A SURVEY OF MOBILE CLOUD COMPUTING FOR RICH MEDIA APPLICATIONS

YI XU AND SHIWEN MAO, AUBURN UNIVERSITY

ABSTRACT

By leveraging Mobile Cloud Computing (MCC), resource-poor mobile devices are enabled to run rich media applications. In this article, we review mobile cloud computing, with focus on the technical challenges of MCC for multimedia applications, and briefly review the prototypes. The article is concluded with a discussion of several open research problems that call for substantial research efforts.

INTRODUCTION

Mobile cloud computing is a new technology of increasing interest from both academia and industry. Advances in cloud computing have made it possible to provide infrastructure, platform, and software as services for users from any computer with an Internet connection. Mobile cloud computing then extends such services to mobile devices. As there are several billions of mobile phone subscribers world-wide, mobile cloud computing has the potential to have far-reaching impacts in the wireless industry and in our society.

The last decades have witnessed tremendous increase in the popularity of video and interactive video services (e.g., video conferencing and online gaming). With the proliferation of mobile devices, there are huge interests for people to watch video and play online games on mobile devices. According to a recent study [1], among all the mobile data traffic across the world, 66.5 percent will be video related by 2017. This number was 51 percent by the end of 2012. As mobile devices are limited by computation, memory, and energy, it may not serve as platforms for rich media, were it not for cloud applications and services. It is forecasted that cloud applications will account for 84 percent of the total mobile data traffic in 2017, compared to 74 percent by the end of 2012. To this end, mobile cloud computing is a promising solution to bridging the widening gap between the mobile multimedia demand and the capability of mobile devices.

To fully exploit the potential of mobile cloud computing for rich multimedia applications, several key challenges need to be addressed. As the battery life of mobile device is rather limited, energy efficiency is of great importance. Inherent from wireless communications, mobile cloud computing is also characterized by limited and time-varying bandwidth and large network latency. The intermittent network connection may also cause problems for cloud based rich media applications. In face of fluctuating wireless networks and longer response time, how to ensure an acceptable user experience is a challenging issue.

Furthermore, the open air interfaces make mobile cloud computing more susceptible to malicious attacks, and the distributed storage in the cloud may cause privacy issues. Last but not least, the relatively high network service fees may prevent some consumers from using mobile cloud computing for multimedia applications. In this article, we discuss these technical challenges and review proposed solutions in the literature. We then review prototyping approaches and discuss open research problems in supporting rich media applications with mobile cloud computing that call for substantial research.

The remainder of the paper is organized as follows. First, we discuss technical challenges and proposed solutions of MCC for multimedia application. Existing prototypes are then reviewed. We conclude this article with a discussion of open problems.

TECHNICAL CHALLENGES

OVERVIEW

By integrating mobile computing and cloud computing, mobile cloud computing inherits the advantages of both technologies. The basic mobile cloud computing concept and architecture is illustrated in Fig. 1. Mobile cloud computing can be simply viewed as cloud computing for mobile devices, where both data storage and processing can be outsourced to the cloud. In this way, mobility is well supported while the capability of mobile devices is greatly enhanced by the cloud to enable rich media applications.

In this section, we briefly discuss the challenges need to be addressed and review proposed solutions. These challenges mainly stem from the limitation of mobile devices and unreliable wireless connections. On one hand,

This work is supported in part by the US National Science Foundation (NSF) under Grants CNS-1247955 and CNS-0953513 and through the NSF Broadband Wireless Access & Applications Center (BWAC) at Auburn University.

