Toward Multiplayer Cooperative Cloud Gaming

Wei Cai
The University of British Columbia

Fangyuan Chi
The University of British Columbia

Xiaofei Wang
Tianjin University

Victor C.M. Leung
The University of British Columbia

Cloud gaming has been attracting increasing attention in the game industry. Nevertheless, the benefits of cloud-gaming platforms in facilitating multiplayer games have not been widely discussed in the literature. This article reveals this emerging trend and discusses a supporting network architecture for multiplayer cooperative cloud gaming. The article further examines two existing modalities that adopt cooperation among neighboring players to optimize the quality of service in cloud-gaming services.

Cloud gaming,[1] also known as *gaming as a service* (GaaS), empowers the gaming industry with scalable cloud-computing resources. Different from the traditional method of distributing game software, cloud gaming introduces a third party—i.e., the cloud service provider—into this high-profit industry. In this approach, individual game players no longer need to purchase copies of game software. Instead, cloud service providers pay the copyright fees and host these games on their virtual machines (VMs) to derive continuous revenue. In fact, the only difference between the cloud service provider and conventional multiplayer online game operators is that, rather than handling only the transactions between players, the cloud server utilizes its own computing and storage resources to execute the games for the players. The advantages of cloud-gaming systems include scalability, cost-efficiency, cross-platform support, and an effective antipiracy solution.

Nevertheless, researchers in both industry and academia have yet to explore cloud gaming's huge potential in multiplayer scenarios, which involve more than one player in the same game environment at the same time. In such scenarios, players can interact with each other in partnership, competition, or rivalry, which provides them with opportunities for social communication that are absent in single-player games.

In this article, we consider multiplayer games such as massively multiplayer online role-playing games (MMORPGs—e.g., *World of Warcraft* and *Lineage*) and small-scale networked games

Copublished by the IEEE CS and IEEE ComSoc
2325-6095/18/$33.00 USD ©2018 IEEE

(e.g., *StarCraft*, *Diablo*, and *League of Legends*). Owing to rapid developments in both computer and broadband-network technologies, MMORPGs are becoming a very important part of online entertainment. Game players tend to get connected to play games with a group of peers instead of against AI. This change enhances the gameplay experience and makes players more likely to become attached to this type of entertainment. Similarly, small-scale networked games are also attracting many players owing to their flexibility to form groups to conduct short sessions of gaming.

In fact, applying the cloud-gaming concept to multiplayer gaming introduces additional benefits as follows:

- *The nature of connectivity.* A critical drawback of the cloud-gaming paradigm is the required network connectivity. Indeed, overhead is incurred to establish and maintain the network connections between the cloud and the players' terminals during the gaming session. This may be a reason for customers staying away from cloud gaming. However, this worry is unlikely to impact the decisions of users when it comes to multiplayer games, since network access is already mandatory for such games.
- *Temporary engagement.* An important feature of cloud gaming is that it enables gameplay without downloading and installing the game. Such click-and-play becomes more attractive in a multiplayer scenario where people in proximity are engaged to play the same game temporarily. For instance, several friends at a party might decide to play some video games together but cannot find a game that is installed on all their smartphones. In this case, the benefit of click-and-play cloud gaming becomes self-evident.
- *Gaming fairness*. How to achieve fairness between multiple players is a crucial issue in designing online games. Because the players are competing with each other in the same game scene, the system should respond to their actions immediately. Players of conventional online games may suffer from unfairness, especially when the quality of service (QoS—e.g., the latency or packet loss rate) of their network connections varies. With cloud gaming, players' gaming instances are hosted in the cloud. Hence, message exchanges between instances occur inside the cloud, which makes it easier to maintain a guaranteed QoS. The cloud-gaming system may be able to adapt itself to a terminal's network to provide more fairness. For example, previous research proposed adjusting rendering parameters to reduce video quality for those players with poor network access.[2] With reduced video quality, players with less capable devices or experiencing poor network conditions can be treated fairly in multiplayer games.

