# Client Technology on a Server for Mobile Cloud

Tien-Dung Nguyen | Song Biao | Tang Wei | Jun-Hyung Lee | Eui-Nam Huh
Kyung Hee University

## Abstract

The increasing ubiquity of wireless networks and decreasing cost of hardware is fueling a proliferation of mobile devices. These devices are enabling a new revolution in mobile technology, not only running locally on them, but running on cloud as a service as well. From web browsing, email, or video conferencing, presentations to movie and music entertainment or games, multimedia applications, mobile cloud enables providing such diverse applications. Many technologies have been designed to address the limited hardware and performance in thin client PC. However, with the assorted network and graphic condition, those proposed technologies is obligated to alter aim to adapt mobile cloud. In this paper, we provide a survey of client technology on a Server that can be sufficed the requirements of Mobile Cloud. We also analyze each technology and classify with its individual difficulties and challenges.

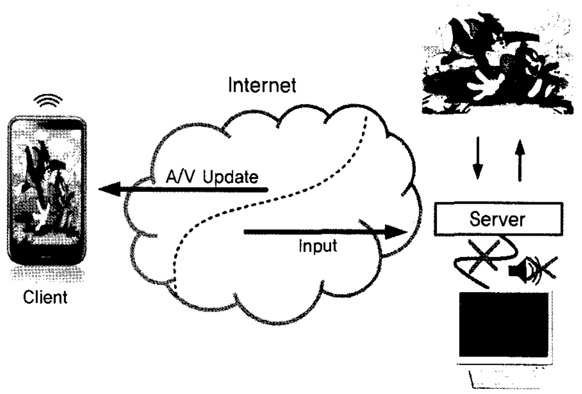
## I. Introduction

According with the development of cloud computing, mobile devices is growing overwhelmingly. The mobile devices are including wireless Personal Digital Assistant (PDA), mobile phone, smartphone (integrated PDA with mobile phone), tablet, etc. These devices are enabling new forms of mobile computing and communication, named Mobile Cloud Computing (MCC). MCC is emerging as one the most important branches of cloud computing, and is still in its infancy.MCC is defined as cloud computing extended by mobility and a new ad-hoc infrastructure based on mobile devices. In essence, mobile users are provided with data storage and processing services on a cloud computing platform rather than on the mobile devices themselves. From a simple perspective, MCC can be thought of as infrastructure where data processing could happen outside of the mobile device, enabling deliver applications similar to what is found in traditional desktop computing environments with any platforms and operation system, including web browsing, email, video conferencing, presentations, movie, music entertainment, (2D or 3D) games and multimedia applications.

Nowadays, the rapid development of MCC promotes the investigation of thin client (remote control) technology. Using thin client system, users are able to remotely control other computers (servers) and delegate actual information processing to them. Thus, thin client technology provides a powerful way to break the barrier between diverse applications and insufficient local hardware/software environment. For example, a mobile device with thin client system permits its user to use the applications running on different mobile platforms (Android, iOS, and so on) or PC platforms (Windows, Linux, and so on).

Thin Client technology is expressed by two vital elements: Remote Control and Platform Isolation technology. Firstly, Remote Control technology is the essential communication method between clients and server. It is defined as a client/server technology that decouples a computer from the mobile devices used to access and interact with it, in particular its monitor, speakers, keyboard, mouse and uses the network to provide a communication channel between mobile devices and the computer (Fig. 1). In this architecture, the Audio/Video (A/V) output from server would not be delivered to locally attached devices (as monitor and speaker), but redirected over the network to a client to be displayed. Secondly, Platform Isolation technology is the fundamental structure to support multi-users on one server. This technology grants permission one server which enables providing separated service based on creating sessions for multi-users. For each user, server isolates resources (input/output) and delivers output (video/audio) to user through the corresponding session.

In this paper, we survey two above technologies in Section 2 and 3, respectively. We then suggest the difficulties and challenges for each technology in Section 4. Finally, we give some conclusions in Section 5.



