2013 IEEE Seventh International Symposium on Service-Oriented System Engineering

Next Generation Mobile Cloud Gaming

Wei Cai, Victor C.M. Leung
Department of Electrical and Computer Engineering
The University of British Columbia
Vancouver, Canada V6T 1Z4
Email: {weicai,vleung}@ece.ubc.ca

Min Chen
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China 430074
Email: minchen@ieee.org

***Abstract*—With the proliferation of smart mobile devices and broadband wireless networks, the mobile gaming market is rapidly expanding among younger generations due to its ubiquitous entertainment features. Cloud-based mobile computing is a promising technology to address the inherent restrictions of mobile devices, such as limited battery lifetime and computational capacity. In this work, we categorize the current approaches to cloud-based mobile gaming and explicitly define Mobile Cloud Gaming (MCG) [1]. Based on its unique features, we propose a framework for next generation MCG systems. Open research issues for MCG are also discussed.**

***Keywords*-mobile; cloud; game; video; architecture; next generation**

I. Introduction

Nowadays, time consumed on mobile devices is increasing compared to desktop computers, not only for work and social activities, but also entertainments. Applications of online social network, videos and games have occupied most of the mobile application market shares. Downloads of mobile games are exploding rapidly.

Similar to desktop games, mobile games are transiting from stand-alone ones to Internet-based multi-player ones. Player-to-player real-time interactions are enabled via high speed Internet connections. Besides, with the development of three dimensional (3D) graphics and better rendering technologies, gaming scenarios are becoming more vivid and realistic. Moreover, embedded sensors (e.g., accelerometers, Global Positioning System (GPS), cameras) are increasingly utilized to provide more flexible gaming user interfaces. These improvements also introduce several challenges in the design of mobile games: i) Satisfying players' expected quality of experience (QoE) during online gaming by provisioning quality of service (QoS) flexibly in heterogeneous network environments, ii) dramatic increase in the demand for computing resources due to advances in 3D rendering, and iii) processing and then transmitting sensory data from mobile clients to the cloud via wireless networks. These challenges are enlarged by limited storage, battery lifetime and central processing unit (CPU) capacity of the mobile devices.

Mobile cloud computing [1] provides a potential solution to these challenges. Well recognized as the next generation computing infrastructure, the cloud is considered to provide unlimited storage and computational resources, which support various types of online services for cloud users. According to different levels of usage, cloud enables IaaS (Infrastructure as a Service), NaaS (Network as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) for users. For mobile cloud computing, a main concern is to utilize the rich resources within the cloud to enhance the functionality of mobile devices and prolong the battery lifetime through better energy efficiency.

As cloud computing evolves to provide Everything as a Service, mobile cloud gaming (MCG) is emerging as a promising form of Gaming as a Service (GaaS). By accessing the rich resources of the cloud, MCG systems are able to overcome the intrinsic constraints of mobile devices. Several studies on MCG issues exist in the literature [2] [3]. However, none of them provides an explicit definition of MCG.

In this paper, we first review the merits and drawbacks of mobile cloud video gaming (MCVG) and mobile browser gaming (MBG), and then provide a definition of MCG in Section II. In Section III we discuss the features of next generation MCG systems and then propose a framework to facilitate its implementation in Section IV. Section V discusses open research issues regarding different components of the framework, including cloud-end, network access, mobile devices, and quality of service and experience (QoSE) in the proposed framework. Section VI illustrates two cases for next generation MCG. Section VII concludes the work.

II. Definition of Mobile Cloud Gaming

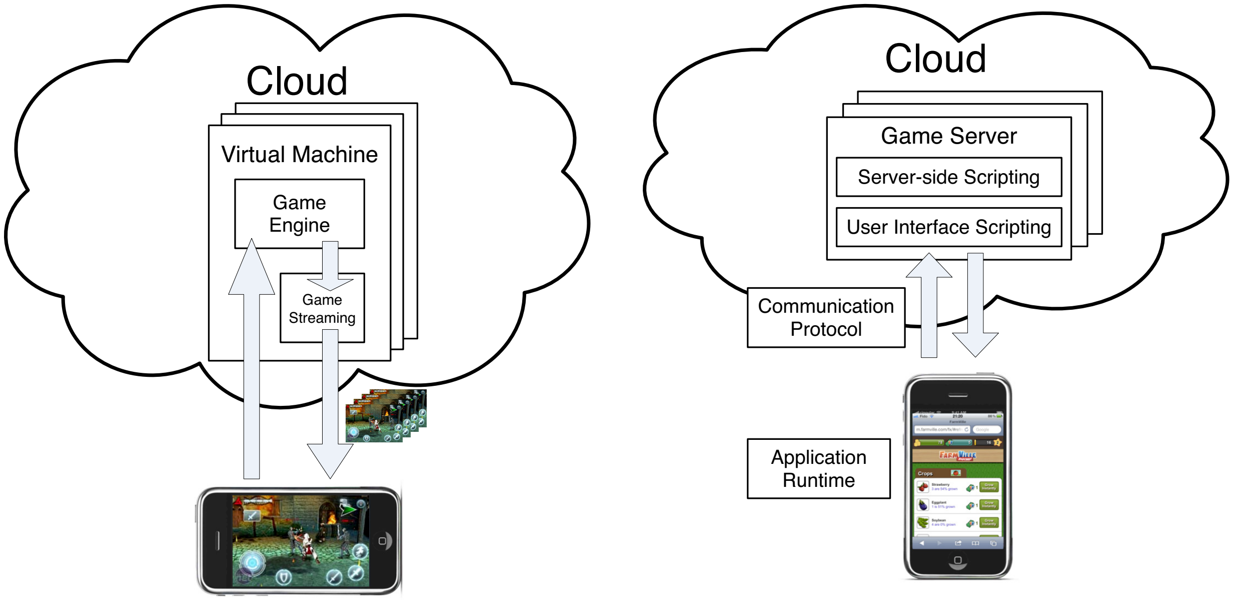
We review and classify existing approaches to MCG into the categories of MCVG and MBG, and then explicitly define MCG.