1536-1284/13/$25.00 © 2013 IEEE

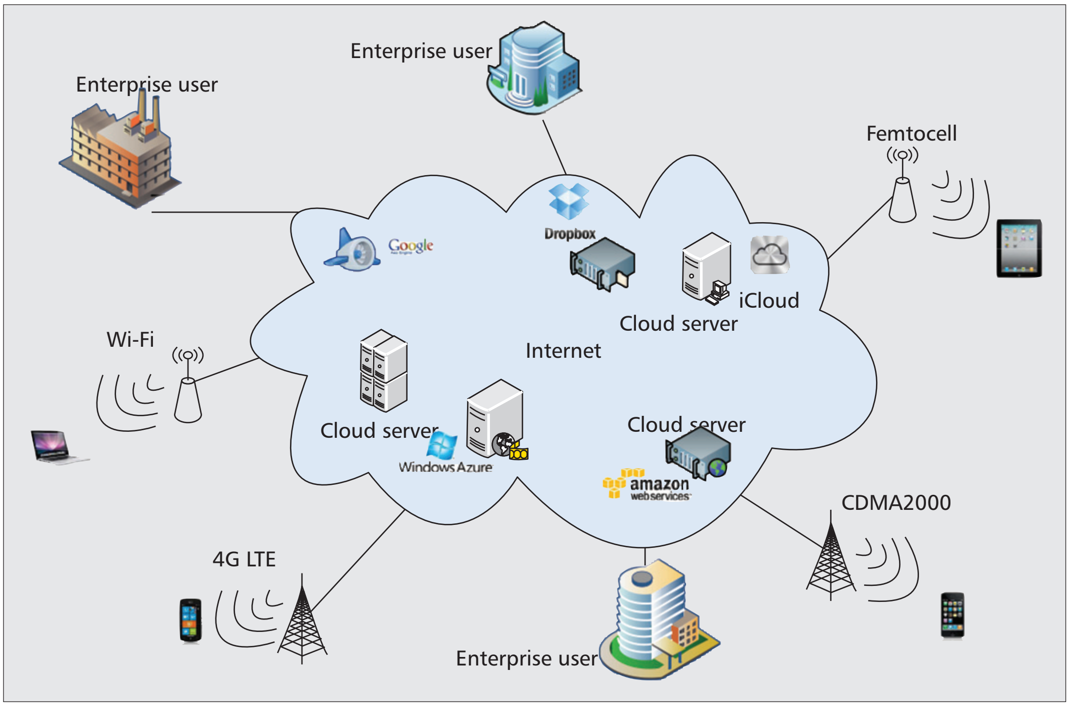

Figure 1. *Illustration of the basic mobile cloud computing concept.*

Limited battery life has been found as the biggest complaint for smartphones. Two main factors contribute to the energy problem. One is relatively limited capacity of batteries. The other is the soaring demand for energy-hungry applications (such as video streaming and online gaming).

although the computational capability, storage and battery life of mobile devices (e.g., smartphones) have been greatly enhanced in recent years, they are still comparatively weak in contrast with computers and more powerful servers. For instance, it is still difficult to run the popular Multiplayer Online Role Playing Game such as World of Warcraft (WoW) on mobile devices. The storage space of mobile device is also limited in contrast to computers. And many mobile devices, especially smartphones, need to be charged almost daily.

Meanwhile, mobile cloud computing relies on wireless networks (e.g., 3G and Wi-Fi) for data and control between the cloud and mobile devices. Compared with fixed and wired networks, wireless networks have limited bandwidth, probably longer latency, and intermittent connectivity. Moreover, under the presence of more mobile devices, the bandwidth available to each device will be further reduced, and network latency can go up and response time for mobile users can be larger.

In summary, the following list of technical challenges needs to be addressed for mobile media clouds.
- How to offload computing tasks?
- How to conserve energy for mobile devices?
- How to cope with the limited bandwidth, intermittent connectivity and longer latency?
- How to ensure the quality of experience (QoE) of mobile users?
- How to ensure security and protect the privacy of mobile users?
- How to minimize the network access cost?

It is worth noting that rather than being independent, these challenges are intertwined and even conflict with each other. For instance, we could save energy by scheduling communicating when the channel is good, which may leads to large latency. And many aspects of the system is related to energy saving, even security [11]. Thus, many existing studies try to achieve an optimal tradeoff among several objectives.

Energy Efficiency

One of the greatest challenges for mobile cloud computing is energy efficiency. In fact, limited battery life has been found as the biggest complaint for smartphones [2]. Two main factors contribute to the energy problem. One is relatively limited capacity of batteries. The other is the soaring demand for energy-hungry applications (such as video streaming and online gaming) [5]. To make the best use of limited battery, especially for multimedia applications, offloading is proposed and investigated in the literature.

Offloading or Local Execution — With the assistance of the cloud, multimedia applications can be either carried out in the mobile device or in the cloud. Executing computation-intensive applications consumes lots of power at the mobile device. One may think that all the computing-intensive applications should be offloaded to and executed in the cloud for energy conservation. However, bulky data transmissions, especially under unfavorable wireless channel conditions, could also consume a large amount of battery power at a mobile device.

Therefore, given the size of data packet L, the wireless channel condition and the application completion deadline T, there is an optimal policy to determine where to execute the current application so that the energy consumption of the mobile device is minimized. Wen, et al investigated this problem in [2]. For mobile execution, the energy consumption for each operation is proportional to the square of voltage supply

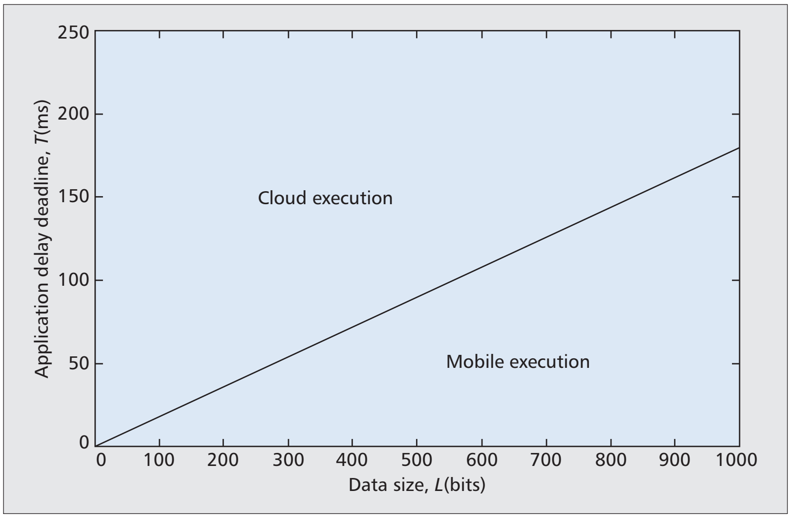

Figure 2. *Optimal execution policy when* n = 4 *([2], ©2013 Wen).*

(i.e., V^2), which is further proportional to the CPU clock frequency f. So the energy per operation is given by

$$\varepsilon_{op} = \kappa f^2, \quad (1)$$

where κ is an capacitance parameter. If the application is executed in the mobile device, the energy consumption can be minimized via optimally scheduling the CPU clock frequency by solving the following problem.

