Cloud Gaming: A Survey

Abstract Here . . .

CCS Concepts: •**Cloud Gaming** → **Gaming as a service;**•**Gaming on demand** → *Mobile Cloud Gaming;*•**Next generation mobile gaming** → Online gaming;•**Multimedia live streaming** → Live video streaming;

General Terms: Cloud Gaming, Mobile Cloud Gaming, Cloud Computing, Distributed Computing

Additional Key Words and Phrases: cloud, game, video, remote

ACM Reference Format:
Gang Zhou, Yafeng Wu, Ting Yan, Tian He, Chengdu Huang, John A. Stankovic, and Tarek F. Abdelzaher, 2010. A multifrequency MAC specially designed for wireless sensor network applications. *ACM Trans. Embedd. Comput. Syst.* 9, 4, Article 39 (March 2010), 11 pages.
DOI: 0000001.0000001

1. INTRODUCTION

Introduction Here . . .

2. CLOUD GAMING

Recently another area of study has emerged as a merger of cloud computing and video gaming, known as “Cloud Gaming” or “Gaming as a service (GaaS)”. In cloud gaming, games are executed on the cloud and their high definition scenes are rendered using intensive cloud machines. After running the games remotely, output of the games, in the form of video frames, is streamed back towards the client. Similarly, users interaction with gaming application is sent back to the game running on cloud over the Internet [Cai et al. 2014a], [Gharsallaoui et al. 2014], [Shea et al. 2013], [Cai et al. 2013], [Chuah et al. 2014], [Wang and Dey 2012], [Huang et al. 2013].

This technology has made it easier for the end users to play intensive games on a thin client without updating software or hardware of their machines.

2.1. Motivation and Challenges for cloud gaming

The mentioned new technology provides many advantages to the players as well as to the developer community of the computer games.

2.1.1. Advantages for the players. Cloud gaming provides the following advantages to the player community of computer games [Cai et al. 2013]:

— **Thin Client:** A very thin client is needed to play the game, as games are being rendered on the cloud, and video is streamed from server. As a result, the client

This work is supported by the National Science Foundation, under grant CNS-0435060, grant CCR-0325197 and grant EN-CS-0329609.
Author’s addresses: G. Zhou, Computer Science Department, College of William and Mary; Y. Wu and J. A. Stankovic, Computer Science Department, University of Virginia; T. Yan, Eaton Innovation Center; T. He, Computer Science Department, University of Minnesota; C. Huang, Google; T. F. Abdelzaher, (Current address) NASA Ames Research Center, Moffett Field, California 94035.

DOI: 0000001.0000001

is required to receive and show the video frames of the game, and send the user's interactions to the cloud.

— **Potential Battery Conservation:** Complex rendering is being done on the cloud which reduces the battery consumption of the client machine.
— **Seamless gaming:** Games are platform independent and the players can seamlessly play the same game resuming from the same level by sitting anywhere in the world.

2.1.2. Advantages for the developers. Apart from being beneficial for the player community, cloud gaming provides following advantages to the developer community of computer games:

— **Secure Gaming:** Data related to the game (e. g. level information) is residing in the cloud and only developer has complete access to all the data. This stops the users from getting the pirated version of the game.
— **Easy Updates:** Game logic and game information are mostly residing on the cloud so the developer can update the cloud part of the game and that would be automatically reflected to the client.
— **Unlimited Cloud Resources:** Developers have no need to worry about the storage or computing resources for the game as the cloud is providing scalable resources[Cai et al. 2013].

2.2. Cloud Gaming Challenges

Despite the advantages provided by cloud gaming, the area still has many challenges which are needed to be managed by the researchers. The challenges faced by the cloud gaming application depend on the architecture of that application. For example, the biggest hurdle in the efficiency of a 'Remote Rendering Cloud gaming application' is **the limited bandwidth**. Since the scenes of the game have to be transmitted over the internet in the form of video frames. Therefore, the bandwidth needs to be fast enough to quickly transmit these frames to the client and interactions of client back to the server. Similarly, the complication in 'Local Rendering application' is the **formation of such instructions** that could be used to completely render the scene of game at client side.[Cai et al. 2014a] Other than these obstacles, rendering the game remotely with efficient encoding is also considered as a problem in this technology. Moreover, recent happening with Onlive and GaiKai has guided the researchers to think about a better business model that can be used by cloud gaming service providers.