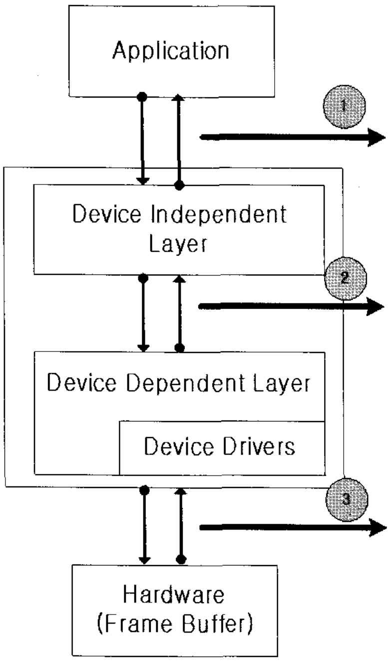

(Figure 1) Remote Control Architecture

## II. Remote Control Technology

A remote control system decouples a server from the mobile devices used to interact with it. In particular, the monitor, speaker keyboard, and mouse no longer need to be directly attached to the physical ports in the server in order to interact with it. Instead, a communication channel is provided between these mobile devices and the server via network connection. A/V output from the server, which would normally be sent to the local video hardware, is instead intercepted and redirected over the network to a client. Similarly, in response to the user interacting with the desktop at the client, input events are generated and sent back to the server. The client and server use a remote control protocol for this back and forth communication.

The remote control technology can be categorized into three distinctive groups based on three layers as (Fig. 2 )shows. This three layer model was first proposed by R.Baratto in his Ph.D. thesis [2]. The original version demonstrates only the display pipeline while we extend the same concept to the audio/input pipeline as well. There exists three interceptions points in the pipeline : (1) between the applications and the device independent layer, (2) between the device independent layer and the device dependent layer, and (3) between the device dependent layer and the hardware layer. To utilize them, the server side must be able handle the application/OS interfaces, the device drivers and the hardware frame buffer, respectively.

At application/OS layer (1), Remote Desktop Protocol (RDP) [1] is a typical protocol developed by Microsoft, which concerns providing a user with a graphical interface to another computer. RDP clients exist for most versions of Microsoft Windows (including Windows Mobile), Linux, Unix, Mac OS X, Android, and other modern operating systems. The advantage of interception at layer 1 is no trouble to isolate data for each application that is beneficial




(Figure 2) Remote Control Layer

way to support multi-users with multi-applications. Nevertheless, the server must work with Microsoft Windows OS environment which support RDP protocol. Besides, using suitable OS and hardware processing capability are required on client side.

At device driver layer (2), THINC uses its virtual device driver to intercept display output in an application and OS agnostic manner [2]. Then server uses that library to convert video/audio to other format and delivery to client. It efficiently translates high-level application display requests to low-level protocol primitives and achieves efficient network usage. Consequently, client' s graphic hardware processing capability is mandatory to translate from data to raw video/audio data aim to display on screen.

At hardware frame buffer layer (3), VNC uses the RFB protocol [3] to remotely control another computer [4]. The benefit of layer 3 interception is that server only delivers the graphical screen (raw data) to the client. Accordingly, client only need to open this raw data without supporting hardware else. However, the big size of raw data causes flooding the bandwidth network. In addition, audio output from all applications is mixed stream, thus it is impossible to isolate audio from each application; this feature supports multi-users connecting to one server. There are some papers that improve VNC for real-time A/V supporting. In [5], a hybrid protocol was proposed to handle multimedia streaming and interactive gaming applications. [10] describes an extension of VNC for effective real-time collaboration through internet. [6] is the new version of VNC enhances video remote display efficiently. Integrating RTP protocol to VNC to provide real-time audio and video transmission for multimedia applications is also presented in [11] [12].

## III. Platform Isolation Technology

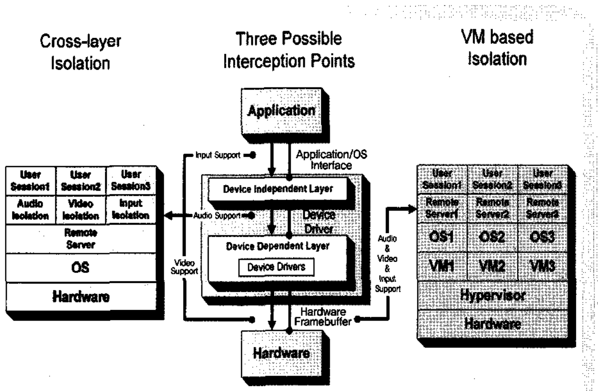
As Section 2 concerns how client/server can communicate together, then in this Section, we focus on the platform isolation technology on server and client (or Virtualization Technology). This technology enables one server supporting many users through separating data section for every user.

