Journal of Theoretical and Applied Information Technology
30th September 2017. Vol.95. No.18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENERGY-EFFICIENT QOE-AWARE DESKTOP DELIVERY SCHEME FOR DESKTOP AS A SERVICE IN MOBILE CLOUD COMPUTING

[1]XUAN-QUI PHAM, [2]CHOONG SEON HONG, [3]EUI-NAM HUH

[1]Doctoral Student, Kyung Hee University, Department of Computer Science and Engineering, Korea

[2,3]Professor, Kyung Hee University, Department of Computer Science and Engineering, Korea

E-mail: [1]pxuanqui@khu.ac.kr, [2]cshong@khu.ac.kr, [3]johnhuh@khu.ac.kr

ABSTRACT

Recently, mobile cloud computing has emerged as the state-of-the-art mobile service provisioning. As one of cloud computing service categories, Desktop as a Service (DaaS) enables mobile users to access their virtual desktop in the cloud with any DaaS clients, such as smartphones, tablets. A desktop delivery protocol is responsible for providing the communication channels through the network between the DaaS client and the DaaS platform in order to deliver display updates and other interaction information. There are many challenges in developing a desktop delivery protocol. In order to guarantee the Quality of Experience (QoE) of users, the protocol needs to maintain display quality when processing many different service requirements. On the other hand, due to the limited capacity of the battery of mobile devices, the protocol should be energy-efficient in the procedure of delivering DaaS. To address these challenges, in this paper, we propose a desktop delivery scheme that takes into account both QoE and energy consumption on mobile devices. Our scheme reasonably uses VNC module and MJPEG module of the hybrid remote display protocol (HRDP) to deliver display updates. The experimental results demonstrate that the proposed scheme can provide good display quality of different kinds of service to mobile clients with low energy consumption.

Keywords: *Desktop Delivery Scheme, Desktop As A Service, Mobile Cloud Computing, Hybrid Remote Display Protocol, Motion Detection*

1. INTRODUCTION

Nowadays, mobile devices such as smartphone and tablet have become so pervasive that they are gradually becoming an indispensable part of our life as the most effective and convenient communication tools not bounded by time and place. By tapping into the cloud environment, where computing resources are available and abundant, mobile devices can bring various computing capabilities "as-a-Service" residing in the infrastructure side to mobile users transparently. The integration of mobile devices and cloud computing results in a new paradigm named mobile cloud computing (MCC), which is the state-of-the-art mobile service provisioning.

Among those services, Desktop as a Service (DaaS), has drawn more and more attention in recent years. DaaS is defined as a cloud service in which the back-end of a virtual desktop infrastructure (VDI) is hosted by a cloud service provider (CSP). Hardware and software resources of the infrastructure are abstracted from the virtualization technology to be employed efficiently and assigned to a virtual desktop. A cloud service customer (CSC) can access the virtual desktop remotely using any DaaS clients, such as desktop computers, mobile devices and thin clients. With DaaS, the CSC can get all of the benefits of desktop virtualization without all the headaches about the cost and complexity of the infrastructure.

In DaaS, a desktop delivery protocol is responsible for providing the communication channels through the network between the DaaS client and the DaaS platform in order to transfer the exchanged information (e.g. display updates, user input, etc.) [1]. Prior to DaaS, many remote display protocols are popularly used in thin client computing, such as Citrix Meta-Frame [2], Remote Desktop Protocol (RDP) [3], Virtual Network Computing (VNC) [4] and Thin-client Internet Computing (THINC) [5]. Because of their inherent advantages, such as lightweight, cross-platform and low maintenance cost [6], these protocols are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

potential candidates for DaaS. However, the current thin client technologies cannot cope with the requirements of a variety of modern services, most of which are display-intensive applications such as video and game. Conventional VNC-like protocols can guarantee the user's quality of experience (QoE) for services which do not require delivering many display updates, such as office application or Internet browsing. However, it typically suffers from serious quality degradation when dealing with high data traffic requirement of video services. So the user with limited bandwidth will experience desktop stutters or freezes since the frames are skipped or lost and it is difficult to ensure a good QoE for the user. In [7, 8, 9], some hybrid remote display protocols are proposed. In hybrid protocol, there is a concept called motion-rate, which indicates the number of changed pixels between adjacent frames. Depending on the motion-rate of the screen, the system will deliver the updates to the client either through the RFB protocol or through the H.264 streaming [8]. However, H.264 encoding and decoding cause high server and client CPU consumption respectively. In addition, it is not possible to adjust the size or resolution of the display area in the server at run time since it will cause re-streaming.