Owing to the above advantages, multiplayer gaming may become the dominant modality of cloud gaming.

Figure 1 illustrates a multiplayer setting in a conventional cloud-gaming framework. It follows a gaming-on-demand model, which has been adopted by some companies, such as OnLive, Gaikai, and G-cluster. These service providers host the video games on their cloud servers and stream the gaming video frames to players' terminals over the Internet. The gaming interactions triggered by game players are transmitted to the cloud server over the same network.

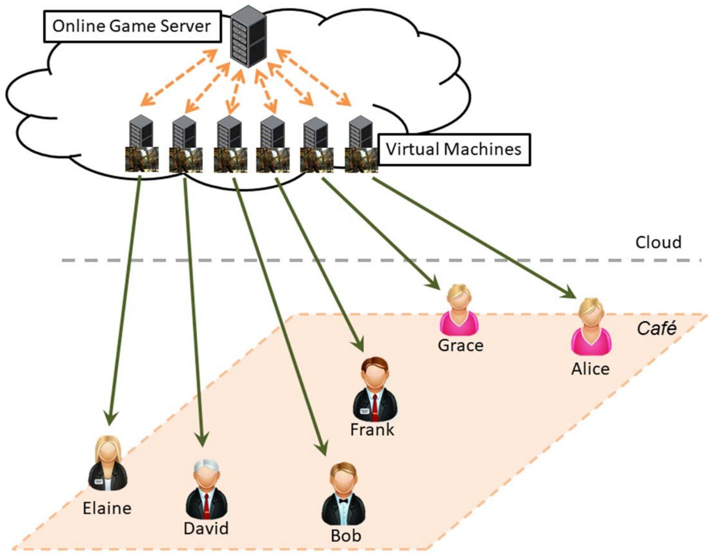

Figure 1. A conventional multiplayer cloud-gaming framework.

With this approach, the cloud-gaming service enables the players to play sophisticated games despite the limited hardware capacity of their terminals, at the expense of increased communication and network consumption. To enable interactions among multiple players in game sessions, the VMs hosted in the cloud communicate with the online game server. Any bottlenecks in the system will likely be due to the high bandwidth consumption by real-time video transmission or response latency issues for players.

Apparently, the use of video compression[2] and network-aware adaptation[3] to compensate for insufficient bandwidth in multiplayer cloud gaming is the same as for the single-player mode, which has been well investigated in previous research. However, approaches to dealing with response latency do differ. Researchers have recently performed studies on optimizing the quality of experience (QoE) for such systems.**Error! Reference source not found.** They indicated that the interplayer-delay problem is threefold:

- minimizing the interplayer delay among interacting players,
- preserving good-enough latency between players' devices and their corresponding VMs, and
- always considering the limitations of the available resources of cloud datacenters.

The authors suggested that the VMs for different players are not always in the same cloud datacenter in reality, and the online game server is always remote from the cloud as well. Therefore, the placement of VMs assigned to players becomes a critical issue. The authors formulated this problem to minimize the overall latency, in order to satisfy the 500 ms requirement in video-gaming systems.[5]

This article investigates the possibility of system optimization with multiplayer cooperative cloud gaming from different perspectives. The motivation of this work comes from two aspects. First, the different content displayed to distinct players in a common game scene is likely to be highly correlated. This is an intuitive deduction, as the avatars tend to interact with each other (collaborating or competing) in a particular scene. Hence, the corresponding gaming videos and virtual environments have high degrees of similarity. Second, the players engaging in the games are in close proximity to each other in some scenarios; this proximity can be physical (e.g., in a common space) or virtual (e.g., within a subnet). This is a typical case when friends connect to each other at a social event or in a high-speed subnet.

On the basis of these aspects, this article presents the idea of multiplayer cooperative cloud gaming, which aims to utilize the high-speed interconnectivity of players' proximity to share their gaming content. In other words, our target is to explore the similarities among players' content and thus construct cooperation mechanisms to optimize system performance. In the following, we discuss the network architecture supporting multiplayer cooperative cloud-gaming systems and explain two future paradigms of this type of cooperation.