A virtual machine (VM) [13] is a tightly isolated software container that can run its own operating systems and applications as if it were a physical computer. A virtual machine behaves exactly like a physical computer and contains it own virtual (i.e. software-based) CPU, RAM hard disk and network interface card (NIC). The essential advantage of VM is multiple OS environments can co-exist on the same computer, in strong isolation from each other. While virtual machines can share the physical resources of a single computer, they remain completely isolated from each other as if they were separate physical machines. If, for example, there are four virtual machines on a single physical server and one of the virtual machines crashes, the other three virtual machines remain available. However, the disadvantage characteristic of a virtual machine is that the software running inside is limited to the resources and abstractions provided by the virtual machine. This drawback limits the number of users connect to platform server. For instance, to support one mobile device, server has to create one instance with Windows XP operation system. This modern OS wastes CPU and memory usages because of some useless functions for mobile devices. To optimize OS for mobile device, an Android OS that supports for PC [7] has proposed named as Android-x86. This OS releases the useless system resources from traditional OS (as Windows 2000 or XP) to adapt for mobile devices. Thus, it can support more users on one server.

The other virtualization technology that supports remote control for multi-users is named as cross-layer isolation technology. (Fig. 3) shows the detail information of cross-layer isolation and the VM based isolation. Unlike the VM based isolation in which the hypervisor takes the responsibility to divide and manage the hardware resources, the cross-layer technique has three isolation components deployed between the user sessions and the remote server. The reason we choose the name "cross-layer" is that the audio isolation, the video isolation and the input isolation are implemented in different layers. To isolate the graphical output of one user session from others, the server assigns a non-overlapping rectangle area for each user session. While hooking the whole screen information from the hardware frame buffer, the server can easily extract any user session from the picture by using the coordinates of the corresponding rectangle. Then using RFB protocol [3], the updates are distributed to the users continuously.

The audio isolation utilizes device driver interception. Prior to the audio signals from multiple user sessions are mixed, the server hooks them and sends to the corresponding clients. On one hand, it is very hard to extract one session' s audio information from a mixed audio signal retrieved from hardware frame buffer; on the other hand, the mobile clients are able to provide the hardware processing capability for audio, RTP (Real-time Transport Protocol) is adopted as the protocol for audio transmission [8].

Unlike video and audio isolation, the input isolation intercepts on the client side and performs the actual input on the server side. The input isolation on server side needs the APIs of server OS. Taking our implementation on Microsoft Window XP as an example, the input isolation module requires the handle of each user session. The user inputs are managed by a multi-queue system. When the server wants to perform a user input, it first activates the corresponding session using the handle, and then simulates the mouse or keyboard event.



(Figure 3) Cross-layer vs. VM based isolation

## IV. Research Challenge

In this Section, we measured the performance on the set of physical machines with the following hardware and software environment with hardware environment (Intel Pentium Dual-Core CPU E5500 2.80GHz, Physical Memory 2.00 GB + Virtual Memory 4.00 GB, graphic card Nvidia GeForce 9300 GS (256MB) and network: 100Mbps) and software environment (MS Windows XP Professional 32bit SP3 + Windows 7 (for Benchmark test), VMware Workstation 6.5.1, Oracle VM Virtual Box Ver 4.1 (for Android OS), i-bench Benchmark tool.

### A. Remote Control Technology

As describing in Section 2, we measured the A/V performance of VNC (RealVNC [4] and UltraVNC [6]), RDP (for Windows 7) [1]. We also inherit the result of ThinC [2] based on the rate between VNC and ThinC. By capturing network traffic with packet monitor and using a variant of slow motion benchmarking tailored for multimedia applications [14], we are able to measure the A/V data size and A/V quality for each technology. We Compared the quality of video through LAN and WAN network. ThinC and RDP is efficient way through LAN and WAN, however, ThinC requires hardware (graphic card) in client side while RDP is developed by Microsoft and not allow to open 3D graphic functions when we play 3D game. Thus, these two ways are not satisfied with mobile devices for playing 3D applications. Contrarily, VNC technology can support displaying when playing 3D games or other applications. It is a new challenging issue to develop future cloud applications and services.

