mobile environment conveniently and ubiquitously. The feature of MCC includes no upfront investment, lower operating cost, highly scalable and easy access etc [26].

The major benefit of cloud computing for mobile devices is ability to run applications between resource constraint devices and internet based cloud. Hence resource constrained device can outsource computation, communication and resource intensive operations to the cloud [24].

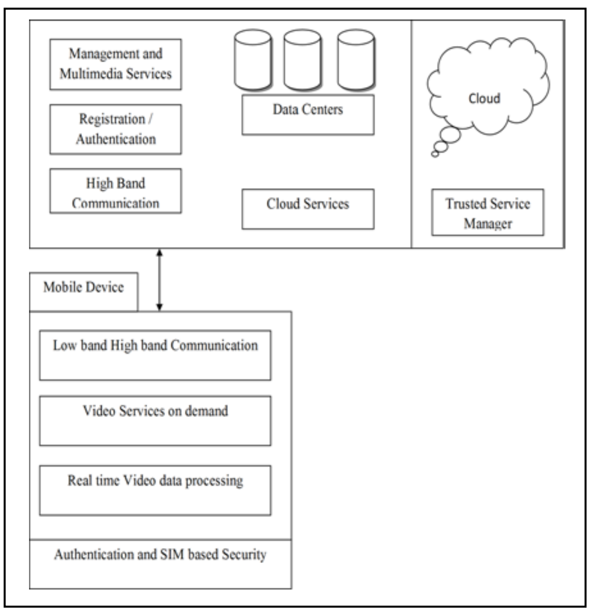

Fig. 1. Service Architecture for real-time video processing over mobile cloud computing environment

A. System Modeling

Here paper is presenting a generalized architecture for real-time video processing that relies on mobile and cloud computing models. By developer point of view following inherited requirement must be supported by architecture.

- Method of video data communication
- Method of security to have secure communication which control the cloud access
- Device/Resource registration used to access mobile cloud system by new users
- Environment constraints which effect the execution of video data processing that is performed in mobile cloud and mobile device

In 2014-2015 the use of Smartphone has changed dramatically. For example now days mobile user expect to have full access to their own data and also have desire to use all kind of application on their mobile devices specially Smartphone. Beside significant evolution in mobile devices, these are still and will always be limited in terms of processing power, storage, band width and energy. They also face the problem of less reliable connection less secure communication in compression of stationary devices.

In order to face these limitations the new generation of mobile applications already relies on, e.g. cloud augmentation as supported by current cellular network standards (3G, LTE) which allows overcoming such intrinsic restriction of mobile devices [9].

Since video processing application typically comprise computational and data intensive tasks, so MCC can be used to speed up the process. In fig 1, MCC architecture for real-time video streaming is given. Such type of cloud augmented application may be advertizing, video gaming, online classes and recognition assistant.

B. MCC Architecture

In general scenario mobile applications just rely on the mobile device capabilities, but some mobile applications may require the cloud resources to provide their full functionality. Such type of applications is called MCC-application. RTVS is one of those type of application, that may be executed entirely on the mobile device, but are able improve their performance by offloading certain parts onto cloud resources.

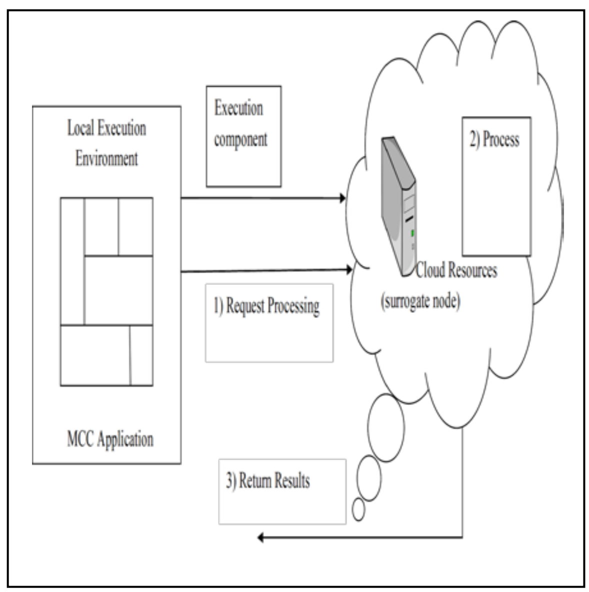

Fig. 2. Offloading execution component to cloud resources (surrogate nodes)

Fig 2 shows a generalized offloading task in which some part of the application's code are processed on a cloud resource which may be called surrogate node and the computed results are given back to the mobile device. There are following common component of MCC which enables this type of computation offloading.

- A partitioner
- Context monitor
- Solver
- Coordinator

Partitioner analyzes the application and determines the part of application's code which has candidature of offloading. Context monitor check the context information like available surrogate node, battery status and network connectivity then this information is used by solver which allocate the surrogate node to application's code . Coordinator perform additional task like discovery, authentication and synchronization.

C. *Essential Requirement of MCC Architecture*

From section II.A and II.B it is clear that MCC application development is not easy as simple mobile application. Thus it needs proper support to ease the development. Inter process communication, remote method invocation or service invocation models cannot be employed as it is in MCC.

Along with all the problem of distributed system like heterogeneity, security, scalability failure handling, concurrency, transparency and quality service some other problem are also exist in MCC, e.g. limited resource etc, these problem must be in consideration at the time of development of MCC applications. Additionally MCC applications should also support reliability, usability, efficiency, maintainability and portability as define in ISO/IEC 2501n. So here we are discussing combined general requirements of this category.

A good MCC application must support following criteria.

1) *Availability*
2) *Scalability*
3) *Usability*
4) *Maintainability*
5) *Security*

Availability: - To perform a computation in MCC environment a reliable connection must be available to all surrogate nodes. This availability must have low latency and high bandwidth.

Portability: - This is an important function in case of MCC. Since there is frequent transfer of application computation between mobile and surrogate. This shifting is dynamic in nature.

