

A Cloudlet-Assisted Multiplayer Cloud Gaming System

Wei Cai · Victor C.M. Leung · Long Hu

Published online: 9 November 2013
© Springer Science+Business Media New York 2013

Abstract The unstable network connectivity is the bottleneck of providing Gaming as a service (GaaS) for mobile devices. Therefore, the most critical technical challenge is to compress and transmit the real-time gaming video, so that during the gaming session, the expected server transmission rate over the bandwidth-limited mobile network can be minimized, while satisfying the quality of experience for the players. Inspired by the idea of peer-to-peer sharing between multiple players, we propose a cloudlet-assisted multiplayer cloud gaming system, in which the mobile devices are connected to the cloud server for real-time interactive game videos, while sharing the received video frames with their peers via an ad hoc cloudlet. Experimental results show that expected server transmission rate can be significantly reduced compared to the conventional video encoding schemes for cloud games.

Keywords Cloud · Game · Video · Network · Encoding · Cloudlet

This work is based in part on our previous paper titled “Multiplayer cloud gaming system with cooperative video sharing”, Presented at cloudcom2012 workshop.

W. Cai (✉) · V. C.M. Leung
Department of Electrical and Computer Engineering,
The University of British Columbia, Vancouver, Canada
e-mail: weicai@ece.ubc.ca

V. C.M. Leung
e-mail: vleung@ece.ubc.ca

L. Hu
School of Computer Science and Technology,
Huazhong University of Science and Technology, Wuhan, China
e-mail: longhu.cs@gmail.com

1 Introduction

Mobile games contribute a huge number of downloads and consequentially potential profits in the application market. However, the hardware constraints of mobile devices, such as computational power, storage and battery, limit the representation of games. According to this, the promising mobile cloud computing [1] technologies attract attentions on making video games as a service [2]. Since the games are rendered on cloud servers, a low-end computer or mobile device may be used to play any kind of game as long as it is able to play video. Industry has already made their moves in this area. OnLive Game System offers a cloud gaming platform and a cloud desktop solution for Windows, Mac OS, Android and iOS devices. Gaikai is a cloud-based gaming service that allows users to play high-end PC and console games via the cloud and instantly demo games and applications from a webpage or internet-connected device. G-cluster uses IPTV set-top-boxes and the target audience are those gamers who already have gaming consoles, as well as the entire family, with a game selection consisting of casual games as well as high end titles.

However, those cloud video games still suffered from the bottleneck of the internet access. The constraints of bandwidth restricts the bit rate of gaming videos, while the jitter and delay effect the quality of experience for the players. Therefore, to efficiently encode and transmit the real-time gaming video become the most critical issue in cloud video gaming system, especially for mobile devices [3]. There has been a plenty of work focusing on the scalable video encoding for cloud games, e.g., [4]. The authors propose a rendering adaptation technique which can adapt the game rendering parameters to satisfy cloud gaming communication and computation constraints, such that the overall

mobile gaming user experience is maximized. However, the solution is intrinsically reduce the video quality to guarantee the player experience.

Another trend for game industry is online multiplayer scenario. Nowadays, game players are no longer satisfied by enjoying the games alone but prefer to connect to others. The interaction between players brings more challenges in the design of game servers.

In this paper, we study the correlations of the gaming videos in multi-player cloud gaming system and propose a cloudlet-assisted multiplayer cloud gaming system, in which the cloud game server is able to efficiently encode and transmit multiple video streams to a group of players, while those players are able to decode their video in cooperative pattern via a cloudlet constructed by secondary network, such as ad hoc wireless local area network (WLAN), for content sharing. To the best of our knowledge, our work is the first approach in applying ad hoc cloudlet on reducing the server transmission rate for cloud gaming systems. The remainder of this paper is organized as following: Section 2 summarizes the related work in industry and academia, and Section 3 studies the correlations of gaming videos between distinct players in the same game. Afterwards, we model the proposed cloudlet-assisted gaming system in Section 4 and design a video encoder in Section 5. Simulation and analysis are performed in Section 6. At last, we conclude our work in Section 7.

2 Related work

2.1 Correlations of videos frames

Inter frames, e.g., P-frame, is a frame in a video compression stream which is expressed in terms of one or more neighboring frames. Its size, which affects the system performance, is subject to the correlation of the encoded video frames. Light field [5] and multiview [6] video Streaming have conducted studies on this topic. Light field is a large set of spatially correlated images of the same static scene captured using a 2D array of closely spaced cameras. The correlations of light field images are studied and formulated in work [7], which indicated the correlation between two different views to a static scene is related to the geographical distances between each other. Interactive multiview video switching [8] designs a pre-encoded frame representation of a multiview sequence for a streaming server, so that streaming clients can periodically request desired views for successive video frames in time.