To solve the above problems, our previous work [10] proposed a hybrid remote display protocol (HRDP) for mobile thin client system. In HRDP, VNC module is chosen to handle the slow-motion display while MJPEG module is adopted for high-motion display. In addition, Graphics Processing Units (GPU) of server was utilized to do a part of JPEG compression task. The experimental results of that paper showed that the proposed protocol provided effective compression to reduce network bandwidth requirements, low-latency and guaranteed quality of video. Nevertheless, the original HRDP [10] lacked a quantifiable holistic solution in view of user's QoE. Neither QoE measurement nor QoE-QoS relationship was drawn. So, in our other works [11, 12], we presented some analysis to derive a QoE-QoS relationship model of the HRDP that considers both the display quality and the server cost. We adopted the IQX hypothesis [20], which quantifies the QoE of streaming service. The QoE and QoS parameters are connected through an exponential relationship given by $QoE=\alpha.e^{-\beta.QoS}+\gamma$. Based on the derived model, we proposed an adaptive delivery scheme, which calculates the QoE scores of its two module (VNC and MJPEG) and automatically selects the appropriate module to provide optimal QoE of users.

Yet energy consumption on mobile client hasn't been considered in those studies. Although it is certainly important to guarantee display quality of services, energy efficiency for battery-driven devices is even more critical in the sense that a mobile device cannot access any services when its battery becomes exhausted. Device battery lifetime is one of the primary obstacles that MCC research faces [13]. While mobile devices are growing ever more advanced, they're still limited by power. The battery hasn't advanced in decades. Moreover, most of research effort on remote display protocols has focused on how to ensure high quality of remote desktop. Meanwhile, little attention has been paid to the energy consumption of mobile devices in the procedure of delivering mobile cloud services [17].

Motivated by the importance of energy efficiency on mobile devices while delivering high service quality in DaaS, in this paper, we propose a desktop delivery scheme that takes into account both QoE in view of display quality and energy consumption on mobile devices. In our scheme, VNC module and MJPEG module of the hybrid remote display protocol (HRDP) are reasonably adopted. To achieve energy efficiency, the strategy is to lower the CPU load or to keep higher CPU idle rate of mobile devices. First, for each module of the HRDP, we do some experiments to quantify the display quality and the client CPU idle rate in terms of display updates, which represent different types of service requirement. Based on the derived results, an energy-efficient QoE-aware desktop delivery scheme has been developed. VNC module has the benefit of low client CPU resource consumption while MJPEG module can ensure display quality for high-motion display updates. To select the best module to encode each block, the proposed scheme calculates a utility function, which determines the tradeoff between the display quality and the client CPU idle rate. The utility function is adaptive to the battery level of mobile devices. Specifically, the proposed scheme keeps track of battery level of mobile devices. When the battery becomes low, the proposed scheme can have a bias towards client CPU idle rate to conserve power on the device. Simulation results show that our scheme is more energy efficient than some existing solutions. Moreover, the energy saving is achieved while good display quality is still guaranteed.

The remainder of the paper is organized as follows. In section 2, an overview of related work is given. In section 3, we conduct some experiments to measure the display quality and the client CPU idle rate in terms of display updates for each

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

module of the HRDP. In section 4, the proposed energy-efficient QoE-aware desktop delivery scheme for DaaS in MCC is presented. Then we describe some comparison results with some existing desktop delivery solutions in section 5, followed by our conclusion and future work in section 6.

2. RELATED WORKS

Recent remote display protocols in thin client system can be categorized into three distinctive groups based on where the display updates are intercepted in graphics pipeline [5]. There are three possible layers including (1) the graphics library layer, (2) the display driver layer and (3) the framebuffer layer. Figure 1 shows these interception points.

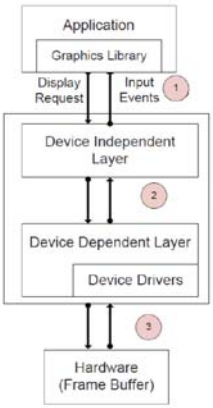

Figure 1: Display Interception Layers

At graphic library (application/OS) layer, only application graphic device-independent commands are transferred and everything below need to be executed on the client. The advantage of intercepting at the highest layer is low data bandwidth consumption. However, the drawback is that the client needs to hold amount of server state and client/server is tightly coupled. In this case, running the user interface on the client and the logic on the server needs continuous synchronization over the network. In high-latency wide area network (WAN), it will cause serious performance degradation. Older thin client system, such as X Window System provide remote display functionality by pushing all user interface processing to the client computer. Meanwhile, more recent remote display systems such as Citrix MetaFrame [2], Microsoft Remote Desktop Protocol [3] allows graphical user interface to be rendered on the host device instead of on the client device, avoiding the need to maintain and run complex window server software at the client.

At device driver layer, the server uses its virtual device driver to intercept display output in an application and OS agnostic manner, e.g., THINC [5]. It efficiently translates high-level application display commands to low-level protocol primitives and achieves efficient network usage. Yet, translation is difficult and depends on client's graphic hardware, especially in mobile thin client environment, where mobile devices have heterogeneous degrees of hardware capability.