A. Mobile Cloud Video Gaming

Gaming-on-Demand is an emerging trend of the gaming industry to take advantage of cloud computing. Cloud gam-

[1] Depending on the context, MCG also stands for Mobile Cloud Game.

978-0-7695-4944-6/12 $26.00 © 2012 IEEE
DOI 10.1109/SOSE.2013.30



(a) Mobile Cloud Video Gaming System

(b) Mobile Browser Gaming System

Figure 1. Two Types of Conventional Mobile Cloud Gaming

ing companies, such as OnLive[2], Gaikai[3] and G-Cluster[4], are leading their way to commercialize the cloud video games. In their approaches, video games are executed in their private cloud servers, and the gaming video frames are transmitted over the Internet to interactive televisions or desktop PCs. In reverse, game players' interactions are sent to the cloud server over the same network.

When the Gaming-on-Demand model is applied to mobile devices with wireless communication capability, we have MCVG. Similarly to Gaming-on-Demand, the cloud server initiates virtual machines (VMs) to simulate the runtime environment of the mobile devices, as illustrated in Fig. 1(a). To support mobile gaming on thin clients, the VMs consist of two key components: game engine servers and game streaming servers. When a player accesses the game server from a mobile device, the connection will be first confirmed by the game content server. Subsequently, it initializes a game engine server and analyzes the factors affecting the QoSE for MCG. The game engine server then loads the client's account information and game data from game content server, and starts to process the game logic and user data to render the raw game video. The generated raw game video is encoded by the game streaming server, and finally sent to the mobile client via the wireless connection. On the other hand, the mobile users inputs are delivered to the cloud server, and accepted by the game content server directly [2].

In fact, with MCVG the cloud is intrinsically an interactive video generator and video streaming server, while the mobile devices are the event controller and video receiver. Due to the constraints of mobile communication networks, technologies regarding video rendering, compressing, and transmission QoS control are of great importance in the system design of MCVG [4][5].

At present, the mobile version of OnLive[5] is available in the Google Play Android App market. To the best of our knowledge, it is the first commercial MCVG application.

B. *Mobile Browser Gaming*

Browser games are an emerging type of computer games, many of which are running on online social network sites with massive users. According to [3], browser games have the following characteristics: i) multiplayer support: a massive number of players in the browser game can interact with each other; ii) web browser as client: the games are played directly in the web browsers and do not need software installations (some of them might need extra add-ons); iii) always on: it is possible to get attacked or receive messages even when a player is not online; iv) different time scales: a browser game can go on in real-time, a game can be turn-based, or a game can executive actions on some time scale based on a queue of received commands; v) each player has a single account, through which he or she controls one character, group or nation. Browser games

[2] http://www.onlive.com

[3] http://www.gaikai.com

[4] http://www.gcluster.com

[5] http://www.onlive.com/mobile

have several different types, including strategy games, role-playing games, manager games, shooter games, and social networking games.

However, the most unique feature of browser games is that the web browser is the gaming container. Along with the development of mobile devices, games on mobile browsers are also emerging, which can be referred as MBG.

As depicted in Fig. 1(b), the game server for a MBG runs the game in the cloud, while the web browser on the mobile device serves as the interface for the game player. According to [6], the essential components in a MBG includes: i) *Game Server*: The server-side application, which consists of server-side scripting and user interface scripting, must be robust and fast reacting. Server side scripting using PHP, Perl, Python, JSP, ASP, etc., provides the main contents and functionalities of the game. In contrast, user interface scripting, e.g., using JavaScript and its extension WebGL, concentrates on handling player inputs, including mouse clicks, drags, etc. ii) *Communication Protocol*: To enable the gaming procedure, communications between the server and the mobile browser are of great importance. Current communication formats, such as XML and JSON, are well adopted in MBGs. iii) *Application Runtime*: On the mobile devices, application runtimes should be installed into the web browser as plugins. Adobe Flash player, Adobe Shockwave Player, Microsoft Silverlight, Java applets and Unity web player are some of the most successful commercial application runtime.

At present, most of the successful commercial MBGs strongly depend on the ecosystem of online social networks, e.g,, FarmVille[6] on Facebook.

C. Definition of Mobile Cloud Gaming

To the best of our understanding, an explicit definition of MCG does not exist. We consider MCVG and MBG as two types of MCG. Consequently, we define MCG as follows: *Mobile Cloud Gaming is defined as interactive gaming utilizing mobile devices that access the cloud as an external resource for processing of game scenarios and interactions, and to enable advanced features such as cross-platform operations, battery conservation, and computational capacity improvement.* MCG is characterized by mobile clients that offload the whole game engines to the cloud, and the fact that game developers and players need not be concerned about the construction of the cloud-end nor their detailed interactions with the cloud. In short, the MCG system utilizes the cloud to host games, in order to provide a light-weight gaming solution for mobile devices.

The featured advantages for MCG include:

- **Thin-Client:** There is no need to install and execute gaming engines on a mobile device. Therefore, the mobile devices only have to install a very thin client such as OnLive, which serves as a video recipient and command transmitter. Some of the clients can even be an add-on for browsers, such as Gaikai.
- **Unlimited Resources:** The cloud server hosts the gaming engines, which implies that the game has unlimited storage and other computational resources.
- **Data Security and Seamless Gaming:** The gaming data resides in the cloud-end, which prevents loss of data due to loss or damage of a mobile device. It also implies that whenever and wherever the players connect to the game, the gaming content and status remain the same. Therefore, the players can have a seamless gaming experience across multiple networks and devices.
- **Potential Battery Conservation:** The cloud will deal with the most complicated part of rendering and computing, whereas the client only have to display the gaming video and transmit players' command. It might potentially conserve battery on mobile devices if the gaming process is sufficiently complex.