2.3. Cloud Gaming History

It is inferred that the idea of Cloud Gaming was emerged from the concept of audio and video streaming [Mungee et al. 1999]. In 2006, De Winter et al. [De Winter et al. 2006] proposed a hybrid thin client protocol for multimedia streaming where they addressed interactive games too. In their work, a realtime desktop streamer was formulated using a video-codec to stream the graphical output of applications after GPU- processing to a thin-client device. Then, in 2007, Eisert and Fechteler [Eisert and Fechteler 2007] introduced the concept of remote rendering of computer games. This work was a part of European project Games@Large [Tzruya et al. 2006] and can be declared as a stepping stone in cloud gaming. Apart from elaborating the concept of rendering the game scenes at server and streaming the video scenes towards the client, the work also explained the idea of local rendering for cloud gaming application where graphic commands were streamed to the client and game scenes were rendered locally. Also, a closer step to this fascinating technology was taken by Eisert and Fechteler[Eisert and Fechteler 2008] in 2008 when they introduced the idea of computer graphics streaming over Local Area Network (LAN). The work was specifically focusing on graphics in video games and how these games can be streamed over the local area network.

In 2008, emergence of OnLive[OnLive] as the first commercial platform for providing 'Gaming as A service' and its high appreciation motivated the business and academic community to take 'Cloud Gaming' more seriously. After this progress, most of the work was done to overcome the challenges (Bandwidth usually) in CG. After OnLive, many other cloud gaming platforms also emerged including Gaikai[Gaikai], NVIDIA GRID[Corporation], and GamingAnyWhere[Huang et al. 2013] which was the first open source cloud gaming platform that has been started to use for research purpose. Most of the cloud gaming platforms are using remote rendering model of CG and game video is being streamed towards the client. But, an important break-through was done by Wei Cai et al. in [Cai et al. 2013] when they proposed and implemented a 'Cognitive Model' of CG in which game components were being transferred from the cloud to the client and vice versa to get a better gaming experience.

2.4. Types of Cloud Gaming

Recent literature describes two kinds of cloud gaming [Hong et al. 2015]:

(1) **Video Streaming Cloud Gaming:** Most appreciated form of cloud gaming in which video is streamed towards the client. It is usually referred as 'Gaming on Demand' and in the following sections, the term Cloud Gaming would be used instead of Video Streaming Cloud Gaming.
(2) **File Streaming Cloud Gaming:** In this form of cloud gaming, a portion of game application is downloaded at the client side to start the game. Once the game is started on client side, the remaining file is downloaded on the fly. Kalydo is one of the companies which use this kind of cloud gaming service [Kalydo].

2.5. Cloud Gaming Application Components

Since, in cloud gaming, output of games is streamed to client over the internet and users' interactions are sent back towards the cloud. Therefore, to do all these functions, a cloud gaming platform mostly have specific components and each of them does its part and forwards its output to other. A general online game would be having four main components[Cai et al. 2014a]:

— Input module
— Game logic
— Networking module
— Rendering module

Based on these components, different architectures are possible for a cloud game. Firstly, these components and their functionality will be explained and then the possible architectures of a cloud gaming application will be analyzed.

2.5.1. Video Renderer. Once the input commands are received by the game logic, game world changes according to inputs provided by the player. Then, video rendering unit renders the updated game frames using graphical processing unit (GPU). Another method used for generating frames of game scenes is to let the game render its video frames and take a snapshot of the game desktop by desktop capture module at a specified frequency. This method is used in one of the two implementations of video source by GamingAnywhere platform. Also, depending on the operating system of the end user, audio of the game is generated by the audio source of the system. For example, GamingAnywhere uses ALSA library for Linux, and Windows session audio API for Windows to capture the audio[Gharsallaoui et al. 2014].

2.5.2. Input Handler. User interactions are received by this module. After receiving the inputs, the module converts these mouse clicks and key strokes to appropriate commands and send those commands to the game logic. This component must be at the client side of a cloud gaming system[Shea et al. 2013].