2.2 Real-time video encoding

Difference from light field and multiview switching, encoding for cloud gaming videos is essentially a real-time encoding process. The cloud encode video frames immediately after the game scenes are rendered. The fundamental idea of encoding is very simple: to starting with an intra-coded frame, e.g., I-frame, and then follows a certain number of inter frames, such as P-frame [9], distributed source coding (DSC) frames [10], etc. Therefore, how to determine the sequence of various types of frames becomes the most critical problems in video encoding, in order to achieve the tradeoff between bit rate and error rate. In recent video encoding research, the GOP (Group of Pictures) [11] length is set to be adaptive, which implies a structure with one I-frame and variable number of inter frames.

3 Correlation of video frames

In this section, we study the frame size for two types of P-frames: *intra-video P-frame* and *inter-video P-frame*. We denote *intra-video P-frames* as video frames that predict to their previous frames in the same video stream and the *inter-video P-frames* as those predict to peer game videos' frames.

3.1 Intra-video P-frame

Frame size of *intra-video P-frame* is subjected to the variance of the game video content. Apparently, if the avatar is experiencing a more dynamic scene, the P-frame size will be larger. For example, when the players are participating in a large Diablo battle field, where many “wizards” and “demon hunters” are casting magnificent full-screen magics, the video content's amplitude of variation will be dramatic. In this work, we assume P_{intra} , the size of *intra-video P-frames*, follows Poisson Distribution:

$$f(P_{intra}) = \frac{\mu^{P_{intra}} e^{-\mu}}{P_{intra}!} \quad (1)$$

where the mean frame size of $E(P_{intra})$ is μ .

3.2 Inter-video P-frame

P_{inter} , the frame size of *inter-video P-frame*, is subjected to the correlation between two videos for two peering game players. Before modeling P_{inter} , we need discuss the types of multiplayer games: *first-person* game, *second-person* game and *third-person* game. In video games, *first-person* refers to a graphical perspective rendered from the viewpoint of the player character, such as Counter-Strike. The

second-person is similar to first-person but rendered from the back of the player character, which means the player can see their avatar in the screen, e.g., Grand Theft Auto. In contrast, *third-person* games provide the players a view from sky, so-called God-view, to easily observe surrounding environment of the avatar and make quick response. Classic *third-person* games include Diablo, Command & Conquer, FreeStyle, etc. For *first-person* and *second-person*, the game video is generated from the view of the avatar, where the correlation model of video frames are similar to a dynamic light field streaming. In contrast, the correlation model for *third-person* games is much simpler: the videos for peering players are very similar, or even identical, when their avatars are geographically close to each other.

To simplify the evaluation, we consider *third-person* games in this work. Since the players are all in God-view, we calculate the overlap of the two video with the avatars' coordinates, given most of the *third-person* games rolls the map with a avatar-centric manner.

Figure 1 demonstrates two correlated videos for player 1 and player 2. Based on the assumption, we derive $R_{overlap}$, the overlap ratio between peering video frames, as:

$$R_{overlap} = \frac{[W - (|X_2 - X_1|)][H - (|Y_2 - Y_1|)]}{WH} \quad (2)$$

where H is the screen height, W is the screen width, (X_1, Y_1) and (X_2, Y_2) denote the coordinates of the two players in the gaming map. Then we formulate the size of inter-video P-frames P_{inter} as follows:

$$P_{inter} = (1 - R_{overlap}) I \rho \quad (3)$$

where I is the size of intra-encoded video frame and ρ is the compression ratio the encoder is able to achieve.

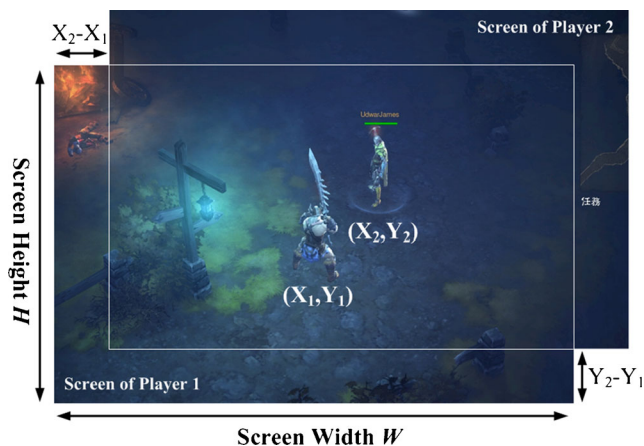


Fig. 1 Correlation of inter-video frames (Diablo III)

4 System modeling

4.1 System overview

The architectural framework of proposed cloudlet-assisted multiplayer cloud gaming system is illustrated in Fig. 2. Similar to the existing cloud gaming work, instances of **Game Engine** are hosted in the cloud to provide gaming services for players. They are connected to a **Multiplayer Game Server** in conventional fashion to facilitate interactions between avatars.