Ultra VNC presents the good quality A/V in LAN network, but not well in WAN because of data packets loss through transmission. We measured in two modes: normal motion and slow motion. Slow motion presents full video data that transmits through network; it is implemented by sending frame by frame in long enough time so that the frame is not lost through network. Normal motion shows the actual video data sending from server to client through network. RDP shows the balanced data size via LAN and WAN but the speed of video slow down gradually. Comparing with RDP, Ultra VNC and ThinC transmit more data packets while similar in LAN environment. Because of delivering raw data buffer frame, thus the size of packet is larger than data from other layers. Large amount transferring data may affect to the real time applications, and occupy more storage in mobile devices which have limited resources. One possible method to reduce frame packet size is using other video format as Motion JPEG (MJPEG) [15]. MJPEG uses a loss form of intra-frame compression based on the discrete cosine transform (DCT). This mathematical operation converts each frame/field of the video source from the time domain into the frequency. Although the bitrates of Motion JPEG is lower than uncompressed video, it is much higher than that of video formats which use inter-frame motion compensation such as MPEG-1. Thus, reducing data size transmission is the one of challenging issues in Mobile Cloud.

The other difficulty is audio support in VNC. The current VNC version does not enable providing audio because of network bandwidth issue. Thus, enabling audio for VNC client via RTP protocol is the other challenging issue.

### B. Platform Isolation Technology

As we mentioned, to carry multi-users on one server, platform isolation technology should be applied through two kinds of platform: cross layer isolation and VM-based isolation. We chose "Video Player" as the testing application since multiple "Video Player" sessions can be supported by both isolation approaches. We used only one physical machine and increased the number of sessions from 1 to 5. In case of VM isolation, we created VM images using the following settings :

- Hardware setting : Share the CPU, graphic card, network of physical machine, Physical memory 512Mb, Virtual memory 768Mb.
- Software environment : MS Windows XP Professional 32bit SP3, Android OS for PC [7], RFB protocol.

We observed that the resource consumption for each isolation approach as follows. The cross-layer isolation performs much better than the Windows-VM isolation with respect to CPU and Memory consumption. Before the CPU usage reach 100%, the cross-layer isolation can support maximally 4 sessions while the VM based isolation can only support 2 sessions at most. After that, since both approaches crashed, we could not get any available data from the monitor software. As suggested in [9], the adoption of VM will degrade the CPU capacity from 10% to 25% regarding Fixed-point or Floating point benchmark. In our scenario, the observation of CPU degradation was even more than that since the tasks assigned to the VMs were more complex than Fixed-point or Floating point benchmark. The memory consumption of cross-layer isolation also shows the efficiency. The memory consumption of Windows-VM isolation, on the other hand, was predefined when the VM was created. However, as the VMs had guest OSs and remote servers, consuming more memory were inevitable. In order to improve Windows-VM isolation, the lightweight OS should be applied to minimize resource (CPU and memory) for every session. Android OS for PC is an example for developing Android services. Although the resource can be minimized considerably, especially in web-browser applications; but it occupies CPU (over 50% for one session) as playing lightweight game. This is one of disadvantages of this way, thus researchers able to follow this trend to create the own efficient session for each client.

According the early explanation, unlike VM-based isolation, cross-layer isolation can support maximally 4 sessions that presents more efficient than Windows-VM isolation. However, cross-layer isolation meets the difficulty in separating audio/video, input (mouse/keyboard) per-application as running multi-users with multi-applications, enabling the new issues in mobile cloud. The isolation of display can be achieved by using frame buffer hooking with non-overlapping window placement. After hooking the whole screen from the frame buffer of graphic card, the server extracts the unique window area for each corresponding thin client user and delivers the information through certain remote display protocol. The isolation of audio is done by device driver hooking. Since the audio signals from different applications have not been mixed at device driver layer, the server is able to send them to corresponding users separately. A multi-queue component is applied to receive and isolate the input events hooked from clients. To enforce an input, the server first activates the corresponding application, and then simulates the event locally. Integrating remote display, audio play and input functionalities with isolation technology becomes challenging issues to support multi-users on one server.

## V. Conclusion

This paper presented client technology on the server for mobile cloud included remote control technology and platform isolation technology. With the rapidly growth of cloud computing and increasing of mobile devices, mobile cloud is predicted as an augmenting market. Thus, the current difficulties that we mentioned need to overcome aim to construct a new infrastructure for mobile cloud. According to those difficulties, some challenging issues could be listed as follows:

- Supporting 3D graphic feature in thin client technology
- Reducing data size transmission over network
- Enabling audio for VNC client via RTP protocol
- Optimizing resource for VM-based isolation
- Integrating remote display, audio play and input functionalities isolation technology

Depend on each technology, researchers can find the individual way to build up more and more useful mobile cloud services.