Scalability:- Application which run on heterogeneous and changing environment required dynamic partitioning. Due to remote execution of partitioned task and return of results to mobile need extra overhead in term of computation and bandwidth. MCC architecture assumes that cloud resource is infinite but this is not the case in practical. Since MCC need surrogate and task are being executed parallel so every partitioned task is assigned a surrogate. This generally require a concurrency handling issue that affect the scalability of solution

Usability: - In MCC there are many issues which affect its usability, like energy issue, bandwidth issue, end user complexity etc. So after considering these issue MCC application should be easy and intuitive. Additionally an adequate abstraction should be there to hide the complexity of implementation of MCC. Adequate abstraction mean without compromising the full potential of MCC application.

Maintainability: - Software development for mobile devices now days is extremely agile and repetitive task. To support this task up to satisfactory level, it is required that the integration of MCC feature does not lead to increase complexity. Common development also suggests ease of bug fixing. Finally the architecture should be depending on open standards so vendor lock-in may be prevented with above facility and maintainability can be achieved up to adequate level.

Security: - Multi tenancy, concurrency and distribution are main feature of any cloud computing architecture. But due to these feature architecture compromises with security. So it is a never ending issue to have secure cloud architecture. Transferring confidential or private data to cloud node (surrogate node) always signals the risk of losing control over confidential or private data. Thus an easy to use solution/tool is requiring which may categorize and prevent the sensitive information to leave the trusted device (Mobile).

D. *Some existing solutions for the essential requirement of MCC*

MCC have special kind of requirement we focus on these requirement and here discussing some existed solution for these requirements. After doing the extensive survey we have classified existing solution according to their nature in terms of performing the computation offloading.

Existing solutions are categorized in six domains

1) *Specific language*
2) *Middle ware and frame work*
3) *Distributed virtual machine*
4) *Universal computable solution*
5) *Native(non VM-base) MCC solution*
6) *Native(VM based) MCC solution*

Specific language:- For the purpose of programming there are many programming languages for e.g. domain specific or for specific scenarios, MCC environment is different kind of environment which need some specific language i.e. developer of MCC application must know new language syntax and programming style and a mobile operating system to support the execution of languages.

Middleware and Framework: - Opposite to specific languages framework provide the facility of easy development. In this programmer/developer is given a framework where he/she just fit the application logic. The frame work provides the full control over execution and calls the proper application specific code.

Distributed VM:- Previously discussed solution categories need the partitioning of application. This must be done by developer explicitly. There are some methods to automate this task, one of the best example of this type of automation is distributed VM. These VM handle the application for their distribution, partitioning, concurrency, synchronization and also preserve global state of application. To do these tasks different VMs run on may node of designated network.

Universal computable solution:- These are some other solution in the area of universal applications. Some system level working models for this are like Vivendi/chrome and gaia. These approaches take care of related data discovery, resource discovery, data distribution, distributed execution and automatic partitioning of applications. With the help of these solutions, in design phase the distribution scenario of application can be dealt efficiently. By this developer is not requiring having panic about these issues in lower level implementation.

Native (Nov-VM Based) MCC solution: - these are basically two type of solutions, VM based and non-VM based, which target MCC applications. In non-VM model idea is to offload some of the application’s tasks on to other devices so master can be unburdening from intensive computational tasks. To accomplish this task, applications are analyzed internally and some development strategies are created which can be applied upon the executing applications.

Native (VM based) MCC solution: - When we need higher degree of distribution transparency we use VM based approaches. Clone cloud is a simulation which used a complete image of a mobile device. This image runs in a VM on a server to execute a part of application. This may also be decided that which threads to offload using a profiler at run time.

E. Application models of MCC

The mobile cloud application models are designed to achieve a particular objective, such as executing applications that have in sufficient resources, for local execution, enhancing applications performance (in terms of computation time), or achieving energy efficiency on mobile devices [2].

Application models must be adopted considered the objectives and their effect on the counterparts [2].

Mobile cloud computing applications are classified in to four categories.

1) *Performance based application models.*
2) *Energy based application models.*
3) *Constrained based application models.*
4) *Multi Objective application models.*

Performance based application models: - Main objective of this model is to improve the performance of mobile device application using cloud resources. Computation which have the intensive requirement of resource are offloaded to the high speed clod, where within very less time the computation will be performed in compare of mobile devices. These applications execute in mobile devices with improved performance with the help of cloud resources.

Energy based application models: - This model is design to save the energy consumption in mobile devices. Intensive resource computation are offloaded in cloud resources so mobile device does not waste it power in computation of such applications. By these models we reduce the computational overhead of mobile device.

Constrain based application models: - Mobile device has some resource constraint environment (e.g. smart phones) like small memory, small battery, slow processor etc. Thus these models are designed to execute the application for which resources are not enough in mobile device. In these models light weight process/applications are run on mobile devices and resource intensive applications are offloaded to cloud where these are executed. Un doughtily these models are useful to execute high resource demanding application on resource constraint devices.

Multi Objective application models: - Some time we require achieving multiple objectives from same models. In this case these types of models are used, for example if we need performance and efficiency at same time then this model is most suitable. More than one objective can be achieved with a fair tradeoff between require objective.

F. Video streaming technique over MCC

In streaming procedure, video clip data file is sent to the end individual in a (more or less) continuous flow. It is simply a strategy for shifting information such that it can be prepared as stable and ongoing flow and it is known as streaming or encoded movie that is sent across information system is known as streaming.

Streaming movie is a series of “moving images” that are sent in compacted form over the internet and shown by the audience as they appear [22].

Video streaming can be elaborated in detail by following points.

1) *Streaming principle*
2) *Video streaming architecture*
3) *Video streaming technique*

Streaming principle: - in case of RTVS media packets should arrive in a timely manner, since delayed packets are treated as lost. In streaming process it is essential for the media packet to reach their location in continuous and regular basis, because the network blockage can be appeared due to extreme wait of the delayed packet. Quality of information may be compromised due to this extreme wait. Beside this synchronization between customer and host server packet may be damaged and mistakes to distribute in the provided movie.