The novelty of our proposed system is to introduce two additional components:

- **Video Encoder Server** is acting as a gateway, which explores the correlations between video frames for distinct players to perform centralized encoding, with the purpose of minimizing server transmission rate. In this work, we consider cloud as a infinite resource provider, therefore, the computational power of the encoder server is unlimited.
- **Ad Hoc Cloudlet** is a cooperative cloudlet constructed by participating mobile devices. They utilize a secondary network, e.g., WiFi ad hoc network, to share the video frames they received from the cloud server. We assume bandwidth for the ad hoc WLAN is sufficiently large for all mobile devices in the immediate neighbourhood to share their frames when needed. Thus, the bandwidth constraint inside the cloudlet is not explicitly modeled.

4.2 Gaming map model

As discussed in Section 3, our work focus on the *third-person* games. Therefore, gaming map is defined as a 2-Dimensional (2D) model with $M \times M$ screen size, since

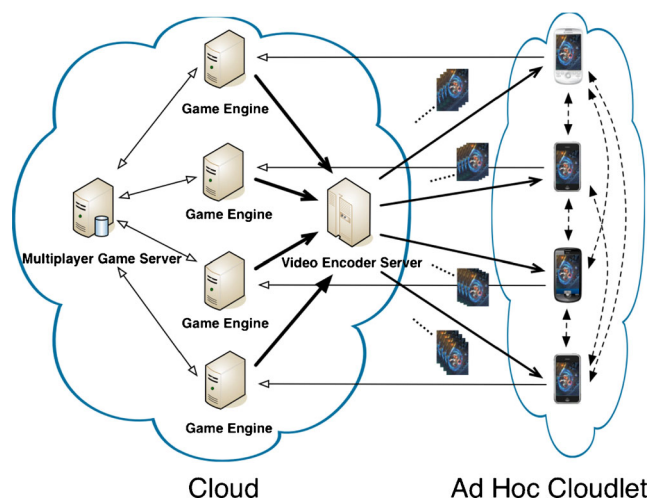


Fig. 2 Cloudlet-assisted multiplayer cloud gaming system

3-Dimensional (3D) scenes are represented as a 2D illustration after rendering from a God-view. Note that, the sizes of gaming maps are various of different types of games. For example, Diablo provides very large map for the players to explore, while FreeStyle, a online basketball game, restricts the players within a basketball field. The impact of map size will be studied later in Section 6. We assume the move unit for players in the map is K pixels and denote W and H as the width and height of the gaming screen. Therefore, the players are able to move their avatars in coordinator (X, Y) , where $X \in [0, MW/K]$, $Y \in [0, MH/K]$. However, if the system restricts the avatar in the center of the screen, the moving area will be restricted to (X, Y) , where $X \in [0, (M - 1)W/K]$, $Y \in [0, (M - 1)H/K]$. In this work, we adopt the later model.

4.3 Player interaction model

Based on the map model, we model the N players' interactions in the same game scene. Specifically, the avatars in the game scenes are able to perform two kinds of movement: *random walk* and *group chase*. This is an observation result from multiplayer games: players are randomly moving and hunting in the world (*random walk*), however, they are also prone to gather together with their teammates or opponents, in order to perform teamwork or competition (*group chase*). For *random walk* movement, the avatars are moving towards a contiguous trajectory in the map. The probability for the avatar to conduct random walk mode is denoted as p_{rw} . In this work, we assume the players are able to stay still or move their characters to adjacent N_{adj} directions with identical possibilities. Therefore, the probability to each adjacent view is $p_{rw}/(N_{adj} + 1)$. For *group chase* movement, the avatar will randomly select another avatar in the scene and move towards it in a certain period of time T_{chase} . Let the probability of group chase movement to be $p_{gc} = 1 - p_{rw}$, and the probabilities of the $N - 1$ target avatars to be equal to $p_{appr} = 1/(N - 1)$.