However, MCG also introduces some drawbacks:

- **Strong Network Dependency:** As GaaS is a cloud service, it cannot be launched without network access. Games cannot be played once the Internet is disconnected.
- **High Bandwidth Consumption:** For MCVG, the cloud server needs to transmit real-time video frames, which is one of the most bandwidth consuming types of traffic. Therefore, the QoS of the network has a large impact on the gaming experience. Moreover, the relative high costs of wide area network (WAN) access, such as 3G, WiMax and LTE, may be an economic burden for GaaS users, which might reduce the customers' interest on MCVG.
- **Resource Limitation on Mobile Browser:** The disadvantages of browser games are magnified on mobile platforms. For instance, in contrast to native games, resources available for browser games could not exceed those accessible by the browser, which may not even be enough for video rendering sometimes.

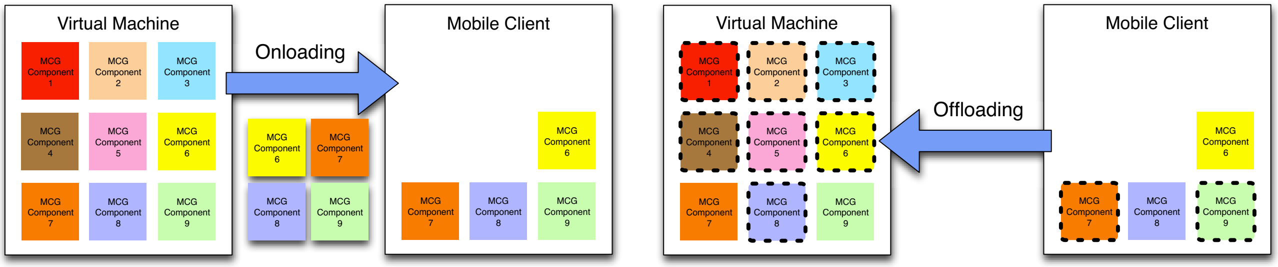
III. Next Generation Mobile Cloud Gaming

In this paper, we propose additional novel features that could be introduced in the design of next generation MCG systems, including dynamic cloud integration, augment-reality gaming, context-aware gaming, multi-player online interaction and seamless cross-platform gaming.

A. Dynamic Cloud Integration

Comparing MCVG and MBG, the only difference is that the graphical rendering is on client side for MBG, while the video frames are generated on the server side for MCVG. The heavy bandwidth consumption of video frame transmissions in MCVG can be replaced by transmissions of graphical command sets in MBG, which consuming less

[6] http://m.farmville.com



(a) Onloading from Cloud to Mobile Client

(b) Offloading from Mobile Client to Cloud

Figure 2. Dynamic Cloud Integration for Mobile Cloud Gaming

bandwidth. The only cost of this bandwidth saving is the requirement of some plug-ins on the browsers to enable graphic rendering. This motivates us to propose dynamic cloud integration for next generation MCG system.

In dynamic cloud integration, a game engine in the cloud is divided into components, including video streamer and others. Components are able to transmit to the mobile devices and execute on the clients as mobile agents. This procedure can be called *onloading*, in contrary to offloading in cloud computing. During a game, the server first launches the game on the cloud and then starts receiving statistics of mobile devices statuses. Based on such information, such as CPU usage, network quality and battery usage, the cloud determines an appropriate proportion of game components to be onloaded to the clients. As shown in Fig. 2(a), MCG components 6, 7, 8, 9 are onloaded to the mobile client, therefore, the functionalities provided by these components can be executed locally.

During the gaming session, the networking environment is ever changing. To overcome these variations and to maintain an acceptable gaming experience, dynamic cloud integration design also enables flexible offloading from mobile client to the cloud. As depicted in Fig. 2(b). Even though the mobile client has previously received copies of components 6 and 8 by onloading, it offloads them to the cloud to reduce the local computing load. Subject to different scenarios, the mobile clients should negotiate with the cloud to achieve an optimization on the offloading percentage of the game engine, which is a tradeoff between network QoS and gaming performance. In fact, MCVG can be considered as a special case of such a tradeoff, in which the cloud hosts all gaming components while the client only serves as an interactive interface for players. In contrast, MBG mode is the opposite case in which all the graphical components are onloaded and executed locally on mobile clients.

B. Augmented-Reality Gaming

In traditional mobile gaming, we call it “mobile” because the games are installed on users’ mobile devices and can be played wherever they go. However, it can hardly be considered “pervasive gaming” in that no one will actually play PlayStation Portable (PSP) or iPhone games while they are walking (actually such a behavior is also not appreciated), since it is really dangerous if the player is absorbed in the game and not aware of her surroundings while moving around. Moreover, the traditional mobile screen is hardly large enough to free the user from holding them closely and focusing on them. In next generation MCG, augmented reality gaming with the assistance of cloud provides a new gaming scenario. Augmented reality is a live, direct or indirect, view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or GPS data. It enables modifications on camera-captured video, so that we can potentially create games on it. The recently invented Google Glass[7] is a very good example of emerging devices that enable augmented-reality gaming. The players are able to see the gaming content directly through the lens, while the gaming content is adapted to the surrounding environment and is delivered from the cloud.