There are two type of streaming

a) *Real-time streaming*

b) *Pre recorded streaming*

The protocol used for streaming purpose is UDP (User data gram protocol)

Video streaming architecture: - Movie streaming scheme based on MCC is represented here

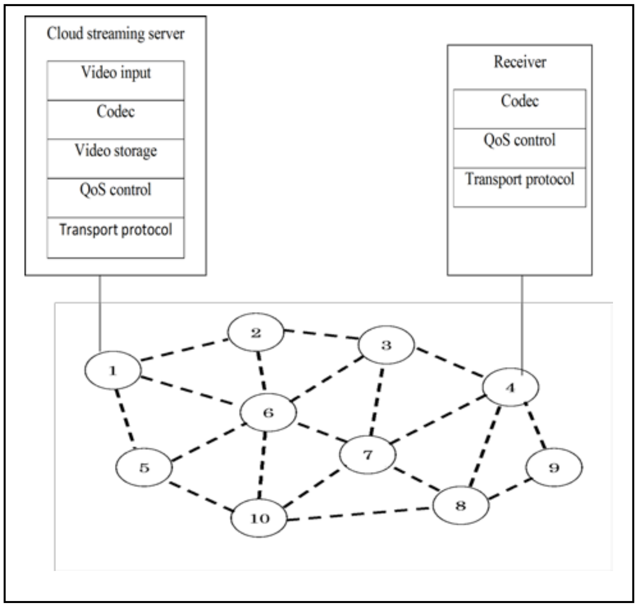

Fig. 3. Video streaming Architecture

In the given architecture there is a streaming host server implemented by cloud based source. This streaming host server is responsible for sending, retrieving and adapting video clip flow. Video clip may online protected for real-time broad casting or pre coded and stored for broad casting on demand, depend on application.

Interactive movie, mobile movie steaming, interactive online game and live broad cast require real-time encoding. But movie on demand like application require pre coding / pre encoded movie.

Video streaming technique: - There are various streaming techniques for differing mobiles devices, smart phones describe bellow [22].

a) Progressive download

b) HTTP live streaming

Progressive download- There are some options to customer to down load a pre coded press data file partitioned in the appropriate codec for the product to play for this frequently used method is HTTP or HTTPS. In this method play back starts just after gradual down load of preserved file. After this by the qualification player is constantly on the downloaded the rest of the material.

HTTP Live streaming: - HTTP live streaming (also known as HLS) is an HTTP based media steaming communication protocol implemented by apple Inc. as part of their QuikTime x and iPhone. Apple's HTTP live streaming protocol (HLS) is an adaptive streaming video delivery protocol for iOS devices. It utilizes the H.264 video codec, which is segmented and encapsulated in MPEG2 transport streams and .M3UP index files to deliver live and on demand video. The device automatically select the most appropriate stream given available bandwidth, CPU and platform constraints, downloads a manifest for that stream and then downloads segmented chunks to the buffer for the playback[22].

Users are provided best experience by HLS steaming, but inclusion of good IT practice and important business coordination is also a benefit of this method that can not be ignored.

Benefits of HLS

1) The best user experience: - Video clip server can maintain multiple version of video clip in different format. Due to this iPad user with a wifi connection are able to stream higher quality version of the video than iPhone user viewing over 3G connection

2) Reach more viewer: - video delivered with HTTP, support router, NAT and firewall setting in comparison of other protocol, by this more user may access your video.

3) Save on data transfer: in case of HLS only a few segments of video are downloaded at a time, opposed to a progressive download. Five minutes of steam video consumer only designated size, so publisher only pay for that data transfer.

4) Secure video content: - The HLS specification provide provision to ensure security of the stream,. This is great information for broad caster or publishers who want to stream licensed content. The entire HLS stream can be encrypted using AES-128.

G. Cloud front live streaming architecture (LSA)

Saurabh Goyal et all [22] have suggested cloud front LSA for live streaming with amzon web services allows to use the feature of adobe flash media server version 4.5, including live video streamlining where users live video is delivered by a service of HTTP requires from the player that is controlled by manifest files. Flash media server 4.5 supports two HTTP file format: HLS for iOS device and HTTP dynamic Streaming (HDS) for flash applications. User can stream high quality media using the free flash media live encoder desktop applications either for windows or for macOS [22]. Author also explore that cloud front delivery service would support on demand real-time media content streaming from flash media server 4.5. in practice this offers a new, flexible low cost contend delivery network (CDN) solution, particularly for users with relatively small or intermittent streaming delivery needs. AWS charges only for bit stored and bit transferred. There is no monthly minimum, no sign up fee or setup fee, and no ongoing costs unless you actually using the services. Author walks through the steps of setting up cloud front streaming and getting it working on user's site. There are following step for this process.

Step 1: Setup an Aws simple storage service (S3) account where content will live.

Step 2: create a bucket in S3 to store media files.

Step 3: Shift content to S3 bucket and set its permission to allow public access.

Step 4: Setup a cloud front streaming distribution that point at S3 storage bucket

Step 5: Now you are ready to stream.

Author in his architecture say cloud front usage Adobe flash media server 4.5 to stream an demand content with Adobe's Real-time messaging Protocol (RTMP). Cloud front accepts RTMP request over port 1935 and port 80.

Cloud front support the following variant of RTMP protocol:

1) *RTMP:- Adobe's Real-time message protocol*
2) *RTMPT:- Adobe streaming tunneled over HTTP*
3) *RTMPE:- Adobe encrypted over HTTP*
4) *RTMPTE:- Adobe encrypted tunneled over HTTP*

Author also suggested that to secure it, just use RTMPE protocol instead of the regular RTMP.