2.6. Possible Architectures of Cloud Gaming Applications

Recent literature provides two different parameters for elaborating CG application architectures as described below:

(1) Models with respect to components' division between client and server
(2) Models with respect to game integration with Cloud Gaming platform

2.6.1. Architecture w.r.t Components' Division. The first approach is discussed in [Cai et al. 2014a], and according to this work, a cloud gaming application could be distributed amongst the client and the server of the game. These different distributions of components are referred as architectures of a cloud game. Based on the mentioned components, possible architectures are [Cai et al. 2014a]:

Cloud gaming model	Components of game at Client	Components of game at Cloud
Remote Rendering	Input Controller	Game Logic, Networking, Database, Video Renderer
Local Rendering	Input Controller, Video Renderer	Game Logic, Networking, Database
Cognitive	Dynamically decided	Dynamically decided

Table I: Different Models of Cloud Gaming

(1) **Remote Rendering:** This model is the ultimate inverse of a local (stand-alone) gaming application. The function of the client in this architecture is just to receive inputs from users, convert those inputs to commands and send those commands to the game residing in the cloud. On the other hand, all the remaining components of the game are residing in the cloud. The architecture works in such a way that whenever a user interacts with the gaming application, input commands are generated and sent over the internet to the game logic using input controller. Then, game scenes are updated and actions are taken using game logic and video rendering unit render scenes as frames. After that, these frames are sent to the client over the internet[Barboza et al. 2010]. The model provides smart solution for the users not having sufficiently intensive hardware to render the graphics of high definition games. But, this can be clearly observed that the bottleneck would be "Bandwidth" in this scenario, as the Internet is playing a key role here.
(2) **Local Rendering:** The model does components' divisions in such a way that video renderer also resides at client side with input controller, and remaining parts of the game are kept in the cloud. The model resolves the bandwidth issues by rendering the scenes at the client side. This architecture resembles with remote rendering model till the input commands update game scenes, but rather rendering the scenes and sending the packets, output instructions are generated and sent towards the client. Then, the client receives these instructions, renders the scenes using these instructions and displays to the end user[Cai et al. 2014a]. As mentioned, in comparison to remote rendering, this model is remarkably efficient in terms of internet bandwidth. Mobile browser gaming has adopted this architecture in most of the applications.

(3) **Cognitive:** This model has dynamic separations of client and cloud parts of the game. In this approach, client and cloud divisions of the game are dynamically decided on behalf of the user resources (e. g. computation and rendering power, bandwidth). Cai et al. (2013) [Cai et al. 2013] proposed this method in their work where cloud gaming application was divided into various parts. Also, the application was able to transfer game components from cloud to the client and vice versa. But, to achieve this collaboration (on-loading and off-loading components), the application must be partitioned into components in such a way that can resolve dependency issues. Also, the work did not elaborate the control mechanism of this components' movement, there should be a component within the application that can make decisions on this on-loading and off-loading actions.

2.6.2. Architecture w.r.t Game Integration with Platform. The work described in [Cai et al. 2016] distributes cloud gaming applications in three different architectures based on how games are integrated with the cloud gaming platforms. Details of these models are as follows:

(1) **Black-box:** In these type of cloud gaming applications, game scenes are rendered in cloud and are streamed back towards the client.
(2) **In-Game Context:** These applications are programmed in such a way that game streaming is made adaptive with using some optimization techniques.
(3) **New Programming Paradigm:** In this architecture, gaming application is distributed between the game client and server.

2.7. Cloud Gaming Platforms

Cloud gaming has been able to grasp the attention of entrepreneurs, that is why various companies are providing "Gaming as a Service". Some of the notable cloud gaming platforms are discussed here:

2.7.1. GamingAnyWhere. GamingAhyWhere is an open-source cloud gaming platform that provides reconfiguration for researchers and developers. Huang et al.(2014) suggested that each component of this video streaming system can be replaced by another component having different protocol or algorithm. Also, their work elaborated the flow of clients communication with game servers and portal servers. Portal server provides an interface to the user for login to a system and selecting the desired game. When the user selects a particular game, the portal server finds and launch that game on a server, and sends the URL of the game server to user. The client connects to the game server and starts playing game [Huang et al. 2014].