$$\varepsilon_m^* = \min_{\psi \in \Psi} \{\varepsilon_m(L, T, \psi)\}, \quad (2)$$

where $\psi = \{f_1, f_2, \ldots, f_W\}$ is the clock frequency vector.

For cloud execution, the energy consumption is related to the data transmission rate and channel condition, which can be modelled as

$$\varepsilon_t(s, g, n) = \lambda \frac{s^n}{g}, \quad (3)$$

where s denotes the transmission rate, g denotes the channel condition, λ denotes the energy coefficient, and n denotes the monomial order ranging from 2 to 5 depending on the modulation scheme. If the application is offloaded to the cloud, the energy consumption can be minimized through optimally scheduling the transmission data rate by solving the following problem.

$$\varepsilon_c^* = \min_{\phi \in \Phi} \mathbb{E}\{\varepsilon_c(L, T, \phi)\}, \quad (4)$$

where $\phi = \{s_1, s_2, \ldots, s_T\}$ is the data rate vector.

For given data size, delay deadline, and wireless channel conditions, the optimal execution policy can be readily figured out by comparing the solutions to Eq. 2 and Eq. 4. For instance, for an application profile $L = 800$ bits, $T = 400$ ms, and $n = 5$, mobile execution consumes 13 times energy than cloud execution. Figure 2 illustrates the optimal policy when $n = 4$.

The problem of whether or not to offload so as to minimize energy cost is also studied in [5] from a different perspective. Previously, there are mainly two kinds of approaches for offloading. The first one relies on the programmer to specify how to partition a program and which states need to be offloaded adaptively. The second approach is to migrate the entire application to the cloud infrastructure. The first one is fine-grained and can lead to considerable energy saving. The second one saves less energy but reduces the burden on the programmer. In order to combine these benefits, MAUI is proposed in [5] to enable fine-grained code offloading to maximize energy savings but with minimal burden on the programmer.

Specifically, with MAUI, programmers annotate the methods of an application that can be offloaded for remote execution in accordance with some rules (e.g., code that implements the application's user interface should not be marked as remotable). MAUI constantly measures the network condition and estimates the bandwidth and latency. When a remote server is available, MAUI uses its solver to determine the cost (e.g., number of states needs to be transferred) and benefit (e.g., number of CPU cycles saved) of offloading the method, and makes a decision for offloading or local execution.

Wi-Fi or 3G — With the evolution of wireless communications and mobile devices, many mobile devices (e.g., iPhone5 and iPad) can use multiple wireless interfaces (e.g., Wi-Fi and 3G networks) simultaneously to transmit data (e.g., videos and photos) or offload an application. Then there comes the question: which network interface to use, Wi-Fi or 3G?

Under many circumstances (e.g., users are at home or in an office), the signal strength of Wi-Fi is stronger than that of 3G networks. Hence, the data rate and energy saving performance of Wi-Fi usually outperform 3G networks. An experiment carried out in [5] showed that smartphones might consume three times more energy using 3G than using Wi-Fi with 50 ms RTT (Round Trip Time), or even five times more energy using 3G than using Wi-Fi with 25 ms RTT. If an HTC Fuze phone (with a 1340 mAH battery) keeps downloading a 100 kB file over a 3G network, its battery dies within 2 hours.

However, 3G networks provide ubiquitous access while the coverage of Wi-Fi is much more limited. Therefore, the balance between Wi-Fi and 3G networks could shift from side to side depending on time and location. How to dynamically assign packets to the Wi-Fi and 3G interfaces? How to adjust packet transmission durations and control the transmit power on each interface according to the current channel conditions, so that the overall energy consumption of the mobile device is minimized?

These important issues are examined in a recent work [3]. The design objective is to minimize the overall energy consumption by optimally allocating packets to two interfaces (Wi-Fi and 3G) and scheduling the transmission durations given the wireless channel conditions, while satisfying a delay requirement. The formulated convex optimization problem in [3] is then decoupled into two sequential sub-problems and solved. Two observations can be made from the results [3].

- The optimal packet allocation policy depends on the noise power ratio of the two interfaces, which follows a "waterfilling" pattern.
- For each interface, transmit the packets with equal duration while filling up the prescribed delay limit.

With these guidelines, given totally 100 packets (each contains 1024 bits), $N_2 = -100$ dBW(noise power of path 2), and delay deadline 50000, the optimal policy and the corresponding energy consumption are shown in Fig. 3. (Note: k_1^* and k_2^* are the optimal number of packets allocated to each interface, N_1/N_2 is the noise power ratio between the two interfaces, and E is the optimal energy consumption.)