5 Design of video encoder

In this section, we design the featured **Video Encoder Server**, which efficiently encodes and transmits the video frames to multiplayer in real-time. Since the cloud server is the centralized approach, it is possible to collect information from all players and derive an optimal encoding solution to minimize the average server transmission rate.

5.1 Optimal encoding solution

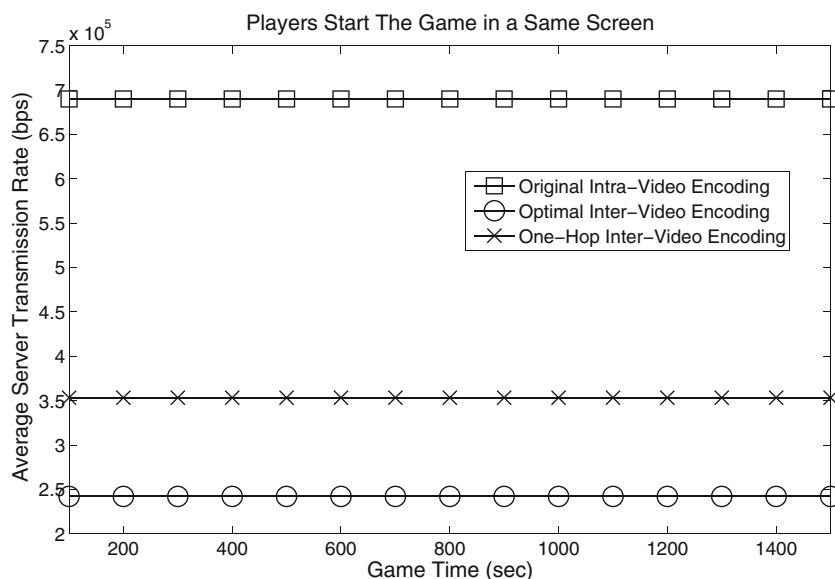
Our solution, called *optimal encoding*, is basically a greedy approach: for each video frame from distinct game players,

the cloud-based encoder compares and selects the minimum frame from three types of encoding solutions, including *I-frames*, *intra-video P-frame* and *inter-video P-frame*. The selection procedure is realized in following 3 steps:

1. *Frame Size Estimation*: The first step is to estimate the frame sizes of *intra-video P-frames* and *inter-video P-frames* for future analysis. We denote $P_{intra}[N]$ as the set of estimated frame sizes of *intra-video P-frames* for N game players, where $P_{intra}[i]$ represents the frame size for i th player. Meanwhile, we store the estimated frame-size of *inter-video P-frames* for all pairs of players in a two-dimensional matrix $P_{inter}[N][N]$, where $P_{inter}[i][j]$ represents the frame size of *inter-video P-frame* for i th player that predicts to the video frame of j th player. The estimation is based on the analysis in Section 3. It is easy to know that the computational complexity of *Frame Size Estimation* is $O(N^2)$.
2. *Grouping*: We denote a “group” for the peers whose video frames are correlated to each other in the specific time. The N players in the same scene may compose one to several groups, given a sufficient large gaming map. Accordingly, the encoder shall perform a grouping algorithm, in prior to optimize the encoding dependency for each group. In our implementation, those $P_{inter}[i][j]$ with larger size than $P_{intra}[i]$ are also eliminated in this step, in order to guarantee that all frames remaining in P_{inter} are “efficiently correlated”. Afterwards, the encoder enumerates the array of P_{inter} to group the m th and n th players into a same group if $P_{inter}[n][m]$ exists. The computational complexity of *Grouping* is $O(N^2)$.
3. *Optimal Encoding*: In each group, the encoder performs optimal encoding to minimize the server transmission rate. One and only one video frame in a group is encoded as an *intra-video P-frame*, while others are encoded as *inter-video P-frames* to minimize the transmission rate. The encoder iteratively selects one video frame to be encoded as *intra-video P-frame* at a time and constructs an *inter-video P-frame* dependency structure with minimum server transmission. At last, the best solution among all results is selected. In each iteration, the encoder encodes a different video frames $F[k]$ as *intra-video P-frame* and creates a directed, weighted graph with the set of P_{inter} . Therefore, seeking an optimal solution with minimum transmission cost is to cut a minimum spanning tree from the root of the $F[k]$. In this work, the encoding procedure is implemented as Prim algorithm with computational complexity of $O(N^2)$.

In summary, the computational complexity of *optimal encoding* is $O(N^2)$.