2.7.2. OnLive. OnLive is a real time video streaming platform for cloud gaming and its clients are available for various operating systems. The platform lies in the category of video streaming platforms of online gaming as scenes are rendered virtually and streamed back to the thin client[Gharsallaoui et al. 2014]. Due to its robust structure and real-time compression techniques, it had been able to get noticed by the business community and Sony has acquired this platform in 2015[OnLive]. Manzano et al.(2014) identified three phases of an OnLive session and this explains the overall flow of client's interactions with the cloud gaming application[Manzano et al. 2014]:

(1) User authentication and allocating the user a particular site for load balancing
(2) OnLive main menu
(3) Game playing

To elaborate their work[Manzano et al. 2014], in the first phase of communication, a user connects with the OnLive authentication server and it is authenticated possibly using HTTP-based messages. Once the user is authenticated, it probably (communication between client and authentication server is encrypted) gets the IP of onLive servers present at more than one locations. Then, client measures its round trip time (RTT) with those servers to find the most suitable one for it to make a connection. After finding the closest server, the client identifies available downstream bandwidth using another measurement session and with the help of RTT and calculated downstream, client adjusts its streaming bit-rate that can also be changed during streaming. In phase two, the client communicates with OnLive main menu server and main menu which is an interactive video, is streamed back towards the client using the same protocol that would later be used for game streaming. Using this menu, player selects a game for playing and this phase ends when the desired game is selected. Finally, communication between client and OnLive gaming server gets started and the game is streamed towards the thin client.

2.7.3. StreamMyGame. This platform provides some extra features apart from cloud gaming services to the users. Using this system, users can record their game-play videos and upload that recorded video to Facebook, Youtube, and other sites. Also, clients can broadcast their videos to the local network and can communicate with other members using community services. [Gharsallaoui et al. 2014]

2.7.4. Amazon AppStream. AppStream service provided by Amazon Web Services (AWS) is typically designed to stream the output of a particular application. After executing the application on a virtual machine of the service, resultant output of the execution is streamed towards the client. Though this service is not especially formulated for game streaming but recently its focus has been shifted to the cloud gaming and by introducing its SDK in Java language, it does provide the capability of **Hybrid Streaming** to the developers.[app]

2.8. Current Game Genre

Depending on some specific parameters, current games are usually categorized in different game classes, and these classes are usually referred as 'Game Genres'. Due to diversity in the games, analyzing them with respect to different parameters (e. g. Game-play, camera-view, purpose et cetera) yields different genres. For example, Lee et. al (2014) have used different parameters to classify the games in various genres. Table II is showing some of their classification work [Lee et al. 2014b]:

Parameter Used	Game Genre Classification
Game-play	Action, Action/Adventure, Racing, Fighting, Puzzle, Role Playing, Shooting, Simulation, Sports, Strategy
Point-of-view	First Person, Third Person, Overhead, Multiple perspectives
Purpose	Education, Entertainment, Exercise, Meditation, Party, Social
Presentation	2D, 3D, Isometric, Static Background, Vertical Scrolling, Side Scrolling, Grid Based, Perspective manipulation

Table II: Different Game Genres

It can be seen in Table II that the resultant game genres will be depending on the chosen parameters for classification.

2.9. Effect of Game Genre on Cloud Gaming Experience

The recent research has found that 'Cloud gaming' is not equally appropriate for each of the current game types. Generally, genres that have drastically changing consecutive frames are considered to be less appropriate to deploy as cloud gaming applications. To look deeper into the problem, study is done to analyze the different game types in terms of their scenes complexity, as it does play a major role in deciding game's friendliness for the cloud. For example, to analyze the motion and scene complexity of games, Claypool (2009) [Claypool 2009] studies the different game classes by using game perspective (Camera view) as the parameter of classification. After identifying four different game types (First Person Linear (Battlefield), Third Person Linear, Third Person Isometric, and Omnipresent), motion and scene complexity for each of the game class is computed. The results of their work have found that 'First Person games' have higher motion (PFIM = Percentage of Forward/backward or intra-coded Macroblocks) than Omnipresent and Third Person Linear games, and Third Person Isometric games are having the lowest PFIM. Also, Omnipresent games are observed to have the highest scene complexity, and Third person Linear and First Person games are considered to have the least complex scenes.

Likewise, the difference in Gameplay and overall structure of the game also influences the rate of delay tolerance and delay sensitiveness. Domineco (2014) [Domenico 2014] has described the effect of game genre on sensitiveness. In the proposed work, the author discusses four game types including Action games, Racing games, Real-Time Strategy (RTS) games, and Puzzle games. Using OnLive as a cloud gaming platform, experiments are conducted on games from each of the mentioned genres, and it is seen that Puzzle and RTS games are more delay-tolerant as compared to Action and Racing games. The reason for this change is that, in later games, complex synchronization over the network is required to determine the collision between game objects after player's interactions. On the other hand, RTS games feature lots of interactions for which the resultant can be computed locally without synchronizing with the server.