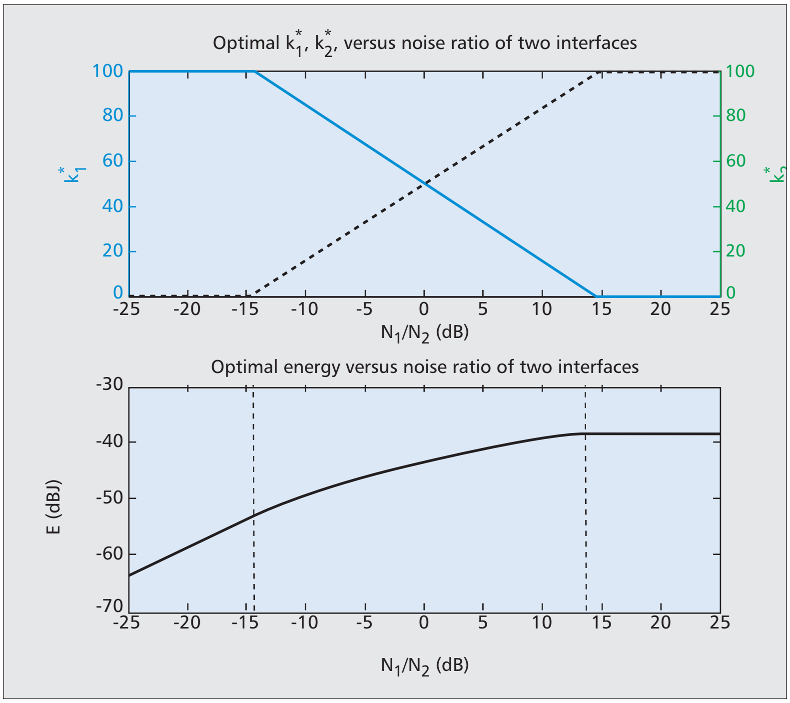

Figure 3. *Optimal packet allocation policy and energy consumption ([3], ©2011 IEEE).*

Living Without Wi-Fi — The previous discussions show that Wi-Fi is a more preferable choice if the mobile user is at favorable locations. When Wi-Fi coverage is not available, one could still use aggregation, compression, and scheduling to reduce energy cost for the cellular network connection [4].

Observing that cellular radio incurs significant delay and energy overhead when switching between the idle and active states, sending data to mobile device in intermittent spurts would consume much more energy. Therefore, it is proposed to merge the data spurts into a single and sustained burst to avoid frequent state switching.

On the other hand, compression is often deployed between the cloud-based proxy and the mobile device to reduce transmission overhead. However, the overall energy saving cannot be achieved without considering the overhead of decompression. The authors in [4] thus propose an asymmetric dictionary-based redundancy elimination technique with minimal decompression overhead. The reduced bandwidth requirement readily translates into conserved energy. It is also observed that 6× higher energy will be consumed with a weak signal than that of a strong signal [4]. Therefore, the authors also introduce a scheduling mechanism to avoid communications when channel condition is poor.

These techniques were implemented in a prototype called Stratus. The test results showed that 50 percent energy saving could be achieved for web browsing using aggregation and compression, and 35 percent energy saving could be achieved for media streaming using aggregation and scheduling.[1]

Ensuring QoE

Conventionally, people play interactive multimedia and multiple player online games on their desktops or laptops. Recently, there are tremendous interests to extend these computation-intensive and latency-sensitive games on mobile devices. The challenge is, with the current PC-based gaming architecture, the storage and computation tasks are mostly carried out by the client's device, while mobile devices are not able to perform such resource-intensive tasks. As a result, currently most online games for mobile devices are light, single-player based, involving a low-level of multimedia, and providing limited user experience.

By leveraging the mobile cloud, the limitations of mobile devices could be readily overcome, making interactive multimedia online gaming feasible for mobile devices. Furthermore, users can start playing a greater number of games without waiting for large files to be downloaded to the mobile device. We examine the problem of ensuring QoE for mobile users using online gaming as an example.

Gaming QoE Modelling — Among mobile cloud media applications, video streaming is one-way communication, while video conferencing and online gaming are essentially interactive two-way communications with tight delay constraints. In [6], a cloud server based mobile gaming approach is presented. The basic idea is to put the computational burden on the cloud server, while the mobile devices just communicate gaming commands to the server.

Although there have been considerable efforts trying to model and understand wireless video streaming, it has been recognized that traditional video quality metrics (e.g., Peak Signal-to-Noise Ratio (PSNR)) cannot be directly applied to model the QoE of mobile online gaming. Mobile online gaming is, by nature, highly interactive and extremely delay sensitive two-way communications. The authors in [6] have identified various objective and subjective factors affecting the cloud mobile gaming user experience (MGUE), as shown in Fig. 4. Besides these factors, game genre also serves as an important factor for determining QoE. For instance, a racing game such as Need for Speed (NFS) has a much more stringent requirement of response time than a role-playing game such as WoW.

