

# Challenges of Mobile Cloud Computing

Rangeena Reghish

Department of Computer Science and Engineering  
KMCT College of Engineering,  
Calicut, Kerala, (India).

**Abstract**— This article summarized the concept of Mobile cloud computing, pointed out many problems about Mobile cloud computing, and proposed the solving methods.

**Keywords**— Mobile Cloud Computing.

## I. INTRODUCTION

Mobile cloud computing is 3+ combination, that is the combination of mobile computing, mobile internet and cloud computing. Mobile computing technology is to share resources and transport data of computers or other intelligent terminal equipments such as cell phones. The essence of mobile cloud computing is to provide valuable, precise and real time information to any clients at any time, at any place.

Mobile cloud computing means that any intelligent terminal equipments such as cell phones and personal computers can obtain services in wireless environment, which is also called mobile cloud computing. Handheld equipments are often individualized from outlook to operating system, which is limited to the requirement off volume and convenience, the processing ability of CPU, storing space, keyboard and screen, battery and bandwidth compared with PC, especially for storage and computing ability.

Mobile cloud computing offers a way to meet users' increasing functionality demands, as distant servers execute all application logic and only user interface functionalities reside on the mobile device. The device acts as a remote display, capturing user input and rendering display updates received from the server. However, short battery lifetime, varying wireless channel conditions, and interaction latency present major challenges for the remote display of cloud applications on mobile devices. Researchers have recently proposed several solutions to these issues.

## II. CHARACTER OF MOBILE CLOUD COMPUTING

### A. Hardware of handheld equipment and independence of the system:

All the computation are carried on in the cloud-far end servers, therefore, mobile cloud computing does not have requirement for handheld equipment, even unintelligent cell phones can realize mobile cloud computing.

### B. Effectiveness of task processing:

All the This advantage is obvious thanks to the processing ability of cloud end. If the interface of

input and output is good enough, we can see the result of tasks directly by cell phones.

### C. Convenience of sharing data:

A large amount of data is stored in the cloud end of servers, enabling sharing data conveniently. If the bandwidth is wide enough, it will work as fluently as locally, which is easy to realize for cell phones. Compared with the data transportation among former clients, the cloud storage method will enhance the convenience of sharing data.

### D. Elimination of regionality:

A Mobile cloud computing eliminates the limitation of regionality, enabling people to obtain what they want at any time and any place from the internet.

## III. MOBILE CLOUD COMPUTING CHALLENGES

Essentially, mobile cloud computing physically separates the user interface from the application logic. The mobile device executes only a viewer component, operating as a remote display for the applications running on distant servers in the cloud. A remote display framework has three components: a server-side component that intercepts, encodes, and transmits the application graphics to the client; a viewer component on the client; and a remote display protocol that transfers display updates and user events between both endpoints.

Using standard thin-client solutions such as Microsoft's Remote Desktop Protocol (RDP), Citrix Systems Independent Computing Architecture (ICA), and AT&T's Virtual Network Computing (VNC) in a mobile cloud computing context is not straightforward. These architectures were originally designed for corporate environments, where users connect over a wired local area network (LAN) to the central company server executing typical office applications. In this setting, the technical challenges are limited because delay and jitter are minimized, bandwidth availability is seldom a bottleneck, and office applications exhibit rather static displays compared to multimedia applications. In a mobile cloud computing environment, the remote display protocol must deliver complex multimedia graphics over wireless links and render these graphics on a resource-constrained mobile device.

All kinds of services of wireless internet of mobile cloud computing will dig into people's life, in the mean time, using all kinds of services of wireless internet will enhance the convenience, however, a lot

of problems still remain in the realization of mobile cloud computing.

*A. The problem with cell phone platform:*

According to the current situation, a smooth client interface is realized in the cell phone end, which mainly includes flash and java. However, currently, most cell phones which support both flash and java are most intelligent cell phones, while most ones in the future will support them, which means that mobile cloud computing is not far away from us.

*B. System problem of cloud end:*

As to system and software of cloud end, cloud end should possess a synchronized system of input and output of equipment like cell phone which is in charge of data processing.

*C. Stability of bandwidth:*

This is a problem which can rely on mobile corporation totally. However, it is also the key problem to maintain the stability of cloud computing, only when the mobile corporation can provide bandwidth that is stable enough, can we talk about mobile cloud computing with low delay.