C. Context-Aware Gaming

Mobile devices are increasingly equipped with GPS, accelerometers, cameras, and other sensors, which enable context-aware MCG. In this case the MCG system needs to support the additional tasks of capturing contexts, determining what context-specific adaptations are needed in the cloud, tailoring candidate gaming contents according to the contexts, and feeding back the adapted gaming scenarios. In other words, novel and adaptive gaming contents can be generated according to the time, location and movement of the game players. Context-aware MCG can potentially offer game players a better gaming experience that better fits the current situation of each player.

D. Multi-Player Online Gaming

Online gaming has tremendous potential and is attracting increasing interest compared with single player games on mobile devices, similar to desktop platforms. However, current cloud game providers only utilize the cloud to host traditional video games. Onlive, the leading GaaS provider, was the first to introduce an MCG client in the Android market; however, only single player games are provided. In fact, the cloud is particularly suitable for optimizing multi-player online games. As the information exchange and interaction between players occur in the cloud-end, the system may reduce the video quality for players with weaker network connections in order to maintain the same QoSE, which also helps to reduce network overhead. In addition, online MCG can utilize the scalable resources in the cloud to provide better system scalability in terms of the numbers of players and gaming scenes.

E. Seamless Cross-Platform Gaming

The digital world has been transiting to a human-centric era. Mobile users frequently interact with multiple devices (phone, tablet, laptop). The trend for mobile games is the same. In fact, the game players are only interested in the gaming contents, but not the gaming platform. They are willing to play the same game via different devices in different locations if the contents can be delivered seamlessly. Therefore, providing a seamless experience for real-time games with cross-platform features is critical. The next generation MCG should enable a same game copy in different system, different screen resolutions, and with different input methods. In addition, the gaming system should provide a mechanism to avoid collisions when a player is playing the same game on different platforms. Moreover, it should be simple and fast to suspend and resume a game, so that the player would not perceive the transition.

IV. Framework for Next Generation Mobile Cloud Gaming

Based on the analysis in Section III, we propose a new framework for MCG system as follows. In our proposal, we consider the cloud as an infinitely expandable computing resource and each mobile device in the MCG system have a corresponding VM residing on the cloud.

A common procedure for the next generation MCG is shown in Fig. 3. To start the gaming, the cloud server first launches the game in the cloud and starts receiving status statistics from mobile clients. According to the statistics, the cloud might intelligently dispatch a portion of the game components to the mobile devices. After receiving these components, the mobile devices are able to dynamically negotiate with the cloud to determine the appropriate proportion of onloading and offloading, in order to satisfy the gaming experience. In other words, the proposed gaming system is able to intelligently switch itself from different modes, including MCVG and MBG.

Fig. 4 depicts the architecture for the proposed MCG framework, which has five components as described below: Cloud Layer, Network Layer, Mobile Layer, QoSE and Security.

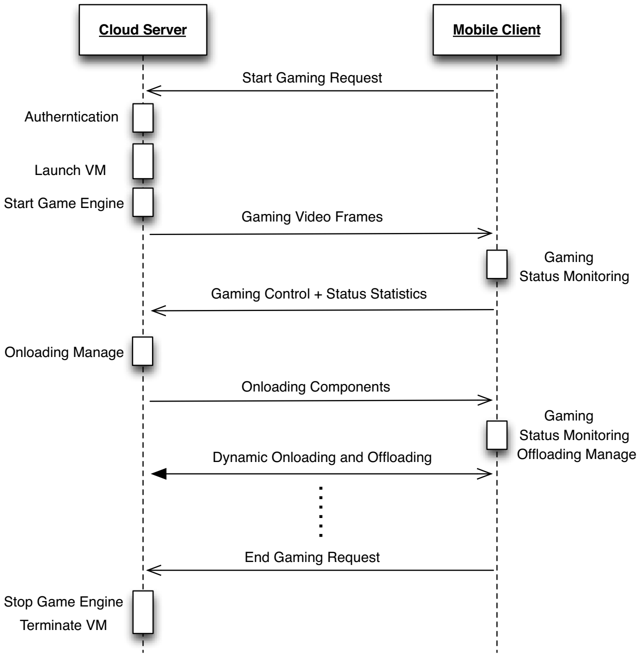


Figure 3. Procedure for Next Generation Mobile Cloud Gaming

A. Cloud Layer

The cloud layer is concerned with the design and implementation of the cloud-end. In mobile cloud computing, the cloud is always assumed to be a stable and rich resource, which is suitable for conducting tasks with heavy computation loads. In fact, the cloud consists of distributed resources, including storage, computing server, etc. In the cloud, these resources should be assembled and the tasks from applications are scheduled to distributed servers automatically. VMs running in the cloud serve as interfaces for mobile devices. These VMs simulate the same hardware and operating systems on mobile devices, which enable them to operate those offloading tasks from the corresponding devices. Once a MCG player initiates a gaming session, the VM should be launched, after the cloud server authenticates the request. The Onloading Manager in the VM facilitates dynamic onloading of the game components. In addition to serving as the game engine, the VMs are also able to communicate with each other, either through the central information exchange of Online Gaming Server, or in an ad-hoc manner. Consequently, the interactions between players can be handled within the cloud-end.