Fig. 3 Players start the game in a same screen



5.2 Multi-hop decoding problem and solution

With the proposed encoder, we are able to determine the optimal solution for inter-video encoding. However, there is a practical issue to be addressed when we consider it as a realistic system: since the encoder groups efficiently correlated video frames and construct a tree to describe the dependency of video frames, a *multi-hop decoding* might be occurred on the mobile devices. For example, given an intra-encode frame $P_{intra}[1]$ for player 1 and two *inter-video P-frame* $P_{inter}[2][1]$ and $P_{inter}[3][2]$ player 2 and 3, the player 3 can only decode his video frame after player 2 decode its frame by receiving the $P_{intra}[1]$ from player 1, which introduces larger system latency. However, gaming

applications are very sensitive to latency; therefore, *multi-hop decoding* might affect the Quality of Experience (QoE) for the players.

The solution to this problem is basically a greedy approach: the frame $F[x]$ with most dependent frames $P_{inter}[y][x]$ in each group is selected to be encoded as $P_{intra}[x]$, and all $F[y]$ are then encoded as $P_{inter}[y][x]$. This process continues on remaining frames until all videos in the group are encoded. With this *One-Hop Inter-Video Encoding*, we restrict the *inter-video P-frame* decoding into one-hop, thus, to achieve acceptable decoding latency. Compared to *optimal encoding*, *One-Hop Inter-Video Encoding* has a relatively lower computational complexity of $O(\log N)$, which also reduces the workload for the cloud.

Fig. 4 Players start the game in random position

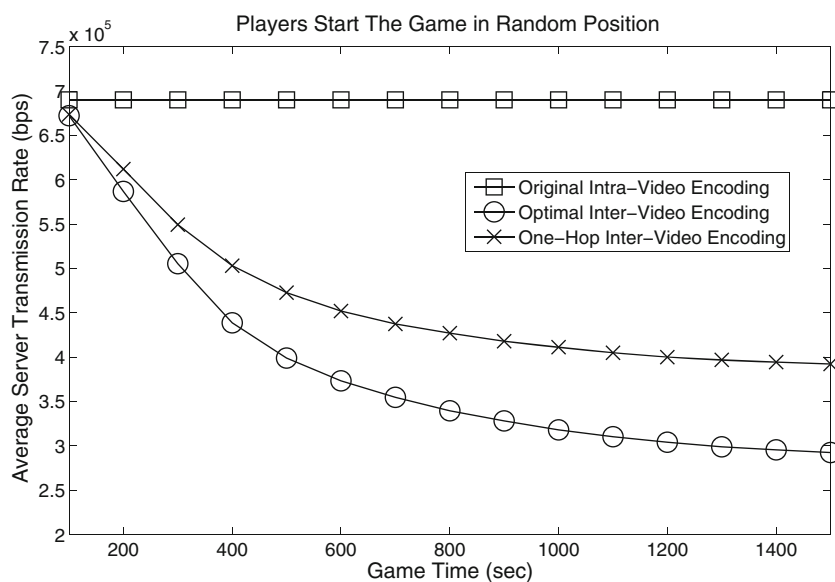
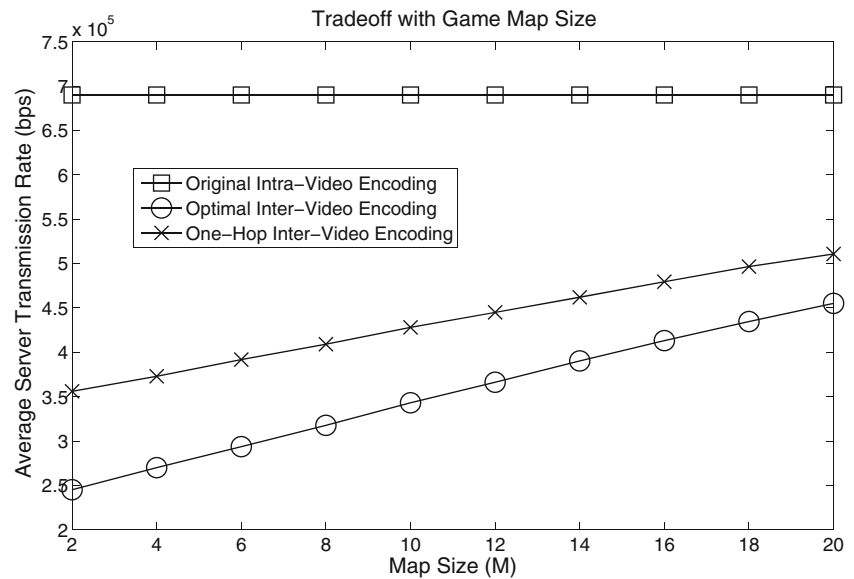


Fig. 5 Tradeoff with game map size

6 Simulation

6.1 Experimental setup

To validate the performance of our proposed system and encoding schemes, we set up the following experiments. For the video data, we download the images from Stanford bunny light field set [12], each image of size 1024×1024 . To encode I- and P-frames, we used a H.263-based codec in [10]. Quantization parameters were set so that the Peak Signal-to-Noise(PSNR) of the encoded frames was around 32dB. As described in Section 3, we assume the P-frames for intra-video and inter-video follows designated distribution and formulation.