Nevertheless, to make the model tractable, not all the factors shown in Fig. 4 are included

[1] *See the demo at: http://research.microsoft.com/apps/video/default.aspx?id=158653.*

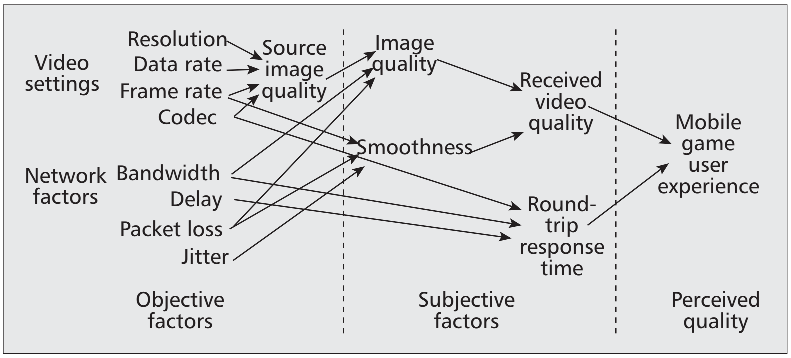

Figure 4. *Factors affecting CMG user experience ([6], ©IEEE 2009).*

in the model. Specifically, only Game Genre, System Configuration (including the codec, resolution, and frame rate), PSNR, Delay, and Packet Loss are included. Game Mean Opinion Score (GMOS) is adopted to quantify QoE, which is a subjective score ranging from 1 (unacceptable) to 5 (excellent). The proposed QoE model is

$$GMOS = f(Game, Config, PSNR, Delay, Loss). \quad (5)$$

According to the ITU-T E-model, it is further formulated as

$$GMOS = 1 + 0.035R + 7 \times 10^{-6}R(R - 60)(100 - R), \quad (6)$$

where $R = 100 - I_C(Game, Config) - I_P(Game, PSNR) - I_D(Game, Delay) - I_L(Game, Loss)$. Further details of I_C, I_P, I_D, and I_L can be found in [6].

The authors then conduct subjective tests. 21 students and staff from UCSD, who had experience of playing these games, participated in the study and provided assessment of their gaming experience. The parameters in the model were determined using linear regression. The model obtained was further validated by another group consisting of 15 participants.

Dealing with Fluctuating Wireless Channel and Busy Cloud Server — The QoE model developed in the previous section can be incorporated into some cross-layer optimization framework for maximizing the QoE of mobile cloud users. For instance, the wireless channel bandwidth may not be sufficient, which may result in large delay of transmission and high packet loss ratio. The cloud server may also be too busy to assist one particular user at a given time, which may result in longer delay and lower rendering frame rate. It is thus important to deal with the fluctuating wireless channel and the available cloud server computational resource for maximizing mobile user's QoE [7].

There are basically two types of cost associated with cloud assisted mobile online gaming:
- The Communication Cost (CommC), which is the bit rate of the game video
- The Computation Cost (CompC), which is the GPU utilization.

On one hand, the game video content complexity greatly affects the video bit rate. Thus CommC can be reduced by bringing down the video complexity. On the other hand, frame time has great impact on CompC and frame time is limited by the bottleneck processor. Thus, CompC can be reduced by alleviating the bottleneck processor load.

Going one step further, four parameters are identified to have impacts on CommC and CompC: Realistic Effect (including color depth, anti-aliasing, texture filtering, and lighting mode), Texture Detail (downsample rate), View Distance, and Environment Detail. So a 4-tuple rendering setting variable S(Realistic Effect, Texture Detail, View Distance, Environment Detail) can be defined. Given all the values of S, their corresponding CommC and CompC are measured by a desktop server with NVIDIA GeForce 8300 graphic card. And there are two opportunities that can be exploited. The first one is by removing unimportant objects in the game video to reduce the video complexity and bottleneck load (hence CommC and CompC). The other one is one could scale CommC and CompC by scaling the texture detail and environment detail.

Gaining these understandings, an adaptation level matrix and its corresponding cost matrix can be constructed. Given current network and cloud server condition, a level selection algorithm can be used to adapt the game video rendering setting to maximize the QoE. For instance, if the network RTT is detected as being greater than a threshold, the algorithm will lower the CommC level by 1, which adjusts S from $S(H,0,60,Y)$ to $S(M,2,300,N)$, meaning from (High Realistic Effect, 0 downsample rate, 60-meter View Distance, Enable Environment) to (Medium Realistic Effect, 2 downsample rate, 300-meter View Distance, Not Enable Environment). Experiments conducted on a commercial UMTS network demonstrate that the proposed rendering adaptation scheme makes the cloud mobile gaming feasible, robust against varying wireless channel and server occupation, hence maximizing the mobile game user's experience.

Dealing with High Response Time — While mobile cloud brings online gaming to mobile devices, it also brings about new problems. One of the problems is probably the longer response time, since game commands and videos need to travel through a stochastic wireless channel. On account of the stringent requirement of response time for online gaming, the risk needs to be well investigated.