NETWORK ARCHITECTURE

As we stated before, system optimization relies on the interconnectivity of players through high-speed networks. It is of great importance to explore how we can utilize different forms of network connectivity and various network access technologies for more satisfactory gaming services, with respect to individual players' physical locations, gaming devices, and QoS requirements. To this end, we first study the underlying network architectures.

Device-to-Device Cooperation

Recent advances in wireless-communication technologies have made it possible for us to choose from a larger number of short-range device-to-device (D2D) communication technologies to provide high flexibility to connect players' terminals for cooperative cloud gaming. Undoubtedly, Bluetooth and Wi-Fi (in both the infrastructure mode and direct mode) are the most popular ones, while the emerging LTE (Long-Term Evolution) D2D communication technique in the LTE Advanced standard[6] will potentially be another popular choice. LTE D2D offers a long communication range (up to hundreds of meters) over the licensed spectrum; hence, the QoS—i.e., link delay and bandwidth—can be mostly guaranteed.

Within a small area, players in close proximity can always cooperate with each other by establishing D2D connections, as illustrated in Figure 2. Not only can they have more fun when competing with friends nearby, they can also directly share video frames or program components with each other in such an ad hoc network, so that gaming resources can be reused efficiently. Particularly, the small transmission delays among players' devices strongly encourage the realistic deployment of video-sharing cooperative cloud gaming.

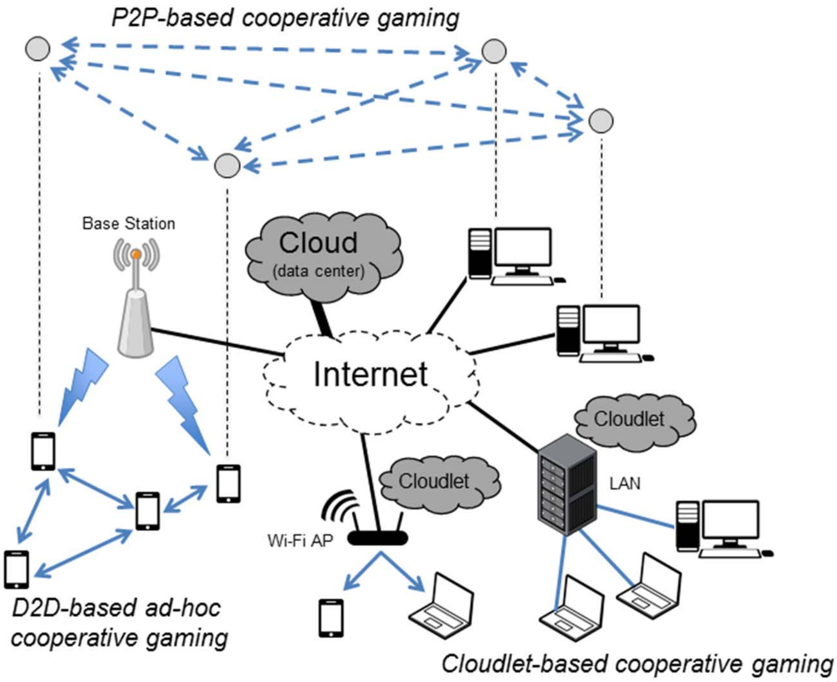

Figure 2. A network architecture for multiplayer cooperative cloud gaming.

However, in D2D-based networks, flexible cooperation without infrastructure support is always constrained by players' locations, and the gaming services may become fragile if players move around, leading to frequent link disconnections. Therefore, how to design a mobile ad hoc cloud-gaming system with self-organization capability becomes interesting and challenging.

Peer-to-Peer Cooperation

To extend gaming services to a large number of people in many places, peer-to-peer (P2P) cooperation could be a better option. A P2P networking overlay consists of virtual connections between the physical terminals (or devices) of players via the Internet,[7] as illustrated in Figure 2. Players never care about how they are interconnected, but just enjoy the acceleration of gaming services due to the optimized sharing of cloud-gaming resources—e.g., rendered models, map files, textures, or even whole clients.