B. Network Layer

The network layer is concerned with efficiently exchanging data packages between mobile devices and the cloud. Information exchange between the cloud and the mobile client is frequent and bandwidth consuming. To enable the mobility of players, wireless networking is mandatory. In

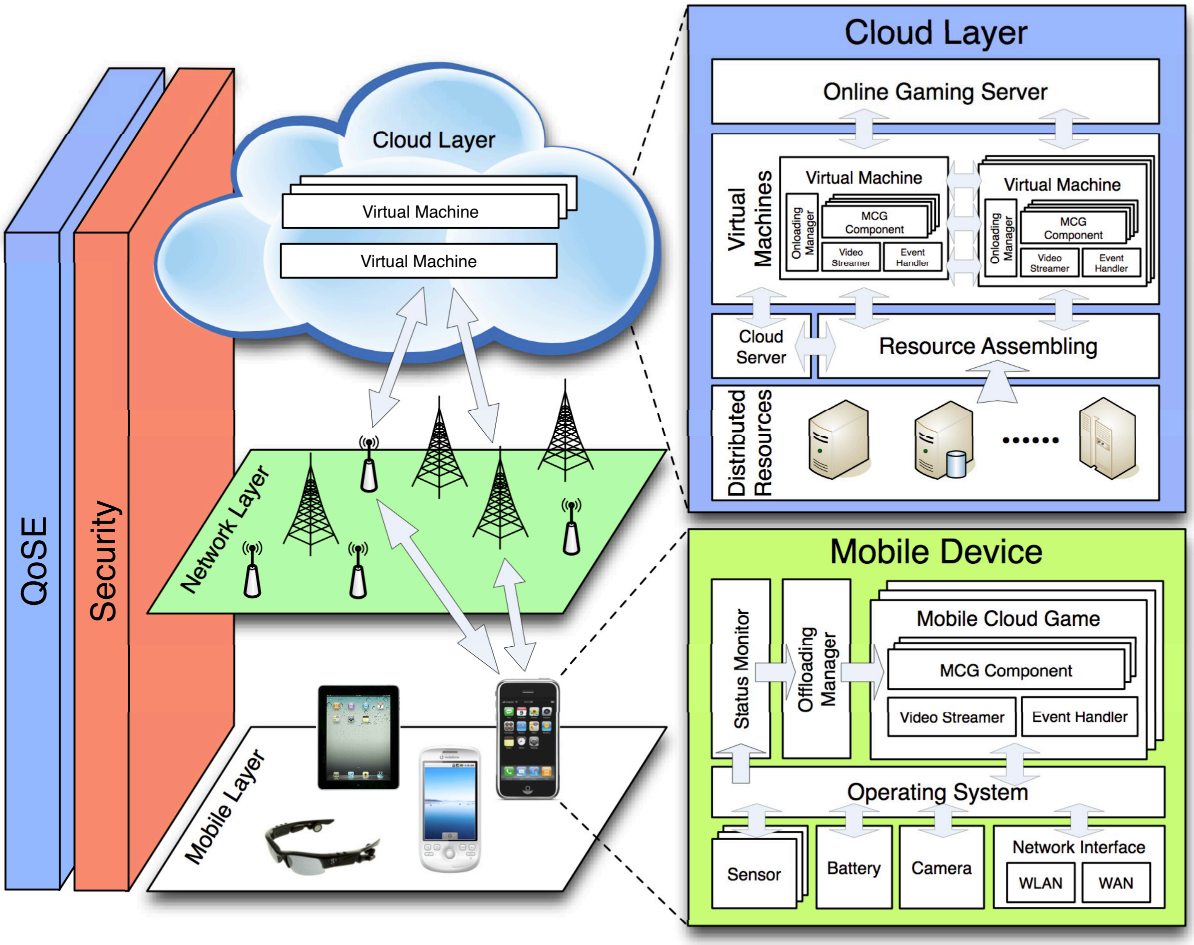


Figure 4. Framework for Next Generation Mobile Cloud Gaming System

addition, automatic transitions among different networks should be supported, especially when high-speed wireless local area networks are available free of charge for the mobile clients. To further improve the user experience, multi-path transmission is an alternative approach to enhance the network capacity.

C. Mobile Layer

The mobile layer is concerned with the design of mobile devices and the games running on it. In order to perform onloading and offloading for a MCG, partitioning of the gaming task is pre-requested. Afterwards, to support intelligent and dynamical offloading, the status monitor is responsible to measure the status of its residing mobile device and report its evaluation to the offloading manager. With the monitored information from the mobile devices, the onloading manager in the cloud decides the onload components, while the offloading manager in the mobile devices dynamically negotiates with the cloud about the offloading percentage. In addition, to support the unique features of mobile gaming, data from sensors in the mobile devices should be updated to the game engine in real-time, so that such information can have the desired effects in the gaming session.

D. QoSE

Throughout the whole MCG system, QoS should be guaranteed, especially for the large amount of video transmissions in the MCVG mode. In case the mobile client offloads a large percentage of the gaming engine to the cloud, the game player's QoE may become significantly different from that in conventional online gaming. In conventional online gaming, the user experience is a function of the network QoS and the presentation algorithm at the client. However, in MCG due to the possibility of onloading and offloading, the relations between the QoS and QoE are much more complex. Especially, MCVG mode enables remote playing of games that are never designed to be played over a network, which may also change the users' perception [8]. In order to distinguish our evaluation for next generation MCGs from traditional online games, we name these performance measures as QoSE.

E. Security

Security is another important concern regarding MCG systems. Along with the development of online gaming market, virtual assets in online gaming are attracting more attentions nowadays. To protect these virtual assets, the MCG design should address the security issues. Package forgery is one of the most common ways to attack an online gaming server. In addition, encryption of transmission packages is mandatory since the wireless network traffic is easy to be sniffed. Furthermore, since the users may be charged for cloud resources, authorization, authentication and accounting procedures are needed. Game players should be identified and authenticated after the cloud server receives his/her game request, and the usage of cloud resources should be properly accounted for.

V. Open Research Issues

There are plenty of open research issues for the proposed MCG system. In this section, we discuss some of the most important issues, in order to provide a guide for future research in this area.