In [8], Response Time (RT) is identified as the period from the time one game command is issued to the time the resulting game video is properly displayed. That is:

$$R_T = D_{UL} + D_S + D_{DL} + D_C, \quad (7)$$

where D_{UL} is network uplink delay, D_S is the server delay, D_{DL} is the network downlink delay, and D_C is client delay (including client playout delay and decoding delay). Using the model developed earlier, the RT threshold for Excellent user experience (RT_E) and Acceptable user experience (RT_A) can be calculated. Since delays related to server and decoding video are mainly fixed, D_{UL}, D_{DL} and D_{PL} are considered for minimizing RT.

Given the values of RT_A and R_{TE}, the user acceptable downlink delay threshold (D_{DL}^{TH}) and uplink delay threshold (D_{UL}^{TH}) can be obtained. Then a set of application layer optimization techniques (including a downlink rate-selection algorithm, an uplink delay optimization technique, and a client playout delay adaptation algorithm) is designed to ensure acceptable gaming response time and video quality, and thus acceptable user experience. Experiments carried out in Noisy network conditions, Loaded network conditions and Mobility conditions validate the efficacy of the proposed techniques.

Scheduling Multiple Gaming Sessions — With the mobile cloud gaming framework discussed above, users are enabled to play rich media online games on smartphones and tablets. However, previous discussions are mainly focused on the gaming experience for a single user. Popular online games like WoW usually have several hundreds of thousands players online simultaneously. If there are too many concurrent requests, even if the powerful cloud servers are capable of serving the requests, the wireless links will surely fall short of capacity. Proper scheduling mechanisms thereby should be introduced to support multiple gaming sessions while ensuring acceptable user experience.

To this regard, WCS (Wireless Cloud Scheduler) is proposed in [9]. For each game session, a particular wireless network (denoted as m) and a cloud server (denoted as n) is assigned. The $MGUE(m, n)$ can be calculated using the model discussed earlier. To quantify the cloud service cost, four parameters are identified, which are RT_A (Acceptable Response Time Threshold, as defined previously), CMG_{Comp} (Computing Resource Requirement), $CMG_{Storage}$ (Storage Space Requirement), and $CMG_{DateRate}$ (Video Data Rate Requirement). With given computing price P_C, storage price P_S, and network price P_N, the cloud service cost can be formulated as follows.

$$Cost_{Cloud} = P_C \times CMG_{Comp} + P_S \times CMG_{Storage} + P_N \times CMG_{DataRate}. \quad (8)$$

Another importance metric is Schedule Rate, which represents the percentage of users actually being served. For instance, if there are 1000 requests from users, but only 700 of them are being served, then the Schedule Rate is 70 percent.

The main objectives of the scheduling algorithm are three-fold:

- Satisfying the QoE requirement of game video by keeping $MGUE(m, n) \geq 3.0$ [6]
- Reducing the cloud service cost by keeping $Cost_{Cloud}$ low
- Provisioning of schedule rate as high as possible

The basic idea of the scheduling algorithm is as follows. Firstly, for user i, figure out the possible solution set PS by satisfying three constraints: $Bandwidth(m) > CMG_{DataRate}(i)$, server n supports the requested game genre, and $MGUE(m, n) > 3.0$. If the set PS is empty for user i, this user cannot be scheduled. Otherwise, choose the optimal channel m^* and cloud server n^* by maximizing the prescribed utility function $F(m, n)$.

To further enhance the performance, a joint scheduling-adaptation algorithm is also proposed in [9]. Typically, adaptation techniques in [7, 8] are employed so that users' communication and computation requests are adjusted adaptively in light of the stochastic wireless channel. The simulation results show that with WCS, the schedule rate is enhanced while the average cloud service cost is minimized and the MGUE is maximized.

Considering that system parameters such as user demand rate and server service time might not be available to the scheduler, a Blind Scheduling Algorithm (BSA) is developed in [10]. Specifically, the BSA is formulated as a stochastic minimization problem with fairness ensured. BSA routes new users to the server whose weighted idle time is the longest, then assigns the available server according to the fairness on idle time. It is demonstrated that BSA is asymptotically optimal in minimizing the steady-state waiting time of all the users. In the Halfin-Whitt heavy traffic (HWHT) regime, the heterogeneous server system outperforms its homogeneous MSP counterpart in terms of the user waiting time.

> If there are too many concurrent requests, even if the powerful cloud servers are capable of serving the requests, the wireless links will surely fall short of capacity. Proper scheduling mechanisms thereby should be introduced to support multiple gaming sessions while ensuring acceptable user experience.

OTHER CHALLENGES AND SOLUTIONS

Stochastic Wireless Connection — The challenges incurred by wireless networks have been considered in the previous discussions. For instance, [2] and [3] focus on the energy consumption and delay deadline tradeoff. Reference [8] is particularly about latency. The relationship between RTT and energy consumption is considered in [5]. The problem formulation of [8] has taken into consideration wireless network, network condition, delay and packet loss.