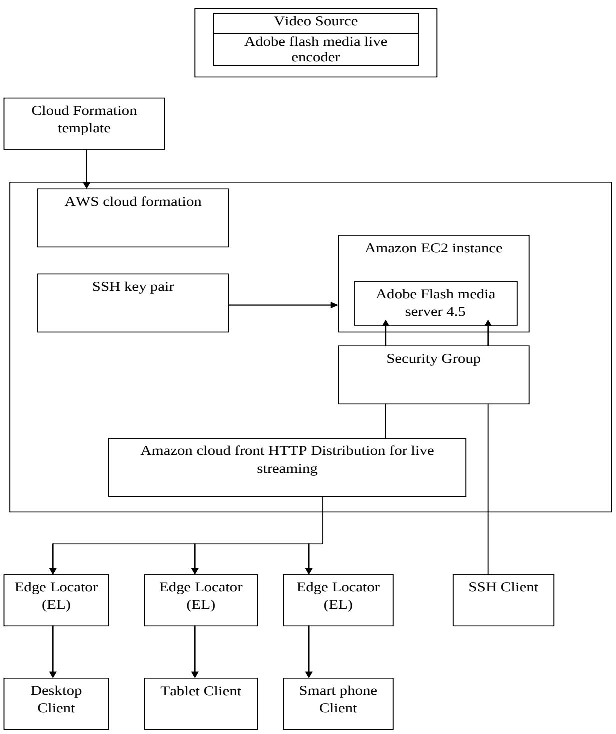

Fig. 4. Cloud front live streaming architecture

According to author statement many IT companies are using HTTP live streaming service to enhance the streaming power in their mobile domain infrastructure.

III. Video Streaming Methods and Parallelization

There are two basic methods of video delivery methods first video delivery via file download and second video delivery via streaming. In file down load method video down load is similar to a file we have down load, but due to its large size there are some drawback of file down load methods, down load approach requires long download method time and large storage space. If anyone wants to see the video then entire video must be downloaded first then it can be play for audience. To overcome above problem we have video delivery via streaming.

A. *Video delivery via streaming*

Video streaming is done to attempts to cope the problems of file downloading method of video delivery. This method also provides significant amount of additional benefit/capability. In video streaming video is splits into parts, then these parts transmitted in succession, and receiver become enable to decode and play back the video as these parts are received, without waiting for entire video to download.

Conceptually video streaming consist of following steps:

Step 1: Partition the compressed video into packets

Step 2: start delivery of these packets

Step 3: Begin decoding and play back at the receiver while the video is still being delivered.

B. *Basic problems in video streaming*

There are number of basic problems those affect video streaming. In the following discussion, we focus on the case of video streaming over the internet since it is an important, concrete example that helps to illustrate these problems. Video streaming over the internet is difficult because the internet offers only best effort services. That is it provides no guarantees on bandwidth, delay jitter or less rate, specially these characteristic are unknown and dynamic. Therefore, a key goal of video streaming is to design a system for reliable delivery of the high quality video over the internet when dealing with unknown and dynamic bandwidth, delay jitter and loss rate [12].

The bandwidth which is available to use b/w two points in the internet is usually unknown and time changing/varying. To overcome the problem of congestion and suboptimal video quality which are the results of mismatch speed of transmitter and receiver, we estimate the available bandwidth and then match the transmitted video bit rate to the available bandwidth.

Accurate estimation of available bandwidth, matching of pre encoded video to estimated channel bandwidth, transmission at a rate which is fair so other concurrent flows in the internet and these entire situations in multicast scenario are the additional considerations that make the bandwidth problem very challenging.

In second problem every packet experience different en-to-end delay that may fluctuate also, at different time. This variation in end to end delay is referred to as the delay jitter. Delay jitter is a problem because the receiver must receive/decode/display frames at a constant rate, and any late frames resulting from the delay jitter can produce problems in the reconstructed video, e.g. jerk in the video [12]. Losses are third fundamental problem in video streaming. Losses those occur in video streaming are depends on particular network under construction.

C. *Media streaming protocols and standards*

In this section we have introduce the network protocol for media streaming over www/Internet. In addition some standards for media streaming are also discussed.

1) *Protocols used for video streaming over Internet*

We review some standards Internet protocol along with media delivery and control protocols

Internet protocols: TCP, UDP, IP.

Initially internet was developed to connect the heterogeneous device to share the information and functioning in which different packet switching technologies were employed. The internet protocol (IP) used to gain base line best effort network delivery to all hosts present in the network, by providing addressing, optimum/best routing, and a common format that can be interpreted/translated by everyone. On the top of the IP there are two transport protocols. First transmission control protocol (TCP) and second user data gram protocol. These protocols play vital role in transmission of packet/datagram.

Some of the difference between TCP and UDP that affects streaming applications are [12].

- TCP operate on a byte stream while UDDP is packet oriented.
- TCP guaranteed delivery via retransmissions, but because of the retransmission its delay is unbounded. UDP does not guarantee delivery, but for those packets which are delivered, their delay is predictable (i.e. one way delay) and smaller.
- TCP provides flow control and congestion control. UDP provides neither. This provides more flexibility for the application to determine the appropriate flow control and congestion control procedure.
- TCP requires a back channel for the acknowledgement. UDP does not require a back channel.

Since TCP/IP provides guaranteed delivery so web and data traffic are delivered with this protocol in which delivery is more important than delay or delay jitter. In contrast UDP/IP is used for media streaming where delay is not acceptable. Compressed media data is usually transmitted via UDP/IP despite control information is usually transmitted via TCP/IP.

Media delivery and Control protocol:

According to the specification given by IETF (internet Engineers Task Force) there are many protocols for media delivery control and description over the internet.

Media delivery:- there are two protocols designed by IETF to support streaming media, first Real-time transport protocol(RTP) and second Real-time control protocol(RTCP). RTP used for data transfer and RTCP for control messages. These protocols are not designed to enable real-time services. RTS are done by underline network only. RTP is not QoS guaranteed and reliable delivery protocol; it provides only support for applications which are time constraint. RTP make it possible by providing a standardized framework for common functionalities such as the stamps, sequence numbering, and payload specification. RTP enables detection of last process. RTCP provides QoS by providing feedback in terms of delay, jitters last packets etc. Feedback message provided by RTCP which is one in every second used by sender to adjust its operations, like, bit rate. Uses of RTP/UDP for media data and RTCP/TCP or RTCP/UDP for control message are conventional approach in media streaming.