*D. The problem of input and output interface:*

Anyhow, the input and output of cell phones are always limited, additional hardware will increase the load, it is also weak in usage, by letting clients to define their own input and output interface, picking out their common functions to form the input interface, and pick out deeper interfaces when needed. As to output interface, we should also make the data briefly, letting small cell phone screen to provide as much information as possible and maintain the convenience.

*E. The problem of service provider:*

The difficulty of developing handheld equipment makes not all IT companies can enter the wireless internet soon, but they are eager to extend the service on the internet to endless internet. In this case, some service providers will no doubt come out and will have a bright future, for example, they can provide information exchange, payment, reading, email delivery, games, SNS and so on. These service providers will provide standard technique interface, making another company to use freely, which is exactly the amazing part of mobile cloud computing.

*A. The problem of the individualization of handheld equipment:*

Currently, there are many kinds of handheld operating systems, if we want to develop application based on handheld equipment rapidly, we must make application software as simple client end. Simple client end means that a large amount of data processing will be put in the cloud end, and the design of technique architecture must be realized according to cloud computing.

*B. The problem with the loss of battery and network flow:*

If an application need a lot of battery and flow, the application will be hard to be used in handheld equipment. In order to solve the problem with the loss of battery and network flow, we can cut down the data exchange rate and amount between handheld equipment and cloud end by optimizing cloud end, which will induce the loss of battery and network flow effectively. As to all kinds of real time messages and noticing service, delivery technique is also very important, access in turn which is popular in PC, cannot be applied to handheld equipment either.

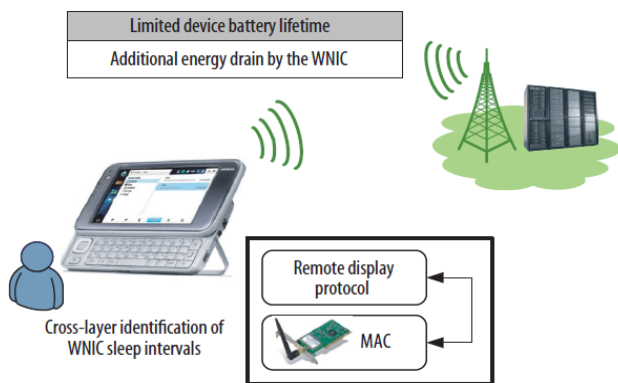
More fundamental challenges for mobile cloud computing lie in the short battery lifetime of mobile devices, the limited and varying bandwidth on wireless links, and the interaction latency between user input and display updates.

**DEVICE BATTERY LIFETIME:**

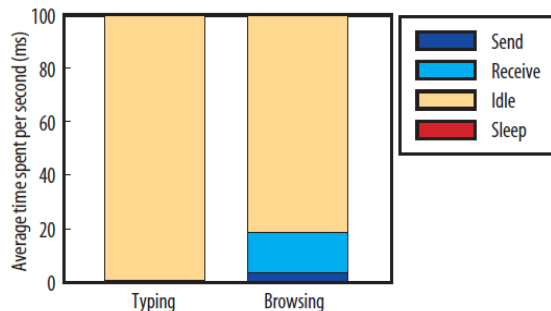
The operational time of mobile devices is often limited when they are extensively used. These battery capacity shortcomings result in short recharge cycles and prevent users from relying completely on their mobile device. During the past decade, advances in nominal battery capacity have been modest. Kostas Pentikousis<sup>5</sup> observed that technological improvements are currently stagnating because of the lack of a major battery technology breakthrough comparable to the advent of rechargeable Li-ion batteries. Consequently, extending device autonomy should primarily be realized by making the device itself more energy efficient.

*Offloading mobile applications:*

At first glance, offloading mobile applications to the cloud appears to be a straightforward way to save on device energy consumption because it reduces the amount of local processing. As Figure 1 shows, however, there is a tradeoff between local processing and network bandwidth consumption, and bidirectional communication with the application server incurs additional energy drain from the battery via the wireless network interface card (WNIC). From an energy perspective, offloading applications from mobile devices is mainly advantageous when large amounts of computation are needed in combination with relatively small amounts of network communication. Demanding applications exchange a significant quantity of data between client and server because they entail a high degree of interactivity and detailed graphics—for example, walking around in a 3D virtual environment or rotating 3D medical images.