In [15], the authors investigate the resource allocation problem for cloud assisted free viewpoint video (FVV) for mobile devices. FVV requires large bandwidth to transmit video. Given limited computation and battery life of mobile devices, it is highly challenging to enable FVV for mobile devices over cellular networks. However, remote rendering using cloud computing represents a promising solution. From the QoE perspective, users mainly care about view quality and interaction delay. To enjoy optimal view quality, the entire FVV rendering are performed in the cloud. To achieve minimal delay, it is proposed to carry out local rendering on the mobile device for the novel view while waiting for the requested view to come. To minimize the total distortion for the entire viewing period, a rate allocation optimization problem is formulated and solved.

It's worth noting that with the advancement of 4G wireless networks, some of these challenges may be partially alleviated. The IMT-Advanced specifies that the peak rate requirement for 4G systems is 100 Mb/s for high mobility communications (e.g., users in a moving car) and 1 Gbps for low mobility communications (e.g., pedestrians). For real-world speed test, 4G networks are expected to work over 10 times faster than 3G networks. Compared with Wi-Fi, 4G network has comparable speed but much broader coverage, which would be highly desirable for mobile cloud applications.

Although highly promising, mobile cloud computing for multimedia applications is still in its infancy. There are many unsolved problems need to be investigated to fully harvest its potential. There are several potential research directions: energy conservation in the cloud; high definition and uncompressed videos; adaptive QoE provisioning; and security and privacy.

Security and Privacy — The open air transmissions of mobile devices and the distributed storage and processing in the cloud make mobile media cloud applications more vulnerable to malicious programs and attacks. As mobile cloud getting more popular, the security issue becomes more and more important. However, mobile devices may not be feasible to run sophisticated computation-intensive anti-virus programs on the mobile device, due to computational and power constraints. Again, we have to resort to the cloud to detect potential vulnerabilities and threats.

A virtualized in-cloud security service for mobile devices is presented in [11]. Experiments carried out on Nokia's N800 and N95 demonstrate three important benefits of the proposed framework:

- Better detection of malicious software
- Reduced on-device resource consumption
- Reduced on-device software complexity

Network Access Costs — As wireless network services such as 4G LTE are more expensive than traditional wired Internet access or Wi-Fi service, network access cost is also a concern for many mobile media cloud users. If a user watches lengthy movies or playing online games for a long period, the data-usage bill could prohibit him from further participation.

To this end, effective compression schemes could be adopted to reduce the data volume, while scalable coding could be used for users to choose from different media formats that are suitable for the device (e.g., screen size) and budget. Alternatively, the authors in [9] use the proposed WCS and the joint-scheduling-adaptation algorithm to reduce the cloud service costs. Table 1 therein gives a real-world illustration.

Prototyping Efforts

Following the discussions of theoretic investigations, it is also interesting to check out the prototyping efforts. With the support of cloud, Stratus is presented in [4]. It comprises a cloud based proxy on the Windows Azure platform and a client proxy on a Windows Mobile phone with the objective of reducing mobile device energy consumption. Employing multiple virtualized malware detection engines in the cloud, implementations of security software on Nokia N800 and N95 are reported in [11].

MAUI is presented and implemented in [5], which aims at reducing energy costs and making smartphones last longer. Typically, the system contains an HTC Fuze smartphone running Windows Mobile 6.5 with the .NET Compact Framework v3.5, and a dual-core desktop with a 3 GHz CPU and 4 GB of RAM running Windows 7 with the .NET Framework v3.5. A resource-intensive face recognition application, a voice based language translation application which requires much RAM and a latency-sensitive video game are successfully implemented.

Melog is presented in [12]. By analyzing GPS data and photos taken during a trip, realtime automatic blogging is enabled. Typically, the estimated time of creating a travel blog is about 15 seconds and a micro-blog is almost negligible. Muse is presented in [13]. Experiments carried out demonstrate that Muse can reduce network traffic, loading time and response latency of remote display and interaction.

The AMVSC framework is proposed and implemented in [14], which focuses on adaptive mobile video streaming in the cloud. The system has a virtual server with 6 virtual CPU cores (2.66 GHz) and 32 GB memory, and a smartphone Samsung Galaxy II with Android 4.0. Data service is provided by an LTE network. AMVSC shows that cloud computing brings significant improvements to mobile device for video streaming.

Cloud-assisted mobile online gaming are prototyped in [6–8]. Specifically, the wireless network used is either HSDPA networks or 802.11g WLAN deployed at UCSD, and the game server is located in the authors' lab at UCSD. Resource-intensive 3D games such as PlaneShift and online multiple user games such as WoW are successfully launched and played in mobile devices.

Conclusions and Open Problems

Mobile cloud computing can be leveraged to enable multimedia applications on mobile devices. This paper focuses on the technical part of the enabling technologies and discusses the challenges and existing solutions from several aspects such as energy conservation and QoE provisioning. Existing prototypes of mobile media cloud systems are also reviewed.

Although highly promising, mobile cloud computing for multimedia applications is still in its infancy. There are many unsolved problems need to be investigated to fully harvest its potential. We briefly discuss several potential research directions in the following.