2.10. Cloud Friendly Game Genre

Some of the recent studies are specifically done to classify the current game genres with respect to their cloud friendliness. For instance, Lee et al. (2012) [Lee et al. 2012] propose a model to predict the suitability of a particular game to be deployed on the cloud. The suggested model investigates the effect of response latency on users' experience for three game types (First Person Shooting (FPS), Role Playing (RPG), Action). To illustrate, the work analyzes spatial and dynamic changes on the screen for these games, and how frequently the game demands input from the user. Using this model, Action games are found less susceptible to latency and have lower real-time strictness than FPS and RPG games. The reason for this is, in Action games, the users mostly have to provide many inputs for certain attacks, and these attacks last for more than 3 seconds and cause a damage.

Another study is conducted by Suznjevic et al. (2014) [Suznjevic et al. 2014] in which change in subsequent frames is observed. The study suggests that Shooting and Action games are more dynamic and their temporal metrics have the highest values. Therefore, these games can be referred as 'un-appropriate' for cloud.

3. SOLUTIONS DISCUSSED IN PREVIOUS WORK

The basic obstacle in achieving better quality of experience in cloud gaming is its heavy dependence on network bandwidth [Lampe et al. 2013]. Most of the commercial cloud gaming services including OnLive, StreamMyGame, and GaiKai are using 'Remote Rendering Model' of Cloud Gaming which makes it difficult to use in low bandwidth areas. Therefore, studies have been conducted to reduce quality of experience issues at user end. Solutions can be divided in following categories:

— Using adaptive streaming technique [Hong et al. 2015],[Ahiska and Ahiska 2012],[Wu et al. 2015]
— Applying cognitive model for Cloud Gaming [Cai et al. 2015],[Cai and Leung 2014]
— Enhancing video encoding at server side [Shi et al. 2011]
  — Video encoding optimization using graphic rendering contexts [Shi et al. 2011]
  — Efficient encoding using game engine information [Semsarzadeh et al. 2014]
— Frame Prediction [Lee et al. 2015],[Lee et al. 2014a]
— Peer assisted Cloud Gaming [Cai et al. 2014],[Cai and Leung 2012], [Süselbeck et al. 2009]

(We can discuss each of these solutions in detail too)

4. COST EFFICIENT SOLUTIONS FOR SERVICE PROVIDERS

In the past, most of the studies were focused on enhancing the quality of experience of end user (Game player). But, one of the most important entity in the Cloud Gaming technology is the service provider. The one who is providing cloud gaming services to the end user. Recently, one of the widely praised service provider of Cloud Gaming, OnLive, has shut down its services and sold its patents to Sony [Cai et al. 2016]. These kind of events encouraged researchers to work on cost-effective cloud gaming for the service provider. Rendering the gaming application at cloud servers put a lot of load at server side and sending those scenes towards the client also require investment on the network bandwidth. Therefore, studies are done to find solutions for Cloud Gaming Service Providers (CGSPs). For instance, Hong et al. [Hong et al. 2013] investigated the optimum use of virtual machines at cloud gaming servers to give maximum profit to the service provider. Similarly, in the work of Tian et al. [Tian et al. 2015], to minimize the overall cost of CGSPs, adaptive use of resources is proposed from three different angles: (a) Adaptive Data Center Selection; (b) Adaptive virtual machine allocation; and (c) Adaptive Video bit-rate configuration. To consider all the factors involving in an optimum solution, a constrained stochastic optimization problem was formulated and Lyapunov optimization theory was applied to derive a cost effective online strategy. Likewise, 'Cloud Resource Minimization' specifically in the case of Cognitive model of Cloud gaming is done in the work proposed by Cai et al.[Cai et al. 2014b]. In the proposed system, cloud resource monitoring in real-time is done and components of the game are dynamically transferred to selected terminals to reduce its own resource consumption.