Default values for parameters of the simulation were set as follows: number of players in the system N is 8, average game time T is 1000 second, fps(frame per second) is 24, the game map is $M \times M$ screen size, where M is 4, mean intra-video P-frame size μ is 28749 byte, intra-encoded video frame size I is 245296 byte, compression ratio for inter-video frame size ρ is 0.7, screen width W is 1024 pixels and height H is 1024 pixels, pixels for avatar's each move K is 32 pixels, probability of random work P_{rw} is 0.7, time for each chase approach T_{chase} is 5, the avatar is able to move to adjacent 8 directions, which results in $N_{adj} = 8$.

We use the server transmission rate as performance metrics, since the proposed system is aimed to reduce the burden of the network access. We compare the performance of

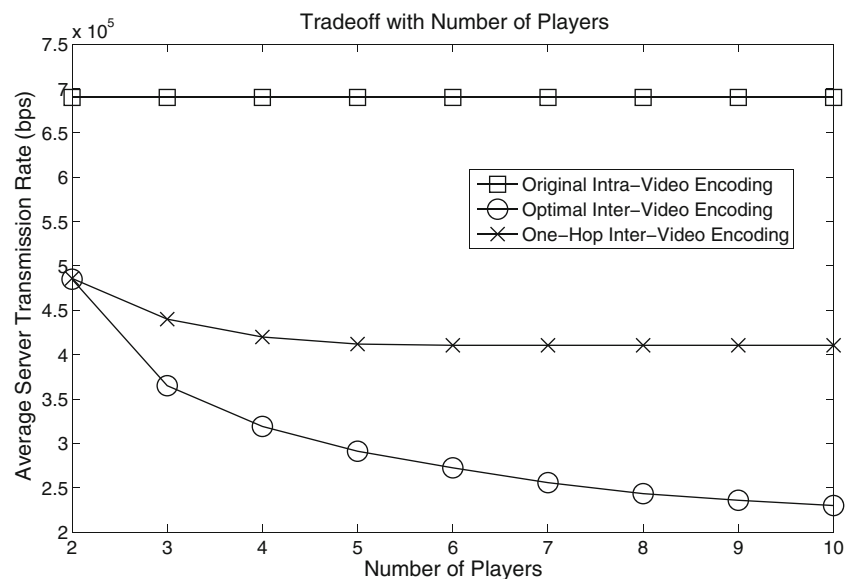
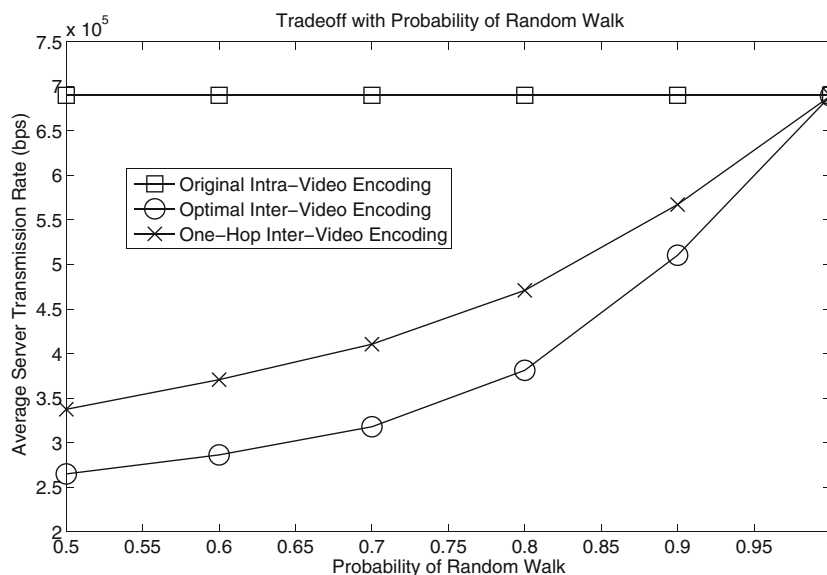
Fig. 6 Tradeoff with number of players

Fig. 7 Tradeoff with probability of random walk



three schemes: *original intra-video encoding*, *optimal inter-video encoding* and *one-hop inter-video encoding*. *Original intra-video encoding* is the simplified version of transitional video encoding, in which the system is able to encode the video frames with one I-frame and infinite successive P-frames ($N_{GOP} = \infty$), which achieves the lowest server-to-client bandwidth in gaming video transmission. *Optimal inter-video encoding* and *one-hop inter-video encoding* are the schemes proposed in Section 5.