In this form of cooperation, QoS cannot be always guaranteed because the gaming performance relies mostly on the conditions of overlay links and other peers. Therefore, players still need to be within neighboring networks to better cooperate with each other. For instance, a subnet in a residential area is acceptable.

The maintenance of the P2P overlay also becomes essential. Fast peer discovery and recovery mechanisms are desired so that players with high mobility can continue the game whenever some players leave and others join. One open issue may be how to adapt existing Internet P2P techniques for cooperative cloud gaming.

Cloudlet Cooperation

In order to accommodate widely distributed players within the centralized cloud, researchers recently have proposed deploying *cloudlets*[Error! Reference source not found.] in the vicinity of players to better support gaming services. As shown in Figure 2, cloudlets are extended clouds running at the edge of the networks. They are able to run most of the cloud-gaming functions independently, so that players can experience much better gaming performance in terms of latency, transmission speed, and so on. Cloudlets can be embedded at Wi-Fi access points, routers, gateways, and even base stations, by incorporating the necessary infrastructure support at the expense of some additional deployment cost.

Cloudlets can also effectively play the role of caches between the cloud and players. This enables effective utilization of the cloud-gaming resources and allows the whole system's computational load and network traffic to be balanced and optimized as much as possible.

VIDEO-SHARING COOPERATIVE CLOUD GAMING

Video-sharing cooperative cloud gaming is built on the conventional cloud-gaming framework, where the contradiction between the high-bandwidth requirement of real-time video streaming and insufficient and unstable network resources remains the most crucial issue. We seek cooperative solutions in video sharing among multiple players, in order to minimize the overall transmission rate from the cloud to the terminals.

The main idea of video-sharing cooperative cloud gaming is to utilize *P-frames* to exploit the correlations among peer players' video frames, similar to frame sharing in multiview video streaming.[9] In video encoding, a P-frame holds only the changes in the image relative to the previous frame. For example, in a scene where an avatar moves across a static background, only the avatar's movements need to be encoded. The encoder does not need to store the unchanged background pixels in the P-frame, thus reducing the frame size.

To extend the use of P-frames in a video stream, we apply a P-frame to encode peer players' frames. Figure 3 illustrates an encoding structure with four players in the same game session. The squares represent the streamed video frames, in which the pair (t, i) denotes a frame sent to player i in time slot t. The arrows represent the frame encoding dependencies. As depicted, the video frames are encoded in two dimensions. We denote those P-frames predicted by their previous frames in the same video stream as *intrastream P-frames* and those predicted from peers' game video frames as *interstream P-frames*.

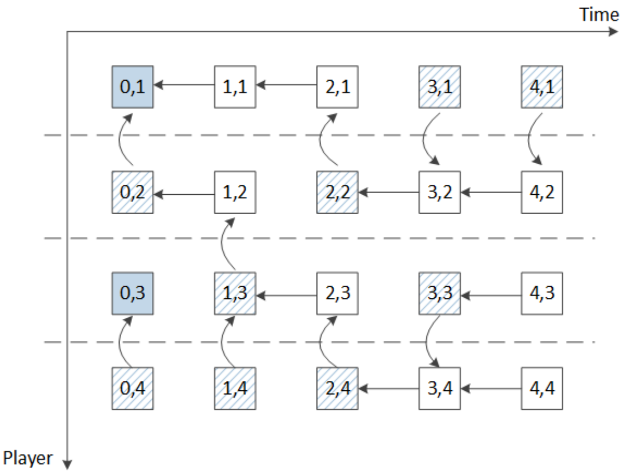

Figure 3. An encoding structure in video-sharing cooperative cloud gaming.

The Framework

The architectural framework of video-sharing cooperative cloud gaming is illustrated in Figure 4. Similarly to existing cloud-gaming proposals, instances of game engines are hosted in the cloud to provide gaming services to players. They generate real-time videos in response to players' actions in the gaming scenarios.