5. USER QUALITY OF EXPERIENCE (QOE) STUDIES

Measuring user quality of experience in the case of cloud gaming is also an open area of study. This is due to the fact that game streaming is different from traditional video streaming because of its dependence on game genre and not being able to buffer at the client side. Several studies are done to investigate the user experience in playing a cloud gaming application. For example, in the study of Wang and Dey [Wang and Dey 2009], a model is formulated to quantitatively measure the user experience of mobile cloud gaming. According to their study, Mobile Gaming User Experience (MGUE) would mainly depend on the subjective factors: response time and video quality. They further explained that the subjective factors are affected by objective factors which can be classified into two groups: Video Settings (Image Resolution, Data Rate, Frame Rate, Codec) and Network Factors (Bandwidth, Delay, Packet Loss, Jitters). Similarly, In the work done by Jarschel et al. [Jarschel et al. 2011] user-perceived quality of experience is studied by conducting subjective user study. To do this study, a testbed is designed to provide a test person with a game experience similar to that of a cloud service. Using the formulated testbed, players were asked to play the gaming application in different scenarios having different values of packet delay and packet loss.The results show that in cloud gaming, it is really important for the players in which direction packet loss occurs. Likewise, another study was done by Claypool and Finkel [Claypool and Finkel 2014] to investigate user quality of experience and players' performance with added latency. To get a more general view about the technology, two user studies were undertaken using two different Cloud Gaming Platforms: (a) OnLive; (b) GamingAnywhere. Moreover, to control the latency between console and CG platform, Dummynet tool was used. Results on two third person avatar games (Crazy Taxi and Neverball) shown that, compared to traditional networked games, cloud based games are sensitive to even modest amount of latency and user performance degrades by up to 25% with each 100 miliseconds of latency.

6. CLOUD GAMING BUSINESS MODELS

Rapid growth in the Cloud Gaming services attracted the business community to come up with new business models. In this context, a case study [Ojala and Tyrvainen 2011] was done by Ojala et al. which elaborates business models of one of the popular platforms of cloud gaming, that is G-Cluster. The service was getting source code of games from the developers/publishers and updating the game according to requirements of streaming and games were streamed to the clients with the help of video on demand service providers and other potential stackholders. The revenue was generated from the end user who were charged with respect to the game publisher. Similarly, in the work done by Moreno et al.[Moreno et al. 2012], A platform 'Kusanagi' with its business model is proposed which is acting as middle-ware between developers/publishers of the cloud gaming applications and the end users. According to their business model, the flow of revenue will go from end-user to this middleware platform and from here to developer/publishers. Providing free trials to the users is recommended in the work so that the users become familiar with the service. After the trial, each game was decided to have its own price after consulting with the developer.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

Acknowledgements Here . . .

REFERENCES

(????).

Yavuz Ahiska and Bartu Ahiska. 2012. Adaptive Application Streaming In Cloud Gaming. (April 27 2012). US Patent App. 13/458,437.

Diego Cordeiro Barboza, Harlley Lima, Esteban Walter Gonzalez Clua, and Vinod EF Rebello. 2010. A simple architecture for digital games on demand using low performance resources under a cloud computing paradigm. In *Games and Digital Entertainment (SBGAMES), 2010 Brazilian Symposium on*. IEEE, 33–39.

Wei Cai, Henry CB Chan, Xiaofei Wang, and Victor Leung. 2015. Cognitive Resource Optimization for the Decomposed Cloud Gaming Platform. *Circuits and Systems for Video Technology, IEEE Transactions on* 25, 12 (2015), 2038–2051.

Wei Cai, Min Chen, and Victor Leung. 2014a. Toward gaming as a service. *Internet Computing, IEEE* 18, 3 (2014), 12–18.

Wei Cai, Min Chen, Conghui Zhou, Victor Leung, and Henry Chan. 2014b. Resource management for cognitive cloud gaming. In *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 3456–3461.

Wei Cai and Victor Leung. 2012. Multiplayer cloud gaming system with cooperative video sharing. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. IEEE, 640–645.

Wei Cai and Victor Leung. 2014. Decomposed cloud games: Design principles and challenges. In *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*. IEEE, 1–4.

Wei Cai, Victor Leung, and Min Chen. 2013. Next generation mobile cloud gaming. In *Service Oriented System Engineering (SOSE), 2013 IEEE 7th International Symposium on*. IEEE, 551–560.

Wei Cai, Victor CM Leung, and Long Hu. 2014. A cloudlet-assisted multiplayer cloud gaming system. *Mobile Networks and Applications* 19, 2 (2014), 144–152.

Wei Cai, Ryan Shea, Chun-Ying Huang, Kuan-Ta Chen, Jiangchuan Liu, Victor CM Leung, and Cheng-Hsin Hsu. 2016. The Future of Cloud Gaming [Point of View]. *Proc. IEEE* 104, 4 (2016), 687–691.