**Fig 1.** Offloading mobile applications to the cloud reduces the amount of local processing, but there is a tradeoff because bidirectional communication with the application server incurs additional drain from the battery via the wireless network interface card (WNIC). A proposed cross-layer power-saving approach operates between the media access control (MAC) layer and the remote display protocol layer.



**Fig 2.** Average time distributions of the four possible WNIC modes—send, receive, idle, and sleep—resulting from remote display protocol traffic in two different scenarios: typing and browsing. The WNIC is mainly in idle mode and seldom enters sleep mode.

**Table 1. Cisco Aironet WNIC power consumption.**

Power mode	Measured power (mW)
Send	1,090-1,550
Receive	1,060-1,380
Idle	1,150
Sleep	300

These idle times are a consequence of the limited frequency of user interactions imposed by the network round-trip time. After some interaction, users must wait until the results become visible on the screen before continuing their work. Furthermore, interactive applications will only update their display when instructed by the user—for example, by entering a URL or clicking on a hyperlink.

Major energy savings are expected when the WNIC transitions to energy-conserving sleep mode during these idle intervals. Sleep mode consumes three to five times less energy than idle mode because the radio interface is turned off. Of course, this implies that the WNIC will miss any incoming data when it is in sleep mode, so the sleep intervals must be carefully chosen.

A proposed cross-layer power-saving approach operates between the MAC layer and the remote display protocol layer (see Figure 1). Because the MAC layer operates on binary data and cannot discriminate between, for example, transmitted user input and Transmission Control Protocol (TCP) acknowledgments, it is unaware of the arrival of the next display update. The appropriate sleep intervals must therefore be determined at the remote display protocol layer, where the display update schedule is established—for example, via a push approach in which the server sends display updates with fixed intervals or a pull approach in which the client sends an explicit request. Correlating the transmission of user input to the network round-trip time predicts the arrival of the next display update. In between two display updates, the WNIC enters sleep mode. This sleep mode is interrupted at regular intervals to transmit user events. Researchers have used cross-layer optimization to reduce WNIC energy consumption by up to 52 percent.

**WIRELESS BANDWIDTH AVAILABILITY:**

Compared to fixed access networks, modern broadband mobile and wireless technologies offer limited and variable bandwidth availability. Universal Mobile Telecommunications System (UMTS) users typically receive up to 384 kilobits per second, while Krishna Balachandran and colleagues<sup>9</sup> reported practical throughputs of 347 Kbps for Long Term Evolution (LTE) and up to 6.1 Mbps for WiMAX. Actual throughput depends on user mobility, interference, and fading effects.

Beyond the higher costs associated with limited broadband availability, other economic factors are motivating the development of highly efficient remote display compression technologies. Mobile users are increasingly confronted with volume-based subscription plans—such as that adopted by AT&T in 2010—and these users will not tolerate the sending of redundant bytes on the network. In fact, the lack of such technologies could impede development of new cloud applications.

*Versatile Graphics Encoding:* As Figure 3 shows, the choice of codec to compress the intercepted application graphics at the server is a tradeoff between visual quality, compression efficiency, and decoding complexity.

Conventional remote display architectures including RDP, ICA, and VNC typically virtualize a layer of the graphic rendering stack at the server and forward intercepted drawing primitives to the client, such as

instructions to draw a rectangle, display a bitmap, or put some text on the screen. This approach is optimal for applications, such as typical office applications, that only update small regions of the display or have a slow refresh rate with respect to the network round-trip time. Bandwidth requirements to remotely display this type of graphics do not exceed 200 Kbps and can be adequately served over wireless links.

On the other hand, encoding multimedia graphics applications would require numerous drawing primitives because they update large parts of the screen at high refresh rates and they often contain fine-grained and complex color patterns. This kind of graphics can be more efficiently encoded using a video codec, such as H.264 or MPEG-4. Using video codecs for remote display purposes is referred to as *interactive live streaming* because the graphics are mainly the result of user interaction, in contrast to regular video streaming, which requires only limited user interaction—for example, to start and stop the video. Researchers have successfully applied interactive live streaming in remote 3D virtual environments and games.