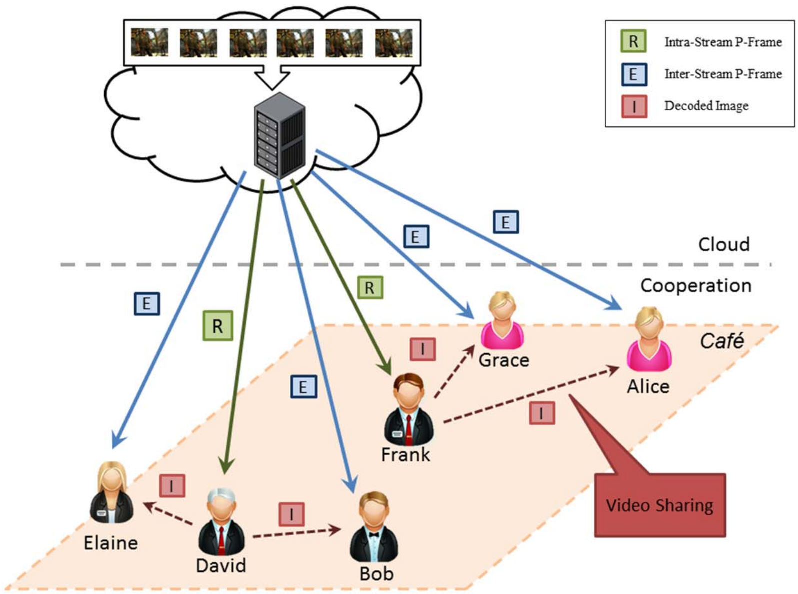

Figure 4. Video-sharing cooperative cloud gaming.

The key features of the proposed system include cooperative encoding in the cloud layer and cooperative decoding in the terminal layer. Cooperative encoding can be performed at a cloud gateway for multiple players' gaming video, which exploits the correlations between video frames for different players to perform centralized encoding with the purpose of minimizing the server transmission rate. With this approach, the video frames to be transmitted to the players are encoded more efficiently.

On the other hand, cooperative decoding is a cooperative mechanism conducted by the participating mobile devices. They share their decoded video frames with other players, in order to help their peers decode the streaming video.

Cooperative decoding can be constructed with two different mechanisms. First, the terminals can utilize P2P overlay networking or D2D connections to transmit their decoded frames to those peers in need. In this case, interstream P-frame decoding is performed on the terminals. Second, these decoding procedures can be automatically completed in a cloudlet, which receives cooperative encoded frames from the cloud and distributes the decoded video frames to the associated terminals. The cloudlet solution exhibits higher efficiency, at the expense of additional hardware requirements.

Challenges and Research Issues

Video-sharing cooperative cloud gaming reduces bandwidth consumption with cooperative encoding. However, it brings several challenges and research issues. First, introducing reference-based encoding brings additional overhead to the system—e.g., an increased workload in the video encoder server. A common assumption in the cloud-computing paradigm is that all game engines and video encoder servers are extremely powerful to perform the encoding, owing to scalable computing resources. However, the cost of cloud resources cannot be neglected in practice. Hence, optimization inside the cloud should be considered.

Second, decoding video frames from predicted images requires additional cloudlet support or using an ad hoc networking model that enables terminals to decode the videos cooperatively. The energy consumption of mobile terminals performing cooperative video decoding and ad hoc network communication can be a critical issue. Furthermore, system performance in the presence of device mobility could also be an important issue.

Researchers have presented a preliminary case study concerning the feasibility of video sharing.[10] In order to minimize network transmissions by exploiting correlations among video frames streamed to different players, there are two prerequisites. First, a player's gaming video should dramatically change over time, which implies that there exist intrastream P-frames with large sizes. Second, there must be similarities of video frames among the players, resulting in small interstream P-frames that can replace large intrastream P-frames, and hence a reduction of the server transmission rate.