At hardware frame buffer layer, the server reads the screen pixels from the frame buffer, encodes the captured screens and transmits the encoded screen updates to the client using remote frame buffer protocol (RFB) [14]. Virtual network computing (VNC) [4] and its derivatives (e.g., TightVNC [15] and TurboVNC [16]) are typical examples of this approach. The advantage of the lowest layer interception is that the client can be very simple and stateless because the server only delivers pixel data to the client. However, it has the critical defect of much more generated display updates data than the above two layers.

In addition to the mentioned solutions, some hybrid protocols are proposed to improve system performance when representing display-intensive multimedia services. Simoens et al. [8] proposed a hybrid remote display that uses the VNC-RFB protocol and H.264 streaming, depending on the amount of motion in the screen images. This solution fully eliminates the high bandwidth consumption issue of VNC encoding. However, H.264 decoding causes high client CPU load that can lead to energy drain from the battery on mobile devices. In our previous works, HRDP was proposed for mobile thin client system [10, 11, 12]. Figure 2 gives a briefly overview of a server in our system. The display updates are hooked from graphic card hardware frame buffer. A high-motion detection divides the whole screen desktop into low-motion areas that will be delivered by VNC module, and high-motion areas that will be delivered by MJPEG module. However, the main objective of these works is to provide the optimal QoE to users. It doesn't take energy efficiency of mobile devices into account.

Besides, there are also researches on energy consumption problem of mobile devices in the procedure of delivering mobile cloud service. Eom et al. [17] proposed a power-aware remote display framework using a hybrid encoding scheme in VNC. Based on the investigation of the correlation between energy consumption and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

various encoding types of VNC, the scheme switches between encoding modes adaptively to the battery level of client devices. However, this solution doesn't take into account the display quality of services. Despite energy-efficiency, using only VNC encoding cannot guarantee the QoE of multimedia services. Meanwhile, Jo et al. [18] proposed an adaptive remote display framework which includes two modes, Screen Mirroring Mode (SMM) and Content Mirroring Mode (CMM). The SMM is the same one used in typical remote display solution. The CMM sends the content instead of screen to the display device if the content is supported by the display device. The scheme will select one of the mirroring mode based on the comparison result of the expected network bandwidth. This approach can improve energy efficiency by supporting CMM. However, the drawback is that it depends heavily on the client capability to enable the CMM.

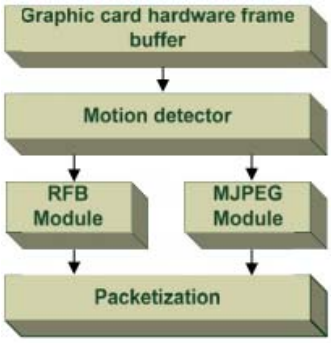

Figure 2: Architecture of The HRDP At The Server

3. PRELIMINARY EXPERIMENT AND ANALYSIS

In DaaS environment, different types of service need different bandwidth requirements to ensure the expected service quality. The bandwidth requirement is directly generated by display updates. For example, if a user watches a multimedia video, the multimedia service will transmit much more display updates and thus consume more bandwidth than the office application service. In our desktop delivery scheme, we will set corresponding pixel updates as x variable. In order to mimic the real network environment, we will set a typical value of network throughput to constraint the maximum network capacity in the experiment. We then use VNC module and MJPEG module of our HRDP to deliver display updates to the users respectively.

In our desktop delivery scheme, display quality is an important factor to evaluate the QoE of DaaS in MCC. On the other hand, the limited capacity of the battery of mobile devices is still a major roadblock, which requires an energy-efficient desktop delivery scheme. One of the potential solutions is trying to keep the client CPU load as low as possible. Thus, we will use the display quality and client CPU idle rate to represent the performance of our system.

In the following sub sections, we first measure our desktop delivery system performance in view of the above metrics with different service requirements. Then we will quantify our system performance in terms of display updates to obtain performance formula for each module of the HRDP.

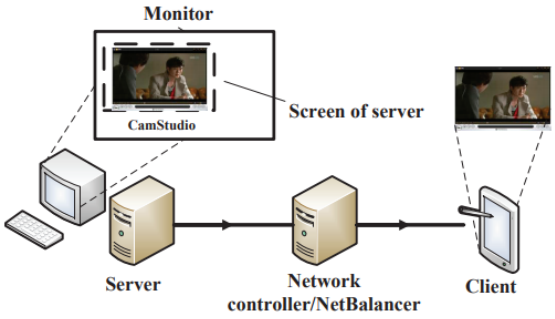

Figure 3: The Experiment Setup for HRDP Measurement

Table 1: Hardware Components

Server	Client
Windows Server 2008	Samsung Note 3, Android 5.0
Intel Xeon CPU X3430 @2.40 GHz	Qualcomm Snapdragon Quad-core @2.30 GHz
8.00 GB RAM	3.00 GB RAM
NVIDIA Quadro FX 3800	Adreno 330
	Li-Ion 3200 mAh battery