Wei Cai, Conghui Zhou, Victor Leung, and Min Chen. 2013. A cognitive platform for mobile cloud gaming. In *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, Vol. 1. IEEE, 72–79.

Seong-Ping Chuah, Chau Yuen, and Ngai-Man Cheung. 2014. Cloud gaming: a green solution to massive multiplayer online games. *Wireless Communications, IEEE* 21, 4 (2014), 78–87.

Mark Claypool. 2009. Motion and scene complexity for streaming video games. In *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 34–41.

Mark Claypool and David Finkel. 2014. The effects of latency on player performance in cloud-based games. In *Network and Systems Support for Games (NetGames), 2014 13th Annual Workshop on*. IEEE, 1–6.

NVDIA Corporation. The Power of Cloud Gaming. (????). http://www.nvidia.com/object/cloud-gaming.html

D. De Winter, P. Simoens, L. Deboosere, F. De Turck, J. Moreau, B. Dhoedt, and P. Demeester. 2006. A Hybrid Thin-client Protocol for Multimedia Streaming and Interactive Gaming Applications. In *Proceedings of the 2006 International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '06)*. ACM, New York, NY, USA, Article 15, 6 pages. DOI:http://dx.doi.org/10.1145/1378191.1378210

A Di Domenico. 2014. An evaluation of the impact of multimodal feedback on interaction performance and experience in mobile gaming. (2014).

Peter Eisert and Philipp Fechteler. 2007. Remote Rendering of Computer Games.. In *SIGMAP*. 438–443.

P. Eisert and P. Fechteler. 2008. Low delay streaming of computer graphics. In *2008 15th IEEE International Conference on Image Processing*. 2704–2707. DOI:http://dx.doi.org/10.1109/ICIP.2008.4712352

Gaikai. Gaikai. (????). https://www.gaikai.com/

Rahma Gharsallaoui, Mohamed Hamdi, and Tai-Hoon Kim. 2014. A Comparative Study on Cloud Gaming Platforms. In *Control and Automation (CA), 2014 7th Conference on*. IEEE, 28–32.

Hua-Jun Hong, De-Yu Chen, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2013. QoE-aware virtual machine placement for cloud games. In *Network and Systems Support for Games (NetGames), 2013 12th Annual Workshop on*. IEEE, 1–2.

Hua-Jun Hong, Chih-Fan Hsu, Tsung-Han Tsai, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2015. Enabling adaptive cloud gaming in an open-source cloud gaming platform. *Circuits and Systems for Video Technology, IEEE Transactions on* 25, 12 (2015), 2078–2091.

Chun-Ying Huang, De-Yu Chen, Cheng-Hsin Hsu, and Kuan-Ta Chen. 2013. GamingAnywhere: an open-source cloud gaming testbed. In *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 827–830.

Chun-Ying Huang, Kuan-Ta Chen, De-Yu Chen, Hwai-Jung Hsu, and Cheng-Hsin Hsu. 2014. GamingAnywhere: The first open source cloud gaming system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 10, 1s (2014), 10.

Michael Jarschel, Daniel Schlosser, Sven Scheuring, and Tobias Hoßfeld. 2011. 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*. IEEE, 330–335.

Kalydo. Kalydo. (????). http://http://www.kalydo.com/

Ulrich Lampe, Ronny Hans, and Ralf Steinmetz. 2013. Will mobile cloud gaming work? findings on latency, energy, and cost. In *Proc. of IEEE International Conference on Mobile Services (MS13)*. 960–961.

Jin Ha Lee, Natascha Karlova, Rachel Ivy Clarke, Katherine Thornton, and Andrew Perti. 2014b. Facet analysis of video game genres. *iConference 2014 Proceedings* (2014).

Kyungmin Lee, David Chu, Eduardo Cuervo, Johannes Kopf, Alec Wolman, Yury Degtyarev, Sergey Grizan, and Jason Flinn. 2015. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. *GetMobile: Mobile Computing and Communications* 19, 3 (2015), 14–17.

Kyungmin Lee, David Chu, Eduardo Cuervo, Alec Wolman, and Jason Flinn. 2014a. Demo: Delorean: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 347–347.

Yeng-Ting Lee, Kuan-Ta Chen, Han-I Su, and Chin-Laung Lei. 2012. Are all games equally cloud-gaming-friendly? an electromyographic approach. In *Network and Systems Support for Games (NetGames), 2012 11th Annual Workshop on*. IEEE, 1–6.