6.2 Experimental result

We perform series of simulation to study the impacts of the parameters on the system and the proposed encoding schemes.

The initial positions of the avatars are very sensitive element for the system, since they are the key factors to determine the correlations between gaming videos. Figure 3 illustrates the simulation result that all of the players are starting their games in a same screen. Apparently, the *optimal inter-video encoding* scheme achieves lowest server transmission rate, while the more practical *one-hop inter-video encoding* scheme also has a significant decrease in bandwidth. In contrast, Fig. 4 shows the result for the case that each avatar choose a random starting coordinate in the gaming map. The gain of video sharing is not significant at the begin, since the avatars are randomly distributed in the map. However, the players are gathering together along the gaming procedure, which derives a significant reduction on the throughput of the cloud server.

Fig. 8 Tradeoff with time for each chase approach

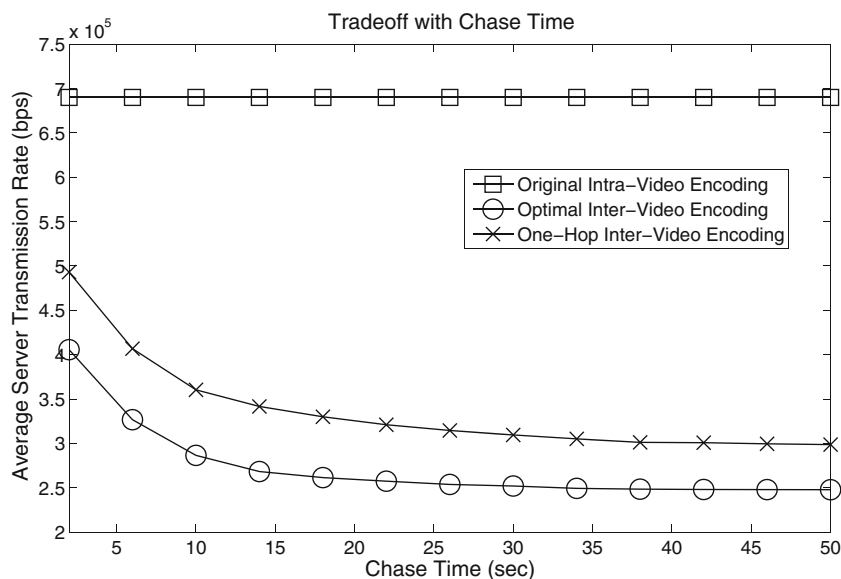
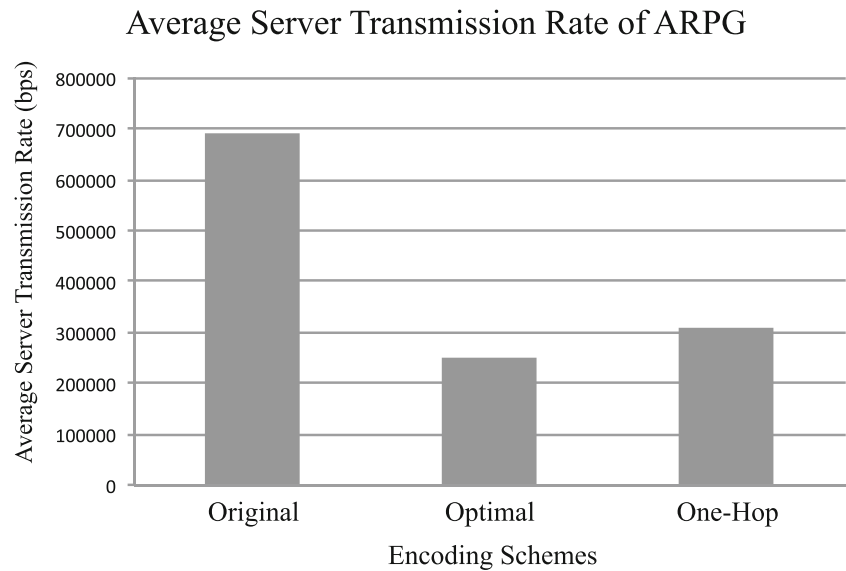