In order to verify the feasibility, we conducted empirical studies on the frame size of these two types of P-frames. *Diablo 2*, a classic multiplayer action role player game (ARPG), was selected as the measurement subject because of its more static perspective, which promised to yield notable reductions. Two players (a sorceress and an assassin) were connected to each other over a wireless local area network and started their ventures simultaneously.

We captured their gaming screens (with 800 × 600 resolution) and encoded them at 30 frames per second by FFmpeg, an H.264 codec widely used in online video-streaming and cloud-gaming systems.[11] With an encoding setting of 0 B-frames and an infinite group of pictures, the frame sizes of the video sequence encoded as intrastream P-frames were recorded. We also extracted image sequences from the two gaming videos and used the FFmpeg codec to encode player 2's images into P-frames by predicting from player 1's concurrent image. Thus, a set of interstream P-frames of player 2 depending on player 1 were derived.

Given the number of intrastream and interstream P-frames, it was straightforward to calculate the optimal encoding solution for the case in which player 2 can exploit the similarity between his and player 1's video sequences. According to the study, the amount of data transmitted was reduced from 111,906,600 bytes to 85,871,751 bytes—a significant 23.26% reduction.

Such performance will ease QoS provisioning for cloud-gaming systems, especially when network bandwidth is insufficient. Furthermore, the more avatars that are involved in cooperative sharing, the higher probability the correlations can be explored among players. Therefore, cooperative video streaming can be further enhanced if the system accepts more than two players.

COMPONENT-SHARING COOPERATIVE CLOUD GAMING

Although a video-based cloud-gaming architecture has been widely studied in academia and used in industry, this architecture still has a few flaws in practical applications. With the almost unlimited processing power provided by the cloud, the biggest challenge that remains is the high bandwidth consumption for video streaming. To alleviate this, the approach of offloading the graphic engine back to the terminals[12] has been introduced to cloud gaming, following the state-of-art partial-offloading systems—e.g., CloneCloud.[13]

This idea is based on Cascade's[14] parallel-programming architecture for game applications that run component-based game tasks in parallel. Specifically, Cascade allows programmers to explicitly express task dependencies in a task dependency graph, which is traversed at runtime by Cascade's job manager when task-scheduling schemes are run. The performance of such an approach has been evaluated by benchmarks; the results indicate that considering the interrelationships between components improves the performance of parallel programming in games (compared to other distributed job-processing algorithms) in terms of reduced execution time.

On the basis of this idea, the concept of component-sharing cloud gaming has been proposed. Instead of running every single piece of game logic on the centralized server and requiring players to transmit each input to the server for processing, component-sharing cloud gaming allows players to share relevant game components with their local peers directly. By handing over certain processing tasks to players, less stress is placed on the access network and the cloud server. This novel architecture enables nearby players who are in possession of mobile devices to play games cooperatively.

Specifically, instead of connecting solely to the cloud, nearby players are also able to utilize and share resources, including processing and storage, over an ad hoc network by forming a cloudlet. For instance, devices with a strong processing capacity for video rendering can eliminate the high expense of real-time video streaming by rendering the video frames locally. To this end, computational tasks are no longer offloaded to the cloud as in other cloud-gaming architectures. Instead, this architecture makes it the first priority to offload the tasks to the local cooperative infrastructure.

The Framework

The architectural framework of component-sharing cooperative cloud gaming is illustrated in Figure 5. As depicted, the players in the cooperative-gaming session perform two actions simultaneously: progressive downloading of segmented game resources and collaborative task allocation. In progressive downloading, game resources, including game data (e.g., textures, geometry, objects, and nonplayer character metadata) and game logic (e.g., game rules, scripts, rendering, and network access) are progressively downloaded onto players' game devices. After the downloading, each player's game progress is advertised in the ad hoc cloudlet, together with the ownership of the downloaded resources.