A. Cloud-End Issues

A four-level architecture [9] is well recognized in cloud design, including Data Centers Layer, IaaS (Infrastructure as a Service) Layer, PaaS (Platform as a Service) Layer, and SaaS (Software as a Service) Layer. For the proposed MCG system, the cloud-end is providing a GaaS. The most critical open research issues in GaaS include:

1) VM Design: In cloud computing, how to virtualize distributed resources is one of the most critical issues for VM design. However, the resource requirement for each game session in an MCG system does not impose a heavy load, so that multiple VMs commonly reside in a single server. Therefore, scheduling of parallel tasks in a server is an important topic, since the gaming procedures are always sensitive to round-trip time (RTT). Moreover, each VM running in the cloud is a request handler for its corresponding mobile client. In order to cope with the operating environments of the mobile devices, VMs should synchronize themselves in time. However, frequent updates might cost a large amount of data transmissions over wireless networks, which may potentially decrease the overall system performance. Therefore, how to design a flexible and efficient VM remains an open issue.

2) Interactions of VMs: In multi-player online gaming, players are able to interact with each other. These interactions involve exchange of information between VMs in MCG system. Traditional online gaming systems always have a request handler server, which interacts with players' terminals in server-client mode. However, as Massive Multiplayer Online Games (MMOGs) are highly delay-sensitive, thus the bottleneck of a handler server can significantly degrade their playability. To deal with this issue, [10] proposes to use peer-to-peer techniques to distribute the MMOGs server functionality and place it at the cloud server centers in order to reduce the additional delay introduced by running the MMOGs clients in the cloud. However, the ad hoc manner also introduces overhead and redundant information dissemination. Therefore, further investigation on the trade-off should be conducted.

B. Game Design Issues

In MCG systems, mobile devices not only serve as an interface for the players but also gaming monitors, environment sensors and partial game executor. The design of MCG involves many issues, resolution of which may even require inter-disciplinary knowledge. In this section, we only discuss some featured open issues for the next generation MCG systems as follows.

1) Partitioning: To achieve onloading and offloading, MCG should be partitioned into components in advance. The first challenging issue is to overcome the problems introduced by data dependency in applications. There has been plenty of work on mobile cloud computing that discussed how to model applications as a consumption graph, and then proposed partitioning algorithms. However, since gaming is always a procedure-oriented application, more specific research and studies should be conducted. Dynamic partitioning [11] can also be considered in MCG. Furthermore, once the games are partitioned into components, they should also be adapted to various client systems and hardware specifications. How partitioning can be done efficiently using existing or novel enabling technologies remains open issues.

2) Client Design: To enable the partial game execution on the mobile devices, the MCG system requires an efficient and flexible client. Mobile browser is an ideal platform for MCG, since it provides a unique execution environment over various mobile operating systems. Even though there has been a lot of work on improving the capacity of next generation mobile browser, the issues introduced by the proposed MCG system should be taken into consideration. Improvement on browser architecture, script languages, and runtime plug-ins are needed. Moreover, the privilege of browsers to access system status and equipped sensors and camera, which is a fundamental need for MCG system, should be granted with privacy protection. An alternative for the MCG client is thin mobile applications. In contrary to traditional games, MCG client applications provide more flexible modes, which enable the transmission and execution of codes from the cloud. Therefore, the client design should consider the compatibility to these onloading game components.

3) Context-Awareness and Augmented-Reality: Context-aware services and augmented reality for mobile cloud computing are no longer a new topic. However, with cloud support, the MCG system introduces more opportunities and challenges. From the system point of view, context-aware technologies enable the mobile devices to get inputs from the surrounding environment, while augmented reality is an enhancement on the output to the users. The cloud is a rich resource that processes and provides information for users from all over the world. Research issues involve the procedure of incorporating the context information into the gaming content, interacting with the massive players in the cloud, and presenting the gaming contents in an augmented reality.

C. Connectivity Issues

To enable acceptable performance of MCG, the quality of network access should be guaranteed. Even though the wireless communication technologies are fast developing, there are still challenging issues for MCG systems. In this section, we discuss some of them as follows:

1) Network Transmission Protocol: There are various communication needs for the proposed MCG system. The transmission packages include but are not limited to: i) control commands, which are frequent and small packets from the users to the cloud; ii) game components, which are executable codes, onloading from cloud to mobile devices; iii) messages, which are communicating components to facilitate the gaming procedure; iv) interaction messages, which enable interactions between players from a VM to another; v) video frames, which are the gaming content from cloud to the mobile devices when the mobile client offload the video streamer to the cloud. Existing protocols, such as SOAP, REST, AJAX, RTS, are designed for general purpose network communications, and might not be able to fulfill the requirements of the proposed framework. Therefore, further research in redesigning the transmission protocol to cope with the MCG system is necessary.

2) Mobility Management: Next generation MCG systems are designed for gaming in mobile scenarios. However, the mobility of mobile devices introduces many critical issues, such as the support of seamless handovers. To provide a ubiquitous gaming experience, the player should not be interrupted when her wireless connection is handed off from one base station to another. One solution is delay-tolerated mechanisms in game design. Take DrawSomething as an example, which is one of the most popular games in iOS and Android platforms. The players connect to the network to access the drawings from peers and the gaming status will be frozen if the network is disconnected. However, not all online games are able to adopt this method, especially real-time battlefield games, e.g., Real Time Strategy (RTS), Online First Person Shooting (OFPS), Massive Multiplayer Online Role-Playing Game (MMORPG). Therefore, seamless handover is of particular interest for the research in MCG system.

3) Multi-Path Network Access: Current mobile devices are always multi-homed, which means they are equipped with difference network interfaces, such as WiFi, 3G, LTE, WiMax, etc. Meanwhile, many heterogeneous wireless networks have overlapping coverage areas, which implies the possibility of multi-path network access. As has been discussed before, the information exchange between cloud and the client is various, which enables the potential improvement of system performance by scheduling different types of packages for multi-path network access. However, even though there have been a lot of studies in this area, practical solutions for multi-path wireless access are still needed.