Media control: - Real-time protocol (RTSP) and session initiation Protocol (SIP) are two protocols for media control; either can be used for media control. In video streaming RTSP is commonly used protocol. RTSP is used to establish a session and to support VCR functionalities like play, pause, seek and record. For voice over IP (VoIP) SIP is used. It is similar to RTSP, but in addition it can support user mobility and a number of additional functionalities.

Media description and announcement: - for this session description protocol is used. Information like video or audio format, codec, bit rate, duration etc are provided by SDP information description.

2) Video streaming standards and specification

Standard based media streaming systems, as specified by 3rd generation partition ship project (3GPP) for media over 3G cellular and the Internet streaming media alliance (ISMA) for streaming over the Internet, empty the following protocols [12].

Media encoding: - MPEG-4 video and audio (AMR for 3 GPP), H.263

Media transport:- RTP for data, usually over UDP/IP and RTCP for control messages, usually over UDDP/IP

Media session control:- RTSP

Media description and announcement:-SDP

Storage format for the compressed media are not specified by steaming standards but most widely used file format is MP4. MP4 file format have ability to include “hint track’ that hints such as packetization boundaries, RTP header and transmission times etc. to simplify the streaming process.

D. Recent In the video streaming over Internet

During the 1919 and early phase of 2000s, attention of research focused on the design and implementation of new streaming protocols, such as the design of RTP specially for streaming media. This was the client server model for video streaming.

After client server model, peer to peer video streaming was introduced. P2P streaming protocol is based on the philosophy that end hosts, called peers. Peers works as both client and server that is not in the case of traditional client server design. Recently in video streaming HTTP video streaming over the cloud is a new paradigm. In P2P streaming protocols, it is required to download and install dedicated applications in user devices. In opposed to this HTTP video streaming allows user to stream videos directly over the web using a standard Internet web browser, without the need to download and install third party applications.

In new concept streaming server are hosted by cloud computing platform which give an ultimate experience to users, so as a result we are now in the experience of a migration to cloud computing and social media as the effective means to host and share video streams.

E. HTTP video streaming over the cloud

Though P2P has an exigency in being highly effective video delivery methods, but it is not convenient to regular users. A stand alone application is require to be installed in user device to play back the streaming video as well as cache must have capacity to store the whole video. . In contrast HTTP video streaming using standards web browser is most convenient to watch video streams. After webRTC was developed [Bergkvist etal. 2012], these issues seemed to have been resolved; a solution to stream video over HTTP has browsers, a solution to stream video over HTTP has seen a substantial amount of industry support [3].

In this section we are presenting HTTP streaming which was developed even before P2P video streaming but now it is more widely used recently in comparison of P2P.

1) Dynamic adaptive streaming over HTTP

a) HTTP streaming from content distribution networks

Content distribution network (CDN) [Peng 2004] extensively support HTTP streaming, partially due to this, HTTP streaming gain rapid growth. In CDN method different server are deployed in different geographical location which is distributed over multiple ISPs. User can stream video from the server which are nearby their location or optimally closed to users. HTTP video streaming is a process of downloading video segments progressively from web servers via HTTP protocol. Clients who have support of HTTP protocol in their device can seek the arbitrary position in the media stream. This is done by byte range requests to the web server.

As a result, CDN can be effectively used for high quality TV content [cahil and sreenam 2004]. Adhikari et al.[2012] discovered that Netflix, the leading on demand internet video streaming provider, accounts for 29.7% of the peak downstream traffic in U.S. and it employs a mix of datacenters and CDNs for video content distribution. Watson [2011] has studied the dynamic adaptive streaming over HTTP (DASH) framework used by Netflex, which is the largest DASH based streaming provider in the world[3].

b) Research problem with DASH

DASH has been proposed to adapt the streaming rates from web server due to its best effort nature of streaming videos over the Internet. DASH was developed in 2010[MPEG 2010} and has become a new standard in 2011 [stockhammer 2011] to enable high quality streaming of media content over the internet, delivery from conventional HTTP web servers [3].

DASH has following feature

- Segmentation
- Media presentation description(MPD)
- Codec Independence
- Rate Adaptation Components
- Rate Adaptation strategies
- User quality experience

Video component is encoded and divided different segments in which initial segments contain the required information for initializing the media decoder, as well as the media segments containing the video data.

MDP tell about the segments from a video presentation. MDP used to get required segment for smooth playback and adjust bitrates or other attributes according to bandwidth estimates according to client request.

DASH is skeptical and its main container is the MP4 and MPEG-TS. It also allows smooth and flawless adoption of the upcoming improved HEVC video codec (i.e. H265).

Rate adaption component are leaved either for client side or server side implementation. There are different strategies for rate adaption to determine that how different version of segments are received by user to achieve the objective, including fairness, high quality and streaming stability. In terms of user quality experience, though rate adaption is highly correlated with the user's video quality experience in DASH streaming such as correlation may require a more detail study.

Cranley et al [2006] demonstrated the dynamic nature of user perception with adaptive video streaming. In the context of DASH, mock et al[2012] have studied the user experience and observe that users prefer a gradual quality change between the best and worst quality levels, instead of about switching. To better guide the design of rate of adaption strategies, a good metric for evaluation the user experience is still an open area of research[Song et al. 2011] [3].

2) Video streaming from the cloud

Major content providers are using HTTP streaming now days. Thus more and more video streaming is done via HTTP streaming. Due to this, sharing of user generated content are increased by many fold which changed the video streaming landscape. The new landscape provides the platform to the users to generate and upload video content dynamically in to the server.