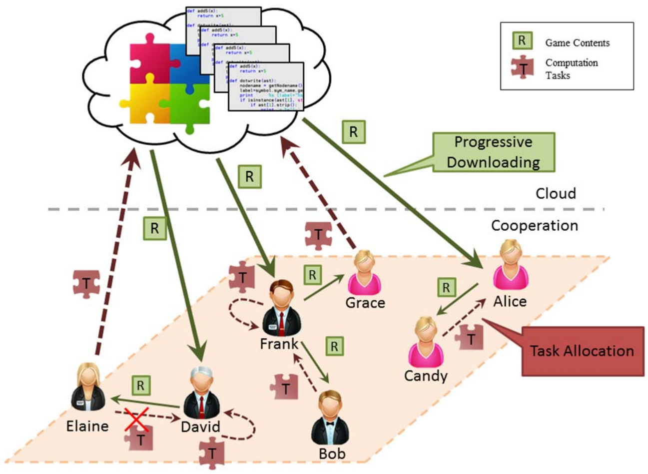

Figure 5. Component-sharing cooperative cloud gaming.

With global knowledge of players' game progress, a decision-making process is designed to locate the best candidate to acquire the needed resources on the basis of the current local and remote game progress, as well as the network condition. If none of the players in the ad hoc cloudlet own the required resources, the system acquires the resources from the cloud directly.

Similarly, prior to offloading each computational task to the cloud, the system runs a decision-making process to select the best candidate to offload the task to, on the basis of the status of each player's device (e.g., CPU and storage status). To ensure system availability, the system offloads the task to the cloud if the local players' terminals do not meet certain requirements (e.g., CPU and storage requirements).

Given that these two processes depend on the players' game progress as well as device status, a global knowledge of players' game status should be maintained by each player and updated in real time. To achieve this, each player is required to periodically send a beacon so that all players are aware of their one-hop neighbors and their game status. If a cloudlet is introduced to the system, we can generally consider it as a powerful peer that is able to execute all computational tasks with connections to all gaming devices.

Challenges and Research Issues

As we mentioned earlier, component-sharing cooperative cloud gaming should be built on the concept of a component-based gaming architecture. The most commonly seen challenge for such an architecture is decomposition complexity or, to be more specific, the decomposition level (e.g., the data level, task level, or function level). The decomposition level defines the frequency with which components interact with each other, and thus the rate of data exchange between components. It is actually the determining factor in the ad hoc cloudlet-based gaming architecture. Since components could be remotely executed, a high data exchange rate (a high decomposition level) between remote components could be highly detrimental to both system performance and the communication cost. Because the decomposition level varies with the game genre, how to find the appropriate level of decomposition remains the biggest challenge.

Furthermore, the beacon messages and the memory used to acquire and store the neighbors' gaming statuses are overheads that require further modeling and analysis. Moreover, efficient and decentralized service discovery, device discovery, and membership management mechanisms should be carefully designed to ensure the scalability of the system.

Recent preliminary research on this topic was concerned with the efficiency of component sharing, from the perspectives of progressive downloading and the costs of task offloading.[Error! Reference source not found.] It addressed this topic by simulations with a D2D network architecture as shown in Figure 2. Transmissions in the D2D network are assumed to have a negligible cost and delay compared to the cloud-to-terminal network. Thus, making it the first priority to offload the tasks to, and acquire the game resources from, the ad hoc cloudlet reduces the overall energy cost for data transmission.

CONCLUSION

Multiplayer engagement will be the next growth direction of cloud-gaming services. By presenting cutting-edge research, we have explored potential opportunities for developers and operators to provide more high-quality content with multiplayer interaction, rather than utilizing the cloud merely as a computing resource. We discussed two ongoing research directions for multiplayer cooperative cloud-gaming systems, including the sharing of video frames and components. We anticipate that our work will motivate more researchers to pursue further studies on implementing and optimizing such systems.