- Energy conservation in the cloud: The prior work on energy saving mainly consider the energy consumption on the client side (i.e., mobile devices). However, for the sake of green communication, a joint optimization framework that considers the energy consumption both in the cloud and on the client side should be investigated.
- High Definition (HD) and uncompressed videos: The existing works rarely discuss High Definition (HD) and uncompressed video streaming under bandwidth constrained wireless networks while taking costs and energy into account.
- Adaptive QoE provisioning: As [9, 10] mainly focus on the new requests, it is still an open problem on how to adapt the scheduled session to the wireless network and cloud dynamics for QoE provisioning.
- Security and privacy: As more and more video applications and online games are developed in the mobile cloud, security and privacy issues regarding particular scenario such as video conferencing should be investigated.

References

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017, Cisco White Paper, Feb. 6, 2013, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white paper c11-520862.html.

[2] Y. Wen *et al.*, "Energy-Optimal Execution Policy for A Cloud-Assisted Mobile Application Platform," Technical Report, Nanyang Technological University, Singapore, Sept. 2011.
[3] Y. Wen, G. Zhang, and X. Zhu, "Lightweight Packet Scheduling Algorithms for Content Uploading from Mobile Devices to Media Cloud," *Proc. IEEE GLOBE-COM'11 Wksps.*, Houston, TX, Dec. 2011, pp. 45–50.
[4] B. Aggarwal, N. Spring, and A. Schulman, "Stratus: Energy-Efficient Mobile Communication Using Cloud Support," *Proc. ACM SIGCOMM'10*, New Delhi, India, Aug.-Sep. 2010, pp. 477–78.
[5] E. Cuervo et al., "MAUI: Making Smartphones Last Longer with Code Offload," *Proc. AMC MobiSys'10*, San Francisco, CA, June 2010, pp. 49–62.
[6] S. Wang and S. Dey, "Modeling and Characterizing User Experience in A Cloud Server based Mobile Gaming Approach," *Proc. IEEE GLOBECOM'09*, Honolulu, HI, Nov.-Dec. 2009, pp. 1–7.
[7] S. Wang and S. Dey, "Rendering Adaptation to Address Communication and Computation Constraints in Cloud Mobile Gaming," *Proc. IEEE GLOBECOM'10*, Miami, FL, Dec. 2010, pp. 1–7.
[8] S. Wang and S. Dey, "Addressing Response Time and Video Quality in Remote Server based Internet Mobile Gaming," *Proc. IEEE WCNC'10*, Sydney, Australia, Mar. 2010, pp. 1–6.
[9] S.Wang, Y. Liu and S. Dey, "Wireless Network Aware Cloud Scheduler for Scalable Cloud Mobile Gaming," *Proc. IEEE ICC'12*, Ottawa, Canada, Jun. 2012, pp. 2081–86.
[10] L. Zhou, and H. Wang, "Toward Blind Scheduling in Mobile Media Cloud: Fairness, Simplicity, and Asymptotic Optimality," to appear in *IEEE Trans. Multimedia*.
[11] J. Oberheide *et al.*, "Virtualized in-Cloud Security Services for Mobile Devices," *1st Wksp. Virtualization in Mobile Computing*, Breckenridge, CO, June 2008, pp. 31–35.
[12] H. Li and X. Hua, "Melog — Mobile Experience Sharing Through Automatic Multimedia Blogging," *Proc. ACM MCMC'10*, Firenze, Italy, Oct. 2010, pp. 19–24.
[13] W. Yu *et al.*, "Muse: A Multimedia Streaming Enabled Remote Interactivity System for Mobile Devices," *Proc. ACM MUM'11*, Beijing, China, Dec. 2011, pp. 216–25.
[14] Min Chen, "AMVSC: A Framework of Adaptive Mobile Video Streaming in the Cloud," *Proc. IEEE GLOBECOM*, Anaheim, CA, Dec. 2012, pp. 1–6.
[15] D. Miao *et al.*, "Resource Allocation for Cloud-based Free Viewpoint Video Rendering for Mobile Phones," *Proc. ACM Int'l. Conf. Multimedia 2011*, Scottsdale, AZ, Nov. 2011, pp.1237–40.

BIOGRAPHIES

YI XU [S'11] (yzx0010@tigermail.auburn.edu) received the M.S. degree in Electronic Engineering from Tsinghua University, Beijing, China in 2010 and the B.S. degree in Electronic Information Engineering from University of Electronic Science and Technology of China, Chengdu, China in 2007. Since 2011, he has been pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL. His research interests include Optimization, Game Theory, MIMO, OFDM, IDMA and Cognitive Radio Networks.

SHIWEN MAO [S'99, M'04, SM'09] (smao@ieee.org) received Ph.D. in electrical and computer engineering from Polytechnic University, Brooklyn, NY. Currently, he is the McWane Associate Professor in the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA. His research interests include cross-layer optimization of wireless networks and multimedia communications, with current focus on cognitive radios, femtocells, 60 GHz mmWave networks, free space optical networks, and smart grid. He is on the Editorial Board of IEEE Transactions on Wireless Communications, IEEE Communications Surveys and Tutorials, and several other journals. He received the NSF CAREER Award in 2010 and is a co-recipient of the IEEE ICC 2013 Best Paper Award and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.