4) Geographical Optimization: Cloud gaming over wireless networks is extremely RTT-sensitive. In networking systems, the RTTs of mobile devices are related to the geographical distance and network hop-count to the corresponding VMs. As the resources in the cloud are also distributed, how to explore the possible path and assign optimal resources to mobile clients is an issue of great interest. Motivated by this, a VM-based cloudlets approach is proposed in [12]. With the concept of cloudlet, the mobile devices can offload their computations to the nearest resource-rich cloudlet instead of cloud. Therefore, the bottleneck of network connection to the cloud can be addressed, since the access to cloudlet can be of higher bandwidth, such as WiFi. For MCG, the cloudlet can be interpreted as a gaming terminal, where players can be connected to start cloud gaming. However, due to the mobility issues, it is hard to design a very successful model. Therefore, this research area is still open for further studies.

D. Dynamic Onloading and Offloading Issues

One of the most important features of the proposed MCG system is the dynamic onloading and offloading mechanism. The procedure needs to work with the measurement of QoSE and the evaluation of device status, in order to provide a satisfying user experience and achieve optimized system performance. Here we list the research issues in this area:

1) QoSE Measurement: Real-time measurement on user-perceived QoSE is the first step to the dynamic onloading and offloading approach. Factors affecting the user QoE for MCG have been analyzed in[2]. Based on the analysis, the authors developed a prototype for real-time measurement of mobile gaming user experience that can be used in real networks. The most significant affecting factors for MCG are network factors (bandwidth, delay, packet Loss, and jitter) and video setting factors (resolution, data rate, frame rate, codec). However, this work only studied the MCVG case. In the proposed MCG system, similar approaches can be considered to measure the user QoSE in real-time.

2) Device Status Evaluation: The Status Monitor on the mobile device is in charge of monitoring its own status, such as CPU usage, battery lifetime, network bandwidth, etc. The statistics of these monitored parameters are taken into account for dynamic onloading and offloading. Therefore, the procedure of monitoring is required to be accurate and real-time. In addition, the system demands an evaluation function to summarize the statistics of these statuses to support the decision making in the offloading manager.

3) Objective Function for Onloading and Offloading: With the measurement of QoSE and real-time statistic parameters of status monitoring, the cloud and the mobile client are able to perform dynamic onloading and offloading. In general, onloadig is a procedure that the cloud migrate some components to the mobile client, in order to enhance the gaming performance and to reduce the network burden during a gaming session. On the other hand, offloading is an opposite procedure with which the mobile client delegates a certain proportion of the computing tasks to the cloud, for the purpose of decreasing the computing load and possibly power consumption on the mobile devices. Apparently, there is a trade-off between onloading and offloading. Notice that onloading is a procedure with running code transmission, while the offloading involves sequences of remote invoking between gaming components. Their overheads in the proposed MCG system should not be neglected. Therefore, to design an objective function, which intelligently considers all factors in dynamic onloading and offloading procedure in real-time, is a critical research issue.

VI. Case Studies

In order to illustrate next generation MCGs, we introduce two examples in this section.

A. Augmented-Reality Cloud Game

Project Glass is a research and development program by Google to develop an augmented reality head-mounted display [7]. In contrary to traditional mobile devices, Google Glasses provide a hands-free display of information on the lenses, integrating the virtual display to the reality in one's vision. In addition, the device enables people to interact with the Internet via natural language voice command. Therefore, it is a perfect solution for augmented-reality cloud games, which can be launched while people are walking.

Fig. 5 illustrates an augmented-reality cloud game. The camera on the glasses continuously captures the player's vision in real-time, and the device transmits the video to the cloud via a wireless network. In the cloud-end, the video analyzer processes the video images with sophistical artificial intelligence technologies, such as pattern recognition. The game logic in cloud then creates gaming contents and delivers them to the game players. These virtual gaming contents, such as coins and bombs, can be displayed in the real scenarios through the lenses. Therefore, the system provides the players a gaming world with mixed virtual and real items. During the gaming session, the players should move their bodies or their vision angels to interact with those virtual items, in order to achieve the designed gaming goals.

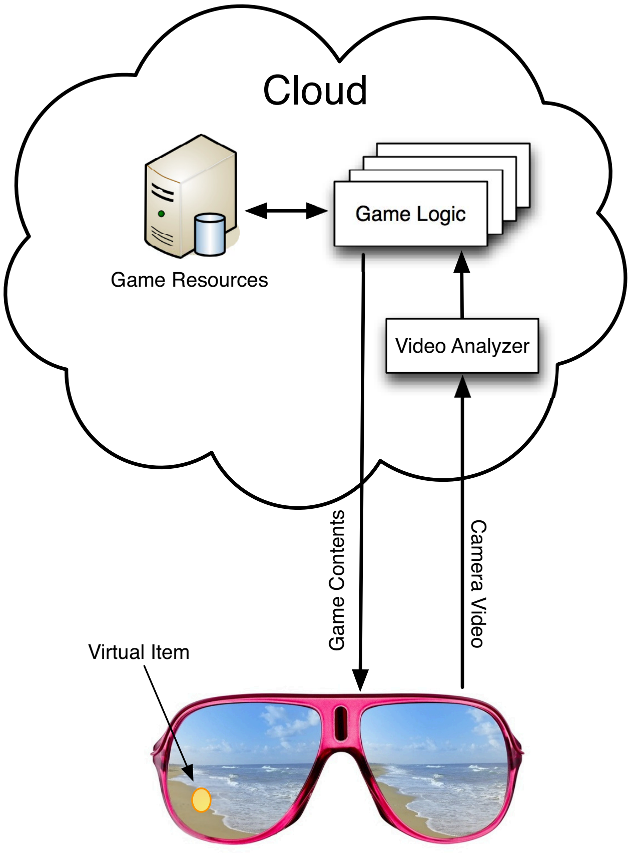


Figure 5. Augment-Reality Cloud Game

This type of games can be used in daily exercise and for health-care purposes. However, how to guarantee the safety of players during the gaming session remains a critical issue for game designers.