Marc Manzano, Manuel Uruena, M Sužnjević, Eusebi Calle, Jose Alberto Hernandez, and Maja Matijasevic. 2014. Dissecting the protocol and network traffic of the OnLive cloud gaming platform. *Multimedia systems* 20, 5 (2014), 451–470.

Christina Moreno, Nicolas Tizon, and Marius Preda. 2012. Mobile cloud convergence in gaas: A business model proposition. In *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 1344–1352.

Sumedh Mungee, Nagarajan Surendran, and Douglas C Schmidt. 1999. The design and performance of a CORBA audio/video streaming service. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*. IEEE, 14–pp.

Arto Ojala and Pasi Tyrvainen. 2011. Developing cloud business models: A case study on cloud gaming. *IEEE software* 28, 4 (2011), 42.

OnLive. OnLive. (????). http://onlive.com/ Accessed: 2015-11-20.

Mehdi Semsarzadeh, Mahdi Hemmati, Abbas Javadtalab, Abdulsalam Yassine, and Shervin Shirmohammadi. 2014. A video encoding speed-up architecture for cloud gaming. In *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*. IEEE, 1–6.

Ryan Shea, Jiangchuan Liu, Edith Ngai, and Yong Cui. 2013. Cloud gaming: architecture and performance. *Network, IEEE* 27, 4 (2013), 16–21.

Shu Shi, Cheng-Hsin Hsu, Klara Nahrstedt, and Roy Campbell. 2011. Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. In *Proceedings of the 19th ACM international conference on Multimedia*. ACM, 103–112.

Richard Süselbeck, Gregor Schiele, and Christian Becker. 2009. 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*. IEEE, 1–2.

Mirko Suznjevic, Justus Beyer, Lea Skorin-Kapov, Sebastian Moller, and Nikola Sorsa. 2014. Towards understanding the relationship between game type and network traffic for cloud gaming. In *Multimedia and Expo Workshops (ICMEW), 2014 IEEE International Conference on*. IEEE, 1–6.

Hao Tian, Di Wu, Jian He, Yuedong Xu, and Min Chen. 2015. On achieving cost-effective adaptive cloud gaming in geo-distributed data centers. *Circuits and Systems for Video Technology, IEEE Transactions on* 25, 12 (2015), 2064–2077.

Yoav Tzruya, Alex Shani, Francesco Bellotti, and Audrius Jurgelionis. 2006. Games@ Large-a new platform for ubiquitous gaming and multimedia. *Proc. BBEurope, Geneva, Switzerland* (2006), 11–14.

Shaoxuan Wang and Sujit Dey. 2009. Modeling and characterizing user experience in a cloud server based mobile gaming approach. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 1–7.

Shaoxuan Wang and Sujit Dey. 2012. Cloud mobile gaming: modeling and measuring user experience in mobile wireless networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 16, 1 (2012), 10–21.

Jiyan Wu, Chau Yuen, Ngai-Man Cheung, Junliang Chen, and Chang Wen Chen. 2015. Enabling adaptive high-frame-rate video streaming in mobile cloud gaming applications. *Circuits and Systems for Video Technology, IEEE Transactions on* 25, 12 (2015), 1988–2001.

Received February 2007; revised March 2009; accepted June 2009

Online Appendix to:
Cloud Gaming: A Survey

A. THIS IS AN EXAMPLE OF APPENDIX SECTION HEAD

Channel-switching time is measured as the time length it takes for motes to successfully switch from one channel to another. This parameter impacts the maximum network throughput, because motes cannot receive or send any packet during this period of time, and it also affects the efficiency of toggle snooping in MMSN, where motes need to sense through channels rapidly.

By repeating experiments 100 times, we get the average channel-switching time of Micaz motes: 24.3 μs. We then conduct the same experiments with different Micaz motes, as well as experiments with the transmitter switching from Channel 11 to other channels. In both scenarios, the channel-switching time does not have obvious changes. (In our experiments, all values are in the range of 23.6 μs to 24.9 μs.)

B. APPENDIX SECTION HEAD

The primary consumer of energy in WSNs is idle listening. The key to reduce idle listening is executing low duty-cycle on nodes. Two primary approaches are considered in controlling duty-cycles in the MAC layer.

DOI: 0000001.0000001