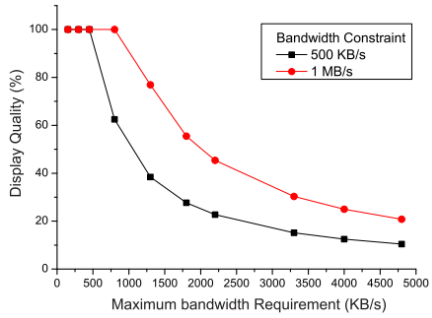

Figure 4: The Display Quality – Bandwidth Requirement Relationship

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

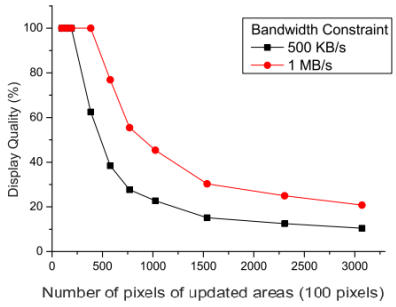

Figure 5: The Display Quality – Display Updates Relationship

3.1 Analysis of Display Quality

To perform the experiment, we first need to set the network bandwidth constraint to simulate the maximum available bandwidth in real network environment. Since the peak throughput of WAN in South Korea can reach 1 MB/s when watching online HD video, we adopt this value as bandwidth constraint. In order to establish different service requirements, we play different motion-rate videos in the server, in which the motion-rate gradually varies from small to large. And then we use the HRDP to deliver these videos to the client where we measure the display quality.

As shown in Figure 3, the experiment setup includes three machines. On one machine, we prepare different motion-rate videos by installing CamStudio to record the video screen. To guarantee the network bandwidth constraint, we configure another machine as a router, which use Netbalancer to control the traffic flows through the system. And a mobile device is used as a client to display the screen updates. The system's hardware information is described in Table 1.

To evaluate the display quality of the client, we employ the slow-motion benchmarking proposed by Nieh et al. [19]. The technique is based on network traffic. The server plays a video sequence slow enough to ensure that there is enough time for every frame to be completely delivered and rendered at the client. The traffic for each frame in this slow motion case is recorded as a reference traffic load for perfect playback without discarded video frames. Then we play the video at normal speed and the traffic load is also recorded. In this case, an amount of data is discarded. According to [6], the video quality VQ at a given specified playback rate P is determined as follows:

$$VQ = \frac{DataTransferred(P)}{DataTransferred(slow-motion)} \quad (1)$$

In our experiment, a HD movie clip is selected. By using CamStudio, we record the clip with same content in different sizes varying from 120x80 to 640x480 pixels. Then we play these videos in the server with unlimited network environment using VNC module. With this condition and the ability of no lossy compression for display updates in the VNC module, we can measure the maximum bandwidth requirement for each video size. In this way, we can get different bandwidth requirements representing different clips in our desktop delivery system. Moreover, the traffic load in unlimited network environment can be recorded as a reference traffic load for perfect playback. Thus, we can transform formula (1) into bandwidth base and get formula (2) to measure the display quality in 1MB/s network.

$$VQ = \frac{DataTransferred(1MB/s)}{DataTransferred(Unlimited\_Network)} \quad (2)$$

Table 2: Display Quality and Video Size Relationship (1MB/s)

Video size (pixels)	Maximum bandwidth requirement	Video quality (%)
120x80	150 KB/s	100
120x120	300 KB/s	100
160x120	450 KB/s	100
240x160	800 KB/s	100
240x240	1.3 MB/s	76.9
320x240	1.8 MB/s	55.5
320x320	2.2 MB/s	45.4
480x320	3.3 MB/s	30.3
480x480	4.0 MB/s	25.0
620x480	4.8 MB/s	20.8

Table 2 shows the maximum bandwidth requirements for different video sizes and the corresponding display qualities that we obtain in 1 MB/s network environment. As we set data traffic amount as x-axis, whereas video display quality as y-axis, Figure 4 gives us a clearly demonstration about the relationship of video display quality with increasing maximum bandwidth requirement. Our experiment also provides a control group at limited bandwidth of 500KB/s to illustrate this general relationship.

As we can see, the display quality of our desktop delivery system heavily relies on actual bandwidth requirement. If the maximum bandwidth requirement is lower than the bandwidth constraint value (here is 1MB/s), the display quality is quite good (100%). Otherwise, the display quality decreases exponentially when the bandwidth

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

requirement increases. We know that the bandwidth requirement is caused by display updates in the server. Hence, we calculate the display updates corresponding to bandwidth requirement of each video size. Figure 5 reveals the relationship between the display quality with the increasing updates size in 100 pixels units when using VNC module. On the other hand, in case of using MJPEG module, which provides efficient data compression technique, we can set the compression rate to ensure that the amount of generated data per second under 1MB/s. In this case, the actual maximum bandwidth requirement is around 500-700 KB/s for a video of size 640x480. And thus, the MJPEG module always guarantees the display quality at the client.