The design of centralized data centers had increased the popularity of cloud computing since 2008. This design is exactly the opposite of peer-to-peer system design. Due to its "pay as you go" pricing model cloud computing platform for video streaming service are getting popular now. There are some examples like Netflex one of the leading video streaming service provider has been reported to restore the cloud services. This paradigm shift generated a demand of research in the area of cloud based video streaming networks, basically topologies in data centers and algorithm of streaming. There are many benefits of cloud based streaming some of those are inherited from cloud infrastructure like reliability, elasticity, cost effectiveness etc, and others are provided by service provider (e.g. Netflex) which include pay by GB for bandwidth resource which leads to long term cost saving. Cloud based video streaming service may also be able to handle busty demand very well.

Beside these benefits cloud based streaming have some shortcoming due to less research and implementation possibilities. There are number of theoretical and practical issue to be addressed while transferring video streaming services to the cloud. The very first and important problem is the heterogeneity and lease pricing of servers. The billing cycle cannot be arbitrary short, since service provider (e.g. Amazon EC2) provides VM on lease by a fix amount of time like hourly basis. Second when VM is required it must be ordered in advance since it take time (few minutes) to instantiate it. So VoD must be predicted accurately by VoD providers.

A generic framework which provide the facility to migrate existing live media streaming service to a cloud based solution [Wang et al. 2012 a] is shown in fig 5. In this frame work there are two layer, layer1 cloud layer and layer2 user layer, both layer used to take and provide lease and adjusts cloud servers to accommodate temporal and spatial dynamic of user demands.

In the comparison of the Internet a decade ago, networked services in web 2.0 era focus more on the user experience, user participation, and interaction with rich media. The users are now actively engaged to be part of a social ecosystem, rather than passively receiving video stream [3]. If we consider streaming services, one most dominating example which reflects this change is youtube, established in 2005.

You tube is now providing services of 4 billion view or more a day, with most of the video data is generated by users. Now researchers are required to take attention about the huge amount of video content over Internet and also its streaming process.

3) Real-time Adaptive Algorithm for video streaming over mobile network

Protocols for streaming video over internet have existed for decades, and a large number of different protocols have been used in various degrees [11]. In the evolution of video streaming protocols adaptive bit rate streaming technique is most popular today.

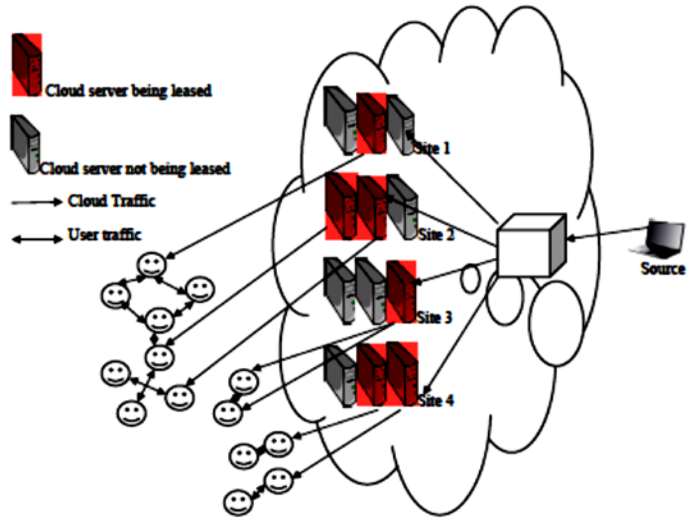

Fig. 5. A frame work for cloud video streaming

A brief history of video streaming is given in table 1.

The protocol used in adaptive streaming is HTTP. HTTP operates on top of TCP, which handle all the data [17].

[Min Xing et al] suggested the optimal video streaming process with multiple links is formulated as a markov decision process (MDP). In this work they have design a reward function to consider the quality of services (QoS) requirements for video traffic, such as the startup latency, playback fluency, average playback quality, play back smoothness and wireless service cost [15]. [Min Xing et al} has proposed an adaptive best action search algorithm to obtain a suboptimal solution, to solve MDP in real-time [15].

TABLE I. THE EVOLUTION OF VIDEO STREAMING TECHNIQUE

S.NO.	**Datagram Streaming**	**Progressive download Streaming**	**Adaptive HTTP Streaming**
1	Uses UDP for transport • Expensive for CDNs • Often problem with firewall	HTT for transport • Uses existing infrastructure to reduce the cost • Firewall friendly	HTTP for transport • Uses existing web infrastructure to reduce the cost • Firewall friendly
2	Send rate close to stream's bit rate	Send rate is unlimited	Receiver limits its download rate when its buffer is full
3	Packet delivery is not guaranteed	Receiver input buffer is unlimited	Uses low bit rate when buffer is empty-> fast start
4	Fast start and little buffering	Play out interruption common	Bit rate adaption reduces play out interruption
5			
6	e.g. Protocol: MMS,RTMP,RTP,PNA	Used in most flash-based video sites	e.g. Protocol: smooth HLS, DASH, HDS
	Complexity of implementation is high	Complexity of implementation low	complexity of implementation medium

IV. Possible Improvement in RTVS Over MCC

The emerging uses of smart phones have clearly indicated an industry trend to seamlessly integrate cloud services with mobile applications. By transferring a small amount of data from mobile device to cloud, user may be provided better experience after taking the advantage of computation power of cloud. A large number of research questions remain open in the area of mobile cloud computing like optimizing the use of cloud computing resources to scale up to millions of mobile users [27].

Here we are listing some possible improvement/findings/future work by which better RTVS can be provided to mobile users

1) Better utilization of inter data center network by fairly allocating and slicing network bandwidth using network virtualization [27].

2) Implementation of network virtualization at the application layer without changing traditional transport protocol [27].

3) Standardization of mobile cloud execution platform for the easier computation offloading [2].

4) New policies synchronization between smart phone and smart phone clone in the cloud [2].

5) Development of security mechanism to secure the clone from illegal access and protect the smart phone user from the malicious VMs executing in the cloud [2].

6) Automatic path prediction and path prediction that fails in video streaming over MCC [11].

7) Investigation of better load allocation between several links with finer granularity [15].

8) Improving the bandwidth estimation accuracy by the consideration of size of the video segment for variable bit rate (VBR) video [15].