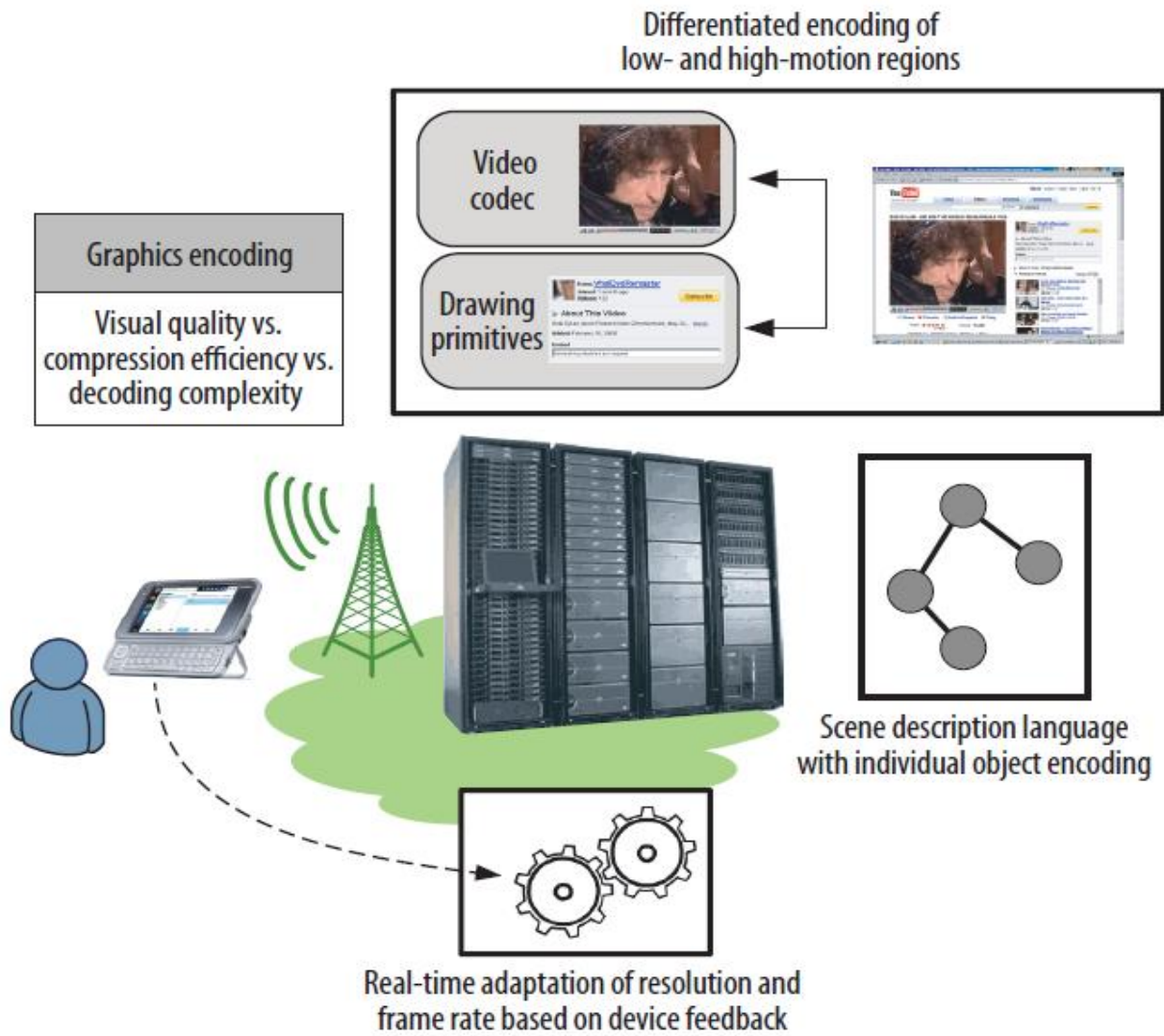
Even when only a single application is used, the characteristics of the graphics on the user display might significantly differ when a user is accessing mobile cloud computing services. For example, a user browsing a Wikipedia page might click on a link that opens a YouTube video in the same browser window. Remote display frameworks must therefore be able to switch seamlessly between multiple encoding modes based on an analysis of graphics at the server (see Figure 3).