Based on the observed exponential decrease of system performance in VNC module, we use the generic model proposed by Fiedler et al. [20] to derive the display quality in terms of display updates. By using MATLAB curve fitting tool, we obtain the fitting function according to exponential model with 95% confidence bounds. The coefficient of correlation is R=0.9966 and the root mean squared error is RMSE=1.5803. The QoE model in the view of display quality is given by:

$$f_0(x) = \begin{cases} 100 & MJPEG\_module \\ 100 & VNC\_module(x < 380) \\ A_0 * e^{-B_0 x} + y_0 & VNC\_module(x \geq 380) \end{cases}$$

with $A_0 = 167.20$, $B_0 = 0.00198$, $y_0 = 21.95$ (3)

where x denotes the pixels of updated areas in 100 pixels unit, $f_0(x)$ denotes system performance in terms of display quality. The threshold of 380, which denotes 38000 pixels, is estimated from Figure 5. Finally, the fitting function model of display quality is illustrated in Figure 6.

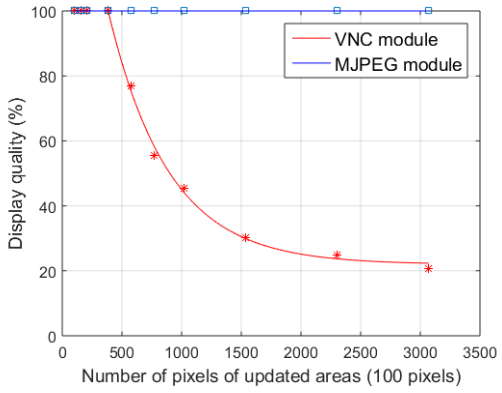

Figure 6: The Display Quality – Display Updates Function Fitting Model for 1 MB/s bandwidth requirement

3.2 Analysis of Client CPU Idle Rate

From previous experiment, we see that the display quality is affected by the bandwidth requirement and the corresponding display updates. Thanks to the effective data compression, MJPEG module can provide better display quality of high motion-rate videos at the client than VNC module. However, it comes with the potential problem of CPU overhead to process the data at the client. In this section, we will study about the relationship between the client CPU idle rate and display updates.

The experiment setup is reused. We play different motion-rate videos in the server to represent different service requirements and then measure the client CPU idle rate when using the two modules of HRDP, which are VNC with lossless compression and MJPEG with lossy compression, to deliver display updates. To profile the CPU usage on mobile device, CPU monitor tool [21] is utilized. We can observe that the VNC module is very efficient regarding client-side CPU consumption since the CPU idle rate remains steady in the range of 90-95%. For MJPEG module, it consumes more CPU power than VNC module to decode the data. And when the video size increases, the CPU idle rate reduces. We also use the exponential function [20] to fit the changing trend of CPU idle rate when using each module. Both are with 95% confidence bounds. The correlation coefficients of VNC module and MJPEG module are 0.9709 and 0.9909 respectively. And the root mean squared error of VNC module and MJPEG module are 0.1754 and 0.3736 respectively. Equation (4) shows the system performance in view of client CPU idle rate with the given number of updated pixels denoted by x. And the fitting function model of client CPU idle rate is illustrated in Figure 7.

$$f_1(x) = \begin{cases} A_1 * e^{-B_1 x} + y_1 & VNC\_module \\ A_1' * e^{-B_1' x} + y_1' & MJPEG\_module \end{cases}$$

with $A_1 = 3.20$, $B_1 = 0.00178$, $y_1 = 92.32$,
$A_1' = 14.34$, $B_1' = 0.00054$, $y_1' = 73.57$ (4)

Journal of Theoretical and Applied Information Technology
30th September 2017. Vol.95. No.18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

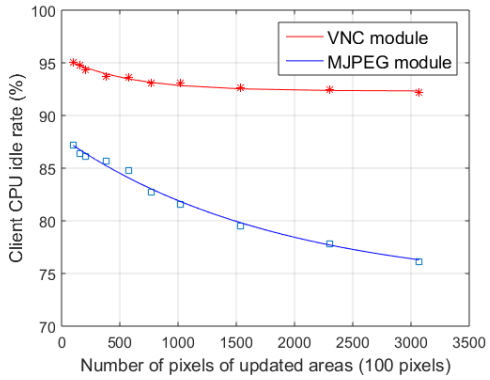

Figure 7: The Client CPU Idle Rate – Display Updates Function Fitting Model

4. PROPOSED SCHEME

Based on the study of the display quality and client CPU idle rate in terms of display updates when using the two modules of HRDP to deliver desktop screen, an energy-efficient QoE-aware desktop delivery scheme has been developed. The scheme first divides the whole screen into non-overlapping blocks. Then it will choose the best fit between the VNC module and the MJPEG module to encode each block. In case of low-motion parts (x<380), one can easily see that VNC module is always better than MJPEG module since it can guarantee display quality without much client CPU overhead. However, in case of high-motion parts (x>380), it becomes a tradeoff between these two factors. The MJPEG module can provide better display quality, but also consume more client CPU resource than VNC module. Now the problem is to define which module we should use to achieve better tradeoff. For each module, we define a utility function denoted by U as follows:

$$U = \alpha_0 f_0(x) + \alpha_1 f_1(x) \quad (5)$$

where $\alpha_0 + \alpha_1 = 1$

The weighting coefficients α0 and α1 indicate which factor has more influence on system's performance. Initially, we set $\alpha_0 = \alpha_1 = 0.5$. However, when the battery level of the mobile device is low, we can have a bias towards client CPU idle rate to conserve power on the device by setting α_0 smaller than α_1. Here, we set the coefficients adaptively to the battery level of the client device. For example, if the battery level of the client device is now 30%, we set $\alpha_0=0.3$ and $\alpha_1=0.7$. In our scheme, we keep track of the battery status of the mobile client by modifying RFB client message format in the VNC client. We add one more byte for the battery level in frame-buffer-update-request message as shown in Figure 8 [17]. If the battery level is lower than a threshold, we set α_0=batteryLevel. Then we can calculate the utility function of the two modules. If the utility value of the VNC module is greater than that of the MJPEG module, we will use VNC module to encode the block so as to achieve energy efficiency. Otherwise, MJPEG module will be utilized. It is worth noting that the threshold can be estimated from the graph of discharge curves in Figure 9 [22]. The graph reveals that cell voltage drops rapidly when the percentage of battery discharge reaches over 60%. Thus the candidate threshold of battery level is about 0.4. Finally, our proposed scheme is presented in Algorithm 1.

Algorithm 1: Proposed scheme

1. Divide desktop screen into W*H meshed blocks of dimension m*n pixels (e.g. 16*16)
2. Count the number of changed pixels in each small block and use a matrix CV[H][W] to store
3. For each block (i,j)
   - 3.1. Calculate the variable for 100 pixels unit x=CV[i][j]*H*W/100
   - 3.2. If (x<380) then encode with VNC module
   - 3.3. Else
     - 3.3.1. $\alpha_0=\alpha_1=0.5$
     - 3.3.2. If (batteryLevel<threshold) then α_0=batteryLevel, $\alpha_1=1-\alpha_0$
     - 3.3.3. $U_{VNC}= \alpha_0 f_0^{VNC}(x)+ \alpha_1 f_1^{VNC}(x)$ $U_{MJPEG}= \alpha_0 f_0^{MJPEG}(x)+f_1^{MJPEG}(x)$
     - 3.3.4. If ($U_{VNC}>U_{MJPEG}$) then encode with VNC module
     - 3.3.5. Else encode with MJPEG module

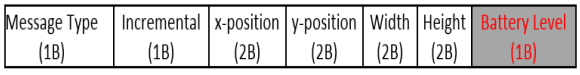

Figure 8: Extended Frame-Buffer-Update-Request Message

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

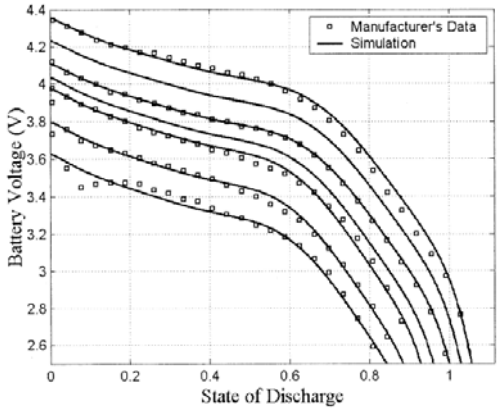

Figure 9: Discharge Curves of A Lithium-Ion Battery At Temperatures of (Top to Bottom) 45°C, 34°C, 23°C, 10°C, 0°C, and -20°C [22]

5. EXPERIMENTAL RESULTS

To validate the proposed scheme, a set of extensive experiments has been conducted. The test environment is same as Section 3. Two use cases of DaaS are considered. The first use case is document operation service, in which a mobile user opens a text editor, then writes some text and includes some figures. The other is video service, in which a user opens a media player to enjoy a video of size 640x480 in 60 seconds.

We compare our scheme with other desktop delivery solutions including RDP, RealVNC, MJPEG streaming, H.264 streaming and the original HRDP [10] in terms of display quality and energy consumption at the client. To gauge energy consumption on mobile device, we use an estimation-based power modeling tool called PowerTutor [23]. The tool runs on background and measure energy consumed by hardware components such as CPU, display, network interface card, etc.

Figure 10 shows the display quality comparison of the mentioned methods. In case of document operation service, all of them achieve very good display quality. The reason is that the maximum bandwidth requirements are less than or approximately equal to the bandwidth constraint (1MB/s) for delivering such low-motion content. However, for video service, in which the data becomes high-motion, the display qualities of RDP and RealVNC dramatically decrease while those of the others are still kept high value. Because of consuming much more bandwidth than the others, RealVNC has the greatest quality loss, which may exceed user's tolerance. RDP also suffers about 50% of quality reduction. Thanks to efficient compression, the display qualities of MJPEG and H.264 streaming are still quite good. Meanwhile, our proposed scheme achieves comparable performance with the original HRDP and both schemes can guarantee over 85% display quality. In fact, there is no much difference in quality perception between these four last mentioned methods.