9) Prediction of future bandwidth in RTVS over MCC environment [15].

*10)*Adjustment of mechanism of pre fetch window size to get optimum performance of online pre fetch algorithm in Adaptive video streaming over HTTP [13].

*11)*Implementation of video proxy in the ISP to mitigate the traffic redundancy of mobile video player in RTVS over MCC environment [1].

*12)*Observing the behavior of adaptive video player by large scale study of traffic redundancy in adaptive video players in RTVS over MCC environment [1].

*13)*Understanding and standardization of the economics between ISPs and video providers to provide good quality video for mobile/end users [1].

*14)*Virtualization and immigrate task from terminal to cloud to achieve better result /QoS in RTVS over MCC environment [10].

*15)*Standardization of MCC services billing for RTVS [30].

*16)*Optimization of performance metrics of MCC data center by non-linear function and multi-dimensional optimization algorithm [20].

*17)*Investigating the issue with flickering effect on link aware reconfigurable point to point video streaming for mobile devices [19].

*18)*Utilization of weight based optimized routing for better RTVS experience over MCC environment [5].

*19)*Development of new system architecture and standards that seamlessly integrate MC, IOT (Internet of things) and protocols that facilitate big video data streaming from IOT to MC [18].

*20)*Employing optimal multiple stopping method to get better RTVS over MCC environment [23].

*21)*Integration of MCC application development with agile development methodologies [8].

*22)*Utilization of QoS control mechanisms to balance between the rapid response of the abnormal network and the avoidance of the rate oscillation in RTVS [6].

*23)*Providing energy efficient green mobile cloud computing using the cloud Exptool[23] [7].

24) An efficient algorithm to determine the redistribution of system resources in a slice after a single node fails in RTVS over MCC environment [16].

*25)*Real-time scheduling of resources (surrogate node) in RTVS over MCC environment [31].

*26)*New scheduling scheme for energy efficient cloud offloading for multi-core mobile devices, while considering downloading the cloud execution output in the model [29].

*27)*Scheduling of the surrogate node and finding the availability of surrogate nodes is a NP complete problem, to compute the schedule, a computational grid of desktop can be used instead of having a large machine [25].

V. Social and Legal Issue of RTVS

Mobile live streaming brings the new way of showing video content to the users. In this field two major apps are launched recently meerkat” in feb 2015 and periscope in march 26 2015. This time these two are the main competitor in the market of mobile live streaming (RTVS). The person who is having smart phone, the Internet and an app can now broad cast live video to the world. This is great for users but is a nightmare for content right holders.

RTVs can be used for business, education, healthcare and training and development. So one side of this technology shows a very positive impact on society, but in another side it has a question of different intellectual property right issue, as well as copy right and cyber security concern. These live streaming apps lets users simply point their smart phones at whatever is happening in front of them, whether they own the right or not, and broadcast it to a potential audience of hundreds of thousands now , millions in future.

Beside many benefits of RTVS, concern has been raised in particular over large right holder such as entertainment studios, sport broadcaster etc suing periscope for users who film content direct from TV or events via their smart phones.

In this violation of IPR and copyright matter one of the most famous examples is Mayweather-Pacquiao boxing fight in 2015.

HBO saw an estimated 10,000 people watching this fight via periscope without paying any amount of money. This raises the alarm to have some standards and rules for RTVS among users, to protect the user's right, social benefits and IPR with CR issue.

VI. Conclusion

After the exhaustive study of different research papers and related article we found that real-time video streaming in mobile devices is now going to be next challenging field for the consideration of new research.

This paper tells that use of mobile cloud computing environment for the RTVS may provides better RTVS experience. Beside this in this study, paper also presented that there are many architectural and QoS issue in this area. E.g. standardization of streaming process for MCC environment, standardization of billing process of MCC resources, scheduling of resources for video segments in different surrogate nodes, prediction of required bandwidth for streaming, and IPR and Copyright issue as well as user's rights, etc.

This paper listed many issues along with possible future improvement and work in the field of RTVS over MCC environment. In future work any of the above explained possible improvement in the area of RTVS process over MCC environment may be implemented.

References

[1] A. Mansy, "NETWORK AND END-HOST SUPPORT FOR HTTP ADAPTIVE VIDEO STREAMING, " Phd Thesis School of Computer Science Georgia Institute of Technology May 2014

[2] A. Rehman Khan, M. Othman, S. Ahmad Madani, and S. Ullah Khan "Survey of Mobile Cloud Computing Application Models, " in IEEE COMMUNICATIONS SURVEYS & TUTORIALS, ACCEPTED FOR PUBLICATION, 2014.

[3] BAOCHUN LI, Z. WANG, J. LIU, and WENWU ZHU, "Two Decades of Internet Video Streaming: A Retrospective View, "in ACM Transactions on Multimedia Computing, Communications and Applications, Vol. 9, No. 1s, Article 33, October 2013.

[4] C. Konstantopoulos, "A parallel algorithm for motion estimation in video coding using the bilinear transformation," in SpringerPlus 2015.

[5] D. Sarddar, R. Bose and S. Sahana, "A novel approach on weight based optimized routing for mobile cloud computing, " in *Brazilian journal of Science and Technology (2015) 2:3*

[6] D. Xiao-dan, H. Qing, L. Yong-hong, and Yu Hong, "The System Construction and the Implementation of QOS Control Mechanism in Intelligent Streaming Media, "in Elsevier Procedia Computer Science Volume 25, pp. 808–813, 2012.

[7] E. Benkhelifa,T. Welsh, L. Tawalbeh, Y. Jararweh, and Anas Basalamah, "User Profiling for Energy Optimization in Mobile Cloud Computing, " in Elsevier Procedia Computer Science Volume 52, pp. 1159–1165, 2015.

[8] F. Almudarra, B. Qureshi, " Issue in adopting Agile Development Principle for Mobile Cloud Computing Applications, " in Elsevier Procedia Computer Science Volume 52, pp. 1133–1140, 2015.