While the characteristics of the application graphics mainly determine the choice of encoding format, the actual encoding parameters must be dynamically adapted to cope with wireless bandwidth fluctuations and heterogeneous mobile device capabilities. Numerous factors impact wireless link quality, including device antenna technology, distance from the access point, user speed and fading, and interference effects. At the same time, the various hardware configurations of commercial mobile devices induce variations in decoding power.

The mobile device regularly reports on the amount of data that the controller encodes per unit of time, a metric reflecting both the device hardware capabilities and the amount of data the device receives. By adjusting the resolution and image quality accordingly, the controller maintains a target frame rate that ensures a smooth visualization experience.

#### *Downstream data peak reduction:*

Interactive applications only update their display when instructed by the user. These display updates usually involve sending a large amount of data to the client in a short interval, which requires an instantaneous bandwidth much higher than the average bandwidth requirement. Furthermore, this bursty



**Fig 3.** The choice of codec to compress the intercepted application graphics at the server is a tradeoff between visual quality, compression efficiency, and decoding complexity. Remote display updates are optimally encoded through drawing primitives or video codecs, depending on the graphics characteristics. Low- and high-motion scenes must be encoded separately, requiring runtime detection algorithms or the use of semantic scene description languages. Video streaming parameters must be adapted to device feedback.

**Table 2.** Packetization overhead of TCP/IP headers when sending a single keystroke to the server.

Protocol	Payload (bytes)	Overhead (percent)
VNC Remote Framebuffer Protocol	8	83.33
Remote Desktop Protocol	6	86.96
Independent Computing Architecture	6	86.96

traffic pattern is unfavorable in wireless network environments, as it might induce additional collisions on the wireless channel. A study of this problem by Yang Sun and Teng-Tiow Tay revealed that repainting the graphical objects after recurring user actions causes considerable redundancy in remote display protocol traffic. Their proposed hybrid cache-compression scheme uses the cached data as history to better compress recurrent screen updates, with the cache containing various drawing orders and bitmaps. Using RDP, they reduced the number of data spikes by 27 to 42 percent, depending on cache size, resulting in global network traffic reductions of 10 to 21 percent.

#### *Optimization of upstream packetization overhead :*

User events are the principal source of upstream remote display traffic from client to server. Individually, each user event embodies only a small amount of information: a key or button ID, one bit to discriminate between the press and release actions, and possibly the current pointer coordinates. Nevertheless, user events induce important upstream traffic because they are often generated shortly after each other. Entering a single character results in two user events to indicate the press and release actions, whereas moving the mouse results in a sequence of pointer position updates.

Usually, the system transmits user events as they occur to minimize interaction latency. Because data packets sent upstream often contain a single user event, headers added at the TCP, IP, and wireless link layer cause large packetization overhead. Table 2 quantifies the packetization overhead of TCP/IP headers of three commonly used remote display protocols—the VNC Remote Frame buffer (RFB) Protocol, RDP, and ICA—when sending a single keystroke to the server. Optional headers and the wireless link layer header further increase total overhead.

Buffering user events at the client for a short period enables the joint transmission of multiple user events. The maximum buffering period is based on a tradeoff of remote display bandwidth reduction and interaction latency. A closed-loop controller running at the client integrates the interaction latency models in terms of this buffering period and the network round-trip time ensuring that the average interaction does not exceed a predefined maximum value by continuously monitoring the current network status and adjusting the buffering period accordingly. The highest bandwidth reductions are achieved for interactive applications with frequent user events and lower round-trip times. For a text-editing scenario and network round-trip times less than 50 ms, researchers have achieved bandwidth reductions up to 78 percent.

## **INTERACTION LATENCY**

While technological advances are likely to overcome bandwidth limitations, interaction latency—the delay users experience between generating some input and seeing the result on their display—is an

intrinsic challenge of mobile cloud computing because the device must communicate even the most trivial user operations to the server.