### Acknowledgement

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2011-0020515)

## References


[1] Remote Desktop Protocol http://en.wikipedia.org/wiki/Remote_Desktop_Protocol

[2] Ricardo Baratto, "THINC: A Virtual and Remote Display Architecture for Desktop Computing and Mobile Devices", Ph.D. Thesis, Department of Computer Science, Columbia University, April 2011.

[3] "The Remote FrameBuffer Protocol" [Online]. Available: http://tools.ietf.org/html/rfc6143

[4] T. Richardson, Q. Stafford-Fraser, K. R. Wood and A. Hopper, "Virtual Network Computing." IEEE Internet Computing, 2(1), Jan/Feb 1998.

[5] D. De Winter, P. Simoens and L. Deboosere. A Hybrid Thin-Client protocol for Multimedia Streaming and Interactive Gaming Applications. In the 16th Annual International Workshop on Network and Operating Systems Support for Digital Audio and Video (2006).

[6] Ultra VNC. http://www.uvnc.com/

[7] "ANDROID x86". Available: http://www.android-x86.org/

[8] "Real-time transport protocol." [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[9] Han, S.M., Hassan, M. M., Yoon, C.W. and Huh, E.N. Efficient service recommendation system for cloud computing market. Proceedings of the 2nd International Conference on interaction Sciences: Information Technology, Culture and Human. (2009). Pp. 839-845.

[10] Tae-Ho Lee; Hong-Chang Lee; Jung-Hyun Kim; Myung-Joon Lee; "Extending VNC for effective collaboration," Strategic Technologies, 2008. IFOST 2008. Third International Forum on, vol., no., pp.343-346, 23-29 June 2008 doi: 10.1109/IFOST.2008.4602895

[11] Yen-Hsiang Chen; Shu-Song Chen; Shanq-Jang Ruan; , "Integrating Bi-Direction Audio and Video Transmission for UltraVNC," Networked Computing and Advanced Information Management, 2008. NCM '08. Fourth International Conference on , vol.2, no., pp.505-508, 2-4 Sept. 2008 doi: 10.1109/NCM.2008.87

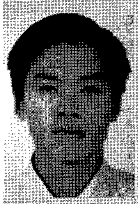
[12] Kheng-Joo Tan; Jia-Wei Gong; Bing-Tsung Wu; Dou-Cheng Chang; Hsin-Yi Li; Yi-Mao Hsiao; Yung-Chung Chen; Shi-Wu Lo; Yuan-Sun Chu; Jiun-In Guo; , "A remote thin client system for real time multimedia streaming over VNC," Multimedia and Expo (ICME), 2010 IEEE International Conference on , vol., no., pp.992-997, 19-23 July 2010 doi: 10.1109/ICME.2010.5582993

[13] "Virtual Machine". Available: http://www.vmware.com/virtualization/virtual-machine.html

[14] Jason Nieh, S. Jae Yang, and Naomi Novik. Measuring Thin-Client Performance Using Slow-Motion Benchmarking. ACM Transactions on Computer Systems, 21(1):87{115, February 2003.

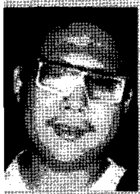
[15] 'Motion JPEG". Available: http://tools.ietf.org/html/rfc2435

## 약 력

**Tien Dung Nguyen**

2008년 HCMC University of Technology Computer Engineering BS
2009년 ~ 현재 Kyung Hee University Computer Engineering Master
관심분야 : Cloud Computing, Thin Client, Multimedia Transmission



**Biao Song**

2008년 Hebei Normal University Computer Science BS
2010년 Kyung Hee University Computer Engineering Master
2010년 ~ 현재 Kyung Hee University Computer Engineering PhD
관심분야 : Cloud Computing, Thin Client

**Tang Wei**

2010년 Hebei Normal University Computer Science BS
2010년 ~ 현재 Kyung Hee University Computer Engineering Master
관심분야 : Cloud Computing, Thin Client, Multimedia Transmission

**이 준 형**

2010년 경희대학교 컴퓨터공학 공학사
2010년 ~ 현재 경희대학교 컴퓨터공학과 석사과정
관심분야 : 클라우드 컴퓨팅, 모바일, Thin Client

**허 의 남**

1990년 부산대학교 전산통계학 공학사
1995년 University of Texas 전산학 공학석사
2002년 The Ohio University 전산학 공학박사
2005년 ~ 현재 경희대학교 컴퓨터공학과 부교수
관심분야 : 클라우드 컴퓨팅, Thin Client, QoS, Security