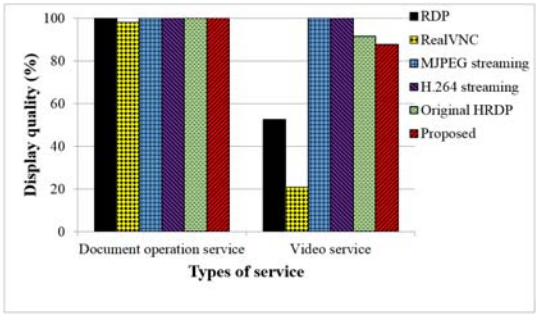

Figure 10: Display Quality of Different Desktop Delivery Solutions

Figure 11 depicts the energy consumption results. Both services have the similar trend of energy consumption when using different protocols. As we can see, RealVNC has the best result for both services. H.264 streaming is the most inefficient solution regarding energy consumption due to high client side decoding complexity of H.264. The MJPEG decoding CPU overhead also causes high energy consumption. A hybrid solution like the original HRDP consumes less energy than MJPEG and H.264. And except RealVNC, our scheme outperforms the others in terms of energy efficiency for both services.

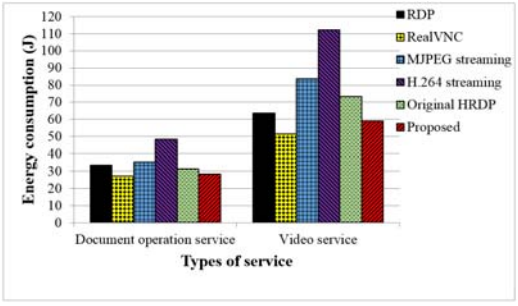

Figure 11: Energy Consumption of Different Desktop Delivery Solutions

From the above experimental results, we have the following observation for desktop delivery. Despite low energy consumption, delivering display updates by RDP and RealVNC

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

will seriously affect the QoE of video service. Meanwhile, if using H.264, we cannot achieve energy efficiency. MJPEG can offer better solution than H.264 since it consumes less energy. To provide better balance between display quality and energy consumption, an appropriate hybrid solution is a promising choice. As mentioned above, the original HRDP [10] is more energy efficient than pure MJPEG and H.264 streaming while it still provides good video display quality. Our proposed scheme can even save more energy since it takes client CPU consumption into account.

To clarify that further, we calculate the energy gains of the proposed scheme by equation (6). Since RDP and RealVNC cannot guarantee the video display quality, we exclude them in this experimental result. Thus, the compared solutions are MJPEG streaming, H.264 streaming and the original HRDP. Gain(A,B) is defined to indicate how much energy can be saved, when we use desktop delivery A instead of desktop delivery B as follows.

$$Gain(A,B) = (E_B - E_A)/E_B * 100\% \quad (6)$$

where E_A, E_B denotes the total energy consumption of A and B respectively.

Figure 12 illustrates the result. The proposed scheme can achieve 26.49%, 45.39%, and 16.24% gains against MJPEG, H.264 and the original HRDP respectively. By saving an amount of energy, the proposed scheme can extend the battery lifetime without compromising much on the video display quality. In other words, the user can enjoy the high quality service for a longer period of time. Hence, user experience in mobile cloud computing settings can be improved.

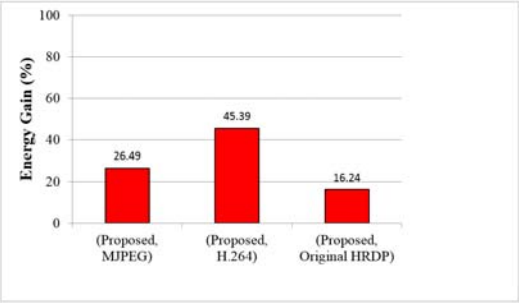

Figure 12: Energy Gain of The Proposed Scheme

6. CONCLUSION AND FUTURE WORK

In this paper, we have considered two aspects of designing a desktop delivery scheme, which are display quality and client's energy efficiency. After exploring the relationships between the display quality and client CPU ideal rate with display updates for each module of the HRDP, we adopt it in developing a desktop delivery scheme that can be adaptive to battery level of client devices to conserve power. The experimental results show the effectiveness of the proposed scheme. Comparing with some existing desktop delivery solutions, our scheme can improve energy efficiency of battery-driven devices while still guaranteeing good display of quality of different kinds of services. However, the experiment is currently only tested in a fixed network environment. Moreover, the MJPEG module uses a specified quality setting to do compression. This manual setting will not always satisfy the display quality requirement in different network bandwidth conditions.