- [1] Accustomed to the responsive interfaces of desktop applications, users will expect the same interactivity in a mobile cloud computing setting. As Figure 4 shows, remote display protocol data needs to traverse numerous links, both wireless and wired, and multiple network elements, each introducing additional propagation and transmission delays on the end-to-end path. Loss-correcting retransmissions on the wireless link, router queuing, suboptimal routing schemes, and firewall processing all entail significant propagation delays. Bandwidth limitations on the wireless link induce additional transmission delays, especially for immersive applications such as virtual environments that transfer highly detailed graphics from the server to the client. Sometimes several client-server interactions are required before a display update can appear on the screen—for example, when the server waits for the client's acknowledgment before sending the remainder of the data.
- [2] Solutions to mitigate interaction latency try to either reduce the number of hops on the end-to-end path by moving the application closer to the client or provide better synchronization mechanisms between client and server (see Figure 4).

#### *Proximate cloudlet infrastructure*

Satyanarayanan and colleagues introduced the concept of *cloudlets*: trusted, resource-rich computers dispersed over the Internet. Exploiting virtual machine technology, mobile devices rapidly deploy their services on the closest cloudlet by uploading an overlay VM to customize one of the generic base VMs commonly available on all cloudlets. The physical proximity ensures low-latency, one-hop, high-bandwidth wireless LAN access—for example, over the latest Wi-Fi 802.11n technology—instead of mobile radio technology access, such as high-speed downlink packet access (HSDPA) or LTE.

Despite its promise, the cloudlet concept may require the transfer of data from the central application server to nearby public infrastructure, which can be undesirable for security or privacy reasons. In these cases, latency optimization strategies must focus on reducing the number of round-trip times required to resynchronize the client device display with the server.

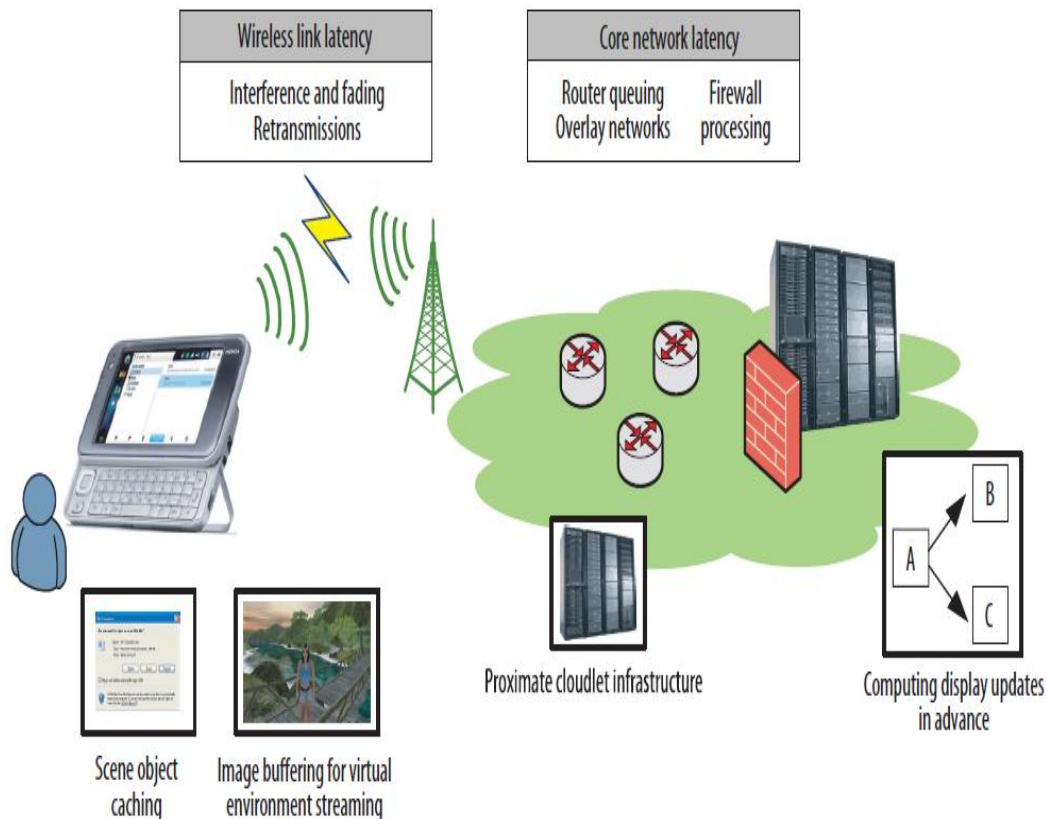
#### *Computing display updates in advance*

Given the current application state, the application server can predict potential display updates and stream them in advance to the client. Contrary to video streaming, in which the frame order is known in advance, in mobile cloud computing, the next display update depends on user input. For example, when a user opens an application menu, the

server can pre compute all dialog windows that can be opened by selecting one of the menu items.