[9] G. Orsini, D Batde, and W. Lamersdorf, "Context Aware Computation Offloading for Mobile Cloud Computing: Requirements Analysis, Survey and Design Guidelines, "in Elsevier Procedia Computer Science Volume 56, pp. 10–17, 2015.

[10] H. Qi, and A. Gani, "Research on Mobile Cloud Computing: Review, Trend and Perspectives, " Research work at Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant UM.C/HIR/MOHE/FCSIT/03 2012

[11] Haakon Riiser, "Adaptive Bitrate Video Streaming over HTTP in Mobile Wireless Networks," in *Series of dissertations submitted to the Faculty of Mathematics and Natural Sciences, University of Oslo No. 1372 2013*.

[12] J. G. Apostolopoulos, W. Tan, and S. J. Wee "VIDEO STREAMING: CONCEPTS, ALGORITHMS, AND SYSTEMS, " by *Streaming Media Systems Group Hewlett-Packard Laboratories Palo Alto, CA, USA,* Mobile and Media Systems Laboratory HP Laboratories Palo Alto HPL-2002-260 September 18th , 2002

[13] K. Liang, J. Hao, R. Zimmermann, and D. K.Y. Yau," Integrated Pre-fetching and Caching for Adaptive Video Streaming over HTTP: An Online Approach, "in MMSys '15, March 18 - 20, 2015, Portland, OR, USA.

[14] M. ALTAMIMI, A. ABDRABOU, K. NAIK, and A. NAYAK, "Energy Cost Models of Smartphones for Task Offloading to the Cloud, " in IEEE Transaction on EMERGING TOPICS IN COMPUTING VOLUME 3, NO. 3, pp. 384-398 SEPTEMBER 2015

[15] M. Xing, S. Xiang, and L. Cai, "A Real-Time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks," in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 32, NO. 4, pp. 795-805, 2014.

[16] N. Thomas, M. Thomas, and K. Chandrasekaran, "Multimedia Streaming using Cloud-Based P2P Systems, "in Elsevier Procedia Computer Science Volume 57, pp. 25–32, 2015.

[17] P. Ibañez Verón and J. Martinez Garcia, " REAL-TIME STREAMING VIDEO FOCUSED ON MOBILE DEVICES, " by School of Innovation, Design and Engineering, Mälardalen University Högskoleplan 1, 722 18 Västerås, Sweden.

[18] S. Kim, "Nested game-based computation offloading scheme for Mobile Cloud IoT systems," EURASIP Journal on Wireless Communications and Networking (2015) 2015:229

[19] S. LEE, S. YOO, J. JUNG, and H. KIM, and J. RYOO, "Link-Aware Reconfigurable Point-to-Point Video Streaming for Mobile Devices, " in ACM Trans. Multimedia Comput. Commun. Appl., Vol. 12, No. 1, Article 9, August 2015.

[20] S. Vakilinia, D. Qiu and M. Mehmet Ali, "Optimal multi-dimensional dynamic resource allocation in mobile cloud computing ," in *EURASIP Journal on Wireless Communications and Networking* 2014, **2014**:201 Page 2 of 14 http://jwcn.eurasipjournals.com/content/2014/1/201

[21] S.S. de Almeida, A. C. de Nazar´e J´unior, A. de Albuquerque Ara´ujo, G.C´amara-Ch´avez, and D.Menotti, "Speeding up a Video Summarization Approach using GPUs and Multicore CPUs," in Elsevier Procedia Computer Science Volume 29, pp. 159–171, 2014.

[22] Saurabh Goel, "Cloud-Based Mobile Video Streaming Techniques, " in Global Journal of Computer Science and Technology Network, Web & Security Volume 12, Issue 17, Version 1.0, 2012.

[23] T. Jing, X. Cui, W. Cheng, S. Zhu, Y. Huo and X. Cheng, "Enabling smartphone-based HD video chats by cooperative transmissions in CRNs," in *EURASIP Journal on Wireless Communications and Networking* (2015) 2015:45.

[24] T. L. Vinh, S. Bouzefrane, J.M. Farinone, A. Attar, and B. P. Kennedy "Middleware to Integrate Mobile Devices, Sensors and Cloud Computing, " in Elsevier Procedia Computer Science Volume 52, pp. 234–243, 2015.

[25] V. Dhir, "Alchemi.NET Framework in Grid Computing, " in Proceedings of the 3rd National Conference; INDIACom-2009 Computing For Nation Development at Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi , February 26 – 27, 2009.

[26] X. Ma, Y. Cui and I. Stojmenovic "Energy Efficiency on Location Based Application in Mobile Cloud Computing: A survey, " in Elsevier Procedia Computer Science Volume 10, pp. 577–584, 2012.

[27] Y. Feng, "ON IMPROVING THE PERFORMANCE OF MOBILE APPLICATIONS USING THE CLOUD, " in thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy Graduate Department of Electrical and Computer Engineering University of Toronto 2013.

[28] Y. Jararweh, L.Tawalbeh, F. Ababneh, A. Khreishah, and F. Dosari, “Scalable Cloudlet-based Mobile Computing Model, “ in Elsevier Procedia Computer Science Volume 34, pp. 434–441, 2014.

[29] Z. Jiang, and S. Mao, “Energy Delay Trade-off in Cloud Offloading for Mutli-core Mobile Devices, ” accepted for publication in a future issue of IEEE access journal 2015 (This work was presented in part at IEEE GLOBECOM 2015, San Diego, CA, Dec. 2015)

[30] Z. Sanaei, S. Abolfazli, A. Gani, and M. Shiraz, “SAMI: Service-Based Arbitrated Multi-Tier Infrastructure for Mobile Cloud Computing, “ Research work at Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant UM.C/HIR/MOHE/FCSIT/03 2012

[31] Z.H. ZHAN, X. F. LIU, Y.J. GONG, J. ZHANG, H. S. CHUNG, and Y. LI, “Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches,” ACM Computing Surveys, Vol. 47, No. 4, Article 63, July 2015.