As for future work, we are considering to perform additional experiments in different communication environment in order to verify the adaptation of the proposed scheme. Further, a dynamic quality setting model for MJPEG module should be integrated into the desktop delivery scheme.

ACKNOWLEDGMENT:

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2013-0-00717) supervised by the IITP (Institute for Information & communications Technology Promotion). Professor Eui-Nam Huh is the corresponding author.

REFERENCES:

[1] ITU-T.Y.3503, "Requirements for desktop as a service", *International Telecommunications Union, Geneva, Switzerland*, May 2014.

[2] Citrix, "Citrix Meta-Frame 1.8", *Citrix White Paper, Citrix Systems*, 1998.

[3] Remote Desktop Protocol, https://msdn.microsoft.com/en-us/library/windows/desktop/aa383015.aspx, 2015.

[4] T. Richardson, Q. Stafford-Fraser, K.R. Wood, and A. Hopper, "Virtual network computing", *IEEE Internet Computing*, Vol. 2, No.1, 1998, pp. 33-38.

[5] R. Baratto, "Thinc: A virtual and remote display architecture for desktop computing and mobile devices", *Ph.D. dissertation, Department of Computer Science, Columbia University*, April 2011.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] W. Yu, J. Li, C. Hu, and L. Zhong, "Muse: A Multimedia Streaming Enabled Remote Interactivity System for Mobile Devices", *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, 2011, pp.216-225.

[7] D. D. Winter, P. Simoens, L. Deboosere, F. D. Turck, J. Moreau, B. Dhoedt, and P. Demeester, "A hybrid thin-client protocol for multimedia streaming and interactive gaming applications", *Proceedings of the 2006 International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2006, pp. 15:1-15:6.

[8] P. Simoens, P. Praet, B. Vankeirsbilck, J. De Wachter, L. Deboosere, F. De Turck, B. Dhoedt, and P. Demeester, "Design and implementation of a hybrid remote display protocol to optimize multimedia experience on thin client devices", *Telecommunication Networks and Applications Conference*, 2008, pp. 391-396.

[9] K. J. Tan, J. W. Gong, B. T. Wu, D. C. Chang, H. Y. Li, Y. M. Hsiao, Y. C. Chen, S. W. Lo, Y. S. Chu, and J. I. Guo, "A remote thin client system for real time multimedia streaming over vnc", *IEEE International Conference on Multimedia and Expo*, 2010, pp. 992-997.

[10] B. Song, W. Tang, T. D. Nguyen, M. M. Hassan, and E. N. Huh, "An optimized hybrid remote display protocol using GPU-assisted M-JPEG encoding and novel high-motion detection algorithm", The Journal of Supercomputing, Vol. 66, No. 3, 2013, pp. 1729-1748.

[11] W. Tang, M. S. Kim, and E. N. Huh, "Analysis of QoE guarantee on hybrid remote display protocol for mobile thin client computing", *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, 2013, pp. 118:1–118:8.

[12] M. A. Layek, T. Chung and E. N. Huh, "Adaptive Desktop Delivery Scheme for Provisioning Quality of Experience in Cloud Desktop as a Service", *The Computer Journal*, Vol. 59, 2016, pp. 260-274.

[13] P. Simoens, F. Turck, B. Dhoedt and P. Demeester, "Remote Display Solutions for Mobile Cloud Computing", *Computer*, Vol. 44, No. 8, 2011, pp. 46-53.

[14] The RFB protocol, https://www.realvnc.com/docs/rfbproto.pdf, 2010.

[15] TightVNC Software, http://www.tightvnc.com, 2015.

[16] TurboVNC, http://www.turbovnc.org, 2015.

[17] B. Eom, C. Lee, H. Lee and W. Ryu, "An adaptive remote display scheme to deliver mobile cloud services", *IEEE Transactions on Consumer Electronics*, Vol. 60, No. 3, 2014, pp. 540-547.

[18] H. Jo, and Y. Kim, "An Adaptive Remote Display Framework to Improve Power Efficiency", *International Conference on Networks & Communications*, 2015, pp. 183-193.

[19] J. Nieh, S.J. Yang, and N. Novik, "Measuring thin-client performance using slow-motion benchmarking", *ACM Transactions on Computer Systems (TOCS)*, Vol. 21, 2003, pp. 87–115.

[20] M. Fiedler, T. Hossfeld and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service", *IEEE Network*, Vol. 24, No. 2, 2010, pp. 36-41.

[21] CPU monitor, https://play.google.com/store/apps/details?id=com.bigbro.ProcessProfiler&hl=en.

[22] L. Gao, S. Liu and R. A. Dougal, "Dynamic lithium-ion battery model for system simulation", *IEEE Transactions on Components and Packaging Technologies*, Vol. 25, No. 3, 2002, pp. 495-505.

[23] K. Saipullah, A Anuar, N. A. Ismail and Y. Soo, "Measuring power consumption for image processing on android smartphone", *American Journal of Applied Sciences*, Vol. 9, No. 12, 2012, pp. 2052-2057.