Given the current user position, the system calculates the possible next user viewpoints in

advance and provides them to the client. When the user actually moves forward, the client fetches the correct viewpoint from its cache.



**Fig 4.** Strategies to mitigate interaction latency focus either on reducing the propagation delay by deploying the application on the proximate infrastructure or on reducing the synchronization between client and server by caching key objects or frames.

#### Image buffering for virtual environment streaming

Due to limitations in mobile bandwidth and mobile device memory resources, in most cases, streaming all possible next display updates in advance is unfeasible. Furthermore, the gains of this precomputing technique are highly dependent on prediction accuracy. A better strategy might be to buffer some key display updates, for which the server only needs to provide a differential update. Boukerche, Pazzi, and Jing Feng<sup>20</sup> evaluated several cache management strategies and reduced the amount of requests during a 300-step movement in a 3D virtual environment from 300 to 145. Of course, in this case, the server response is still required to update the display.

#### Scene object caching

For more static applications, such as office applications, the potential next updates can be more accurately predicted as, for example, a menu layout will rarely change. Consequently, the number of corrective server updates will be more limited. A typical example would be the list of recently opened files in a text editor's File menu. Scene description languages such as MPEG-4 BiFS are particularly

suited to support this client-side handling of user input.<sup>13</sup> The client not only receives graphic updates, but also is informed about the structure of the displayed scene and its composing objects, as well as how the user can manipulate these objects.

By physically separating the user interface from the application logic, mobile cloud computing allows access to even the most demanding applications from intrinsically resource-constrained mobile devices. Developers tailor contemporary remote display optimization techniques to mobile devices' short battery lifetime, the varying and limited bandwidth availability on wireless links, and interaction latency. Although each of these solutions adequately addresses specific mobile cloud computing challenges, an overall approach is currently lacking.

Because of user mobility, the wide diversity of applications, and the varying wireless channel status, the mobile cloud computing context is highly dynamic. Future research should therefore focus on the design of a comprehensive framework that integrates the existing solutions and activates the most appropriate one depending on the current device, network, and cloud server status.

#### IV. CONCLUSIONS

The number of mobile users all around the world is 42,800,000, which takes up 1.1 % of all the cell phone users. In the future 5 years, the internet will develop rapidly and in 2014, the number of users will rise up to 0.998 billion, taking up 19% of all users around the world. 2009 is the 3G year for China, when mobile cloud computing will become popular. Various kinds of internet services using mobile cloud computing will also be popular with people's daily life. Mobile cloud computing will not only become popular but also develop continuously during all challenges, which are the opportunities for us.

#### REFERENCES

- [1] Man Yuan. *Mobile Computing*. Harbin: Harbin Industrialization University Press. 2008. (8) : 2- 10
- [2] Canfeng Chen. *Wide Band Mobile Internet*. Beijing: People's Delivery and Electricity Press. 2005 : 43-49.C.
- [3] T. Kim et al., "An Energy-Aware Transmission Mechanism for WiFi-Based Mobile Devices Handling Upload TCP Traffic," *Int'l J. Comm. Systems*, May 2009, pp. 625-640.
- [4] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" *Computer*, Apr. 2010, pp. 51-56.
- [5] V.S. Pendyala and S.S.Y. Shim, "The Web as the Ubiquitous Computer," *Computer*, Sept. 2009, pp. 90-92.
- [6] K. Balachandran et al., "Performance Assessment of Next-Generation Wireless Mobile Systems," *Bell Labs Technical J.*, Feb. 2009, pp. 35-58.
- [7] M. Mitrea et al., "BiFS-Based Approaches to Remote Display for Mobile Thin Clients," *Proc. SPIE*, vol. 7444, 2009, p. 74440F, doi:10.1117/12.828152.
- [8] V.S. Pendyala and S.S.Y. Shim, "The Web as the Ubiquitous Computer," *Computer*, Sept. 2009, pp. 90-92