REFERENCES

1. R. Shea et al., "Cloud gaming: architecture and performance," *IEEE Network*, vol. 27, no. 4, 2013, pp. 16–21.
2. S. Chuah, N. Cheung, and C. Yuen, "Layered coding for mobile cloud gaming using scalable Blinn-Phong lighting," *IEEE Transactions on Image Processing*, vol. 25, no. 7, 2016, pp. 3112–3125.
3. S. Wang and S. Dey, "Cloud mobile gaming: Modeling and measuring user experience in mobile wireless networks," *Mobile Computing and Communications Review*, vol. 16, no. 1, 2012, pp. 10–21.
4. Y. Chen, J. Liu, and Y. Cui, "Inter-player Delay Optimization in multiplayer cloud gaming," *Proc. IEEE 9th International Conference on Cloud Computing* (CLOUD 16), 2016, pp. 702–709.
5. M. Claypool and K. Claypool, "Latency and player actions in online games," *Communications of the ACM*, vol. 49, no. 11, November 2006, pp. 40–45.
6. K. Doppler et al., "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Communications Magazine*, vol. 47, no. 12, 2009, pp. 42–49.
7. B. Knutsson et al., "Peer-to-peer support for massively multiplayer online games," *IEEE Conference on Computer Communications* (INFOCOM 04), 2004, p. 107.
8. G. Lewis et al., "Tactical Cloudlets: Moving Cloud Computing to the Edge," *2014 IEEE Military Communications Conference*, 2014, pp. 1440–1446.
9. D. Ren et al., "Coding structure and replication optimization for interactive multiview video streaming," *IEEE Transactions on Multimedia*, vol. 16, no. 7, 2014, pp. 1874–1887.
10. W. Cai et al., "Quality of experience optimization for cloud gaming system with ad-hoc cloudlet assistance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 12, 2015, pp. 1–14.
11. C. Huang et al., "GamingAnywhere: the first open source cloud gaming system," *ACM Transactions on Multimedia Computing Communications and Applications*, vol. 10, no. 1, 2014, pp. 1–25.
12. D. Meilnder et al., "Bringing mobile online games to clouds," *Proc. of IEEE Conference on Computer Communications Workshops* (INFOCOMW 14), 2014, pp. 340–345.
13. B. Chun et al., "Clonecloud: Elastic execution between mobile device and cloud," *Proc. 6th ACM Conference on Computer Systems* (EuroSys 11), 2011, pp. 301–314.
14. A. Tagliasacchi et al., "Cascade: A parallel programming framework for video game engines," *Proc. of the Workshop on Parallel Execution of Sequential Programs on Multi-core Architectures* (PESPMA 08), 2008.
15. F. Chi et al., "Ad-hoc Cloudlet Based Cooperative Cloud Gaming," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, November 2015, p. 1.

ABOUT THE AUTHORS

Wei Cai is a postdoctoral research fellow in the University of British Columbia's Department of Electrical and Computer Engineering. His research interests include cloud computing, cognitive systems, the Internet of Things, finance technology, and software engineering. Cai received a PhD in electrical and computer engineering from the University of British Columbia. He's a member of IEEE. Contact him at weicai@ece.ubc.ca.

Fangyuan Chi is a software engineer at Veeva Systems, a company that delivers cloud-based business solutions for the global life sciences industry. Her research areas include cloud gaming, ad hoc networks, and parallel computing. Chi received a master's of applied science in electrical and computer engineering from the University of British Columbia. Contact her at fangchi@ece.ubc.ca.

Xiaofei Wang is a professor at Tianjin University's School of Computer Science and Technology. His research interests include future content-centric networking and in-network caching for next-generation mobile networks. Wang received a PhD in electrical engineering and computer science from Seoul National University. He's a member of IEEE. Contact him at xiaofeiwang@ieee.org.

Victor C.M. Leung is a professor of electrical and computer engineering and the holder of the TELUS Mobility Research Chair at the University of British Columbia. His research interests include wireless networks and mobile systems. Leung received a PhD in electrical engineering from the University of British Columbia. He's a fellow of IEEE, the Royal Society of Canada, the Canadian Academy of Engineering, and the Engineering Institute of Canada. Contact him at vleung@ece.ubc.ca.