B. Context-Aware Cloud Game

An example of context-aware cloud gaming is gaming onboard a vehicle. People nowadays prefer to entertain themselves with games when they are trying to kill their time onboard a bus or subway train. The mobility of a vehicle provides a new gaming experience for players.

In this gaming scenario, the vehicular game reports GPS information to the cloud via a wireless network, so that the cloud is able to deliver corresponding gaming contents to the mobile devices. For example, when the player is in the urban area, the environment of the game is set to be in the crowd and busy; when the player is in the suburb area, virtual wild animals might appear in the game and attack the avatar. Furthermore, the game also collects the mobility information with its equipped accelerators and the cloud utilizes these sensed data to facilitate various gaming contents. For example, when the vehicle is accelerating, the avatar in the game will enter a speed-up mode, such that the player has less response time to deal with the challenges in the game. In addition, with the assistance of the cloud, the game is able to search for peering players, e.g., those in the same vehicle, thus introducing more interactive gaming scenarios, such as encountered challenges.

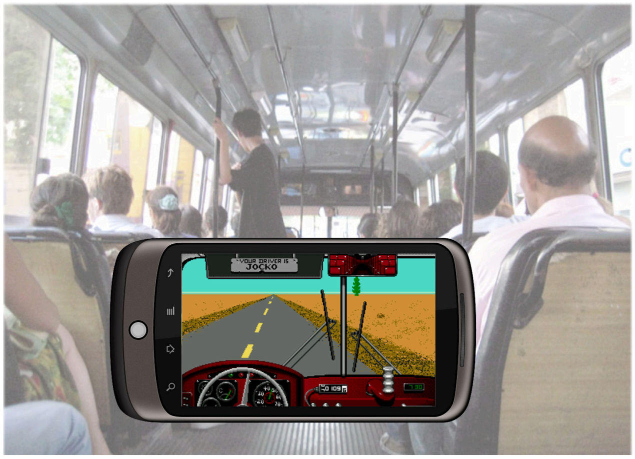


Figure 6. Context-Aware Cloud Game

As shown in Fig. 6, the player is controlling a virtual bus when he/she is on a real bus. Compared to augmented-reality cloud games, the player can focus on the gaming content while the vehicles are moving them towards their destinations.

VII. Conclusion

With the emergence of cloud computing, the future Internet infrastructure is evolving to enable everything-as-a-service. As a cloud-based mobile gaming system, MCG is a promising paradigm for gaming delivery. In this work, we have studied two types of cloud-based mobile games, i.e., mobile cloud video games and mobile browser games. We have explicitly defined the term of mobile cloud computing. We have discussed the features of MCG and proposed a framework for next generation MCG systems, which should support the dynamic cloud integration, pervasive gaming, context-aware gaming, multi-player online gaming and seamless cross-platform gaming. Open research issues for the proposed MCG framework have been reviewed and discussed.

Acknowledgement

This work is supported by a University of British Columbia Four Year Doctoral Fellowship and by funding from the Natural Sciences and Engineering Research Council.

References

[1] L. Guan, X. Ke, M. Song, and J. Song, “A survey of research on mobile cloud computing,” in *Proceedings of the 2011 10th IEEE/ACIS International Conference on Computer and Information Science*, Washington, DC, USA, 2011, ICIS ’11, pp. 387–392.

[2] S. Wang and S. Dey, “Modeling and characterizing user experience in a cloud server based mobile gaming approach,” in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 30 2009-dec. 4 2009, pp. 1 –7.

[3] J.-M. Vanhatupa, “Browser games: The new frontier of social gaming,” in *Recent Trends in Wireless and Mobile Networks*. 2010, vol. 84 of *Communications in Computer and Information Science*, pp. 349–355, Springer Berlin Heidelberg, 10.1007/978-3-642-14171-3-30.

[4] S. Wang and S. Dey, “Rendering adaptation to address communication and computation constraints in cloud mobile gaming,” in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, dec. 2010, pp. 1 –6.

[5] W. Cai and V. Leung, “Multiplayer cloud gaming system with cooperative video sharing,” in *Proceeding of The 2012 International Workshop on Emerging Issues in Clouds (in conjunction with 4th IEEE International Conference on Cloud Computing Technology and Science)*, Taipei, Taiwan, 2012, CloudCom 2012, IEEE.

[6] J.-M. Vanhatupa, “On the development of browser games - technologies of an emerging genre,” in *Next Generation Web Services Practices (NWeSP), 2011 7th International Conference on*, oct. 2011, pp. 363 –368.

[7] D. Goldman, “Google unveils ’project glass’ virtual-reality glasses,” in *Money*, apr. 2012.

[8] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hossfeld, “An evaluation of qoe in cloud gaming based on subjective tests,” in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, 30 2011-july 2 2011, pp. 330 –335.

[9] I. Foster, Yong Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in *Grid Computing Environments Workshop, 2008. GCE ’08*, nov. 2008, pp. 1 –10.

[10] R. Suselbeck, G. Schiele, and C. Becker, “Peer-to-peer support for low-latency massively multiplayer online games in the cloud,” in *Network and Systems Support for Games (NetGames), 2009 8th Annual Workshop on*, nov. 2009, pp. 1 –2.

[11] B. Chun and P. Maniatis, “Dynamically partitioning applications between weak devices and clouds,” in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing &#38; Services: Social Networks and Beyond*, New York, NY, USA, 2010, MCS ’10, pp. 7:1–7:5.

[12] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14 –23, oct.-dec. 2009.