Fig. 9 Server transmission rate of ARPG simulation

To better investigate the impact of players' interaction, we adopt the *random starting position* model for following experiments. Figure 5 studies the size of gaming map. According to the result, larger map will reduce the opportunity of correlations between videos, thus, yields poorer performance of the proposed encoding schemes. Figure 6 depicts the relations between transmission rate and the number of players in the system. With a fixed gaming map, more players will increase the chance of correlated videos generated in the cloud server, therefore, achieve a lower transmission rate for each player. An interesting phenomenon from this figure is that, as the increase of number of players, the decrease rate for *optimal inter-video encoding* scheme is more significant than *one-hop inter-video encoding* solution. The reason is that, multiple hop correlations between gaming videos are prone to occur when more players are connecting to the same game, which benefits the *optimal inter-video encoding* scheme but not the *one-hop inter-video encoding*.

Figures 7 and 8 concern on the impact of the players' behavior. As the matter of fact, the higher probability of random walk is, the chances for the avatars to share similar videos is smaller. Therefore, the performance of proposed encoding schemes are worse. Similarly, the longer time for each *group chase* approach is, the easier for the players to share correlated video frames, thus, results in a significant gain for the cloudlet-assisted system.

6.3 Case study for ARPG

With the simulation platform, we also perform a case study for Action Role-Playing Games (ARPG), e.g., the Diablo 3. As a typical gaming scenario, 3 players involved in a same scene. Map size M is set to 40, since the ARPG game

scenes are larger. In general, the avatars are starting the game together from the saved point, e.g., the camp, or a transport terminal in the battlefield. Therefore, we assume the initial coordinates of avatars are identical. Gaming time is set to 1 hour, which is a typical length for group play. Based on our gaming experiences in Diablo 3, 0.6 and 10 are reasonable settings for the possibility of random work P_{rw} and the time for chasing teammates T_{chase} .

As shown in Fig. 9, for the designated multiplayer ARPG games, the *optimal encoding* scheme reduces the server transmission rate by up to 64%. In the mean time, the more practical *one-hop restricted* encoding scheme also achieves a 54% transmission decrease.

7 Conclusion

In this paper, we discuss the correlation of video frames in multiplayer cloud gaming system and then propose a cloudlet-assisted system for cloud game participants to share their received video frames, thus, to reduce the bandwidth workload for the cloud server. Extensive experiments have shown that the expected overall server transmission rate for multiplayer ARPG is reduced by up to 64% with optimal encoder and 54% with more practical one-hop restricted encoding solution.

References

1. Song W, Su X (2011) Review of mobile cloud computing, in Communication Software and Networks (ICCSN). IEEE, 3rd Int Conf pp 1 C4

2. Wang S, Dey S (2009) Modeling and characterizing user experience in a cloud server based mobile gaming approach. In: Global Telecommunications Conference, GLOBECOM 2009. IEEE, pp 1 C7
3. Chen M (2012) AMVSC: A Framework of Adaptive Mobile Video Streaming in the Cloud. IEEE, Globecom 2012. Anaheim, USA
4. Wang S, Dey S (2010) Rendering adaptation to address communication and computation constraints in cloud mobile gaming. In: Global Telecommunications Conference (GLOBECOM 2010). IEEE 2010, pp 1 C6
5. Levoy M, Hanrahan P (1996) Light field rendering. In: ACM SIGGRAPH, New Orleans, LA
6. Merkle P, Smolic A, Muller K, Wiegand T (2007) Efficient prediction structures for multiview video coding. In: IEEE Transactions on Circuits and Systems for Video Technology, vol 17, no 11, pp 1461 C1473
7. Cai W, Cheung G, Kwon T, Lee S-J (2011) Optimized frame structure for interactive light field streaming with cooperative cache. In: IEEE International Conference on Multimedia and Expo. Barcelona, Spain
8. Cheung G, Cheung N-M, Ortega A (2009) Optimized frame structure using distributed source coding for interactive multiview streaming. In: IEEE International Conference on Image Processing. Cairo, Egypt
9. Cheung G, Ortega A, Cheung N-M (2009) Generation of redundant coding structure for interactive multiview streaming. In: Seventeenth International Packet Video Workshop, Seattle
10. Cheung N-M, Ortega A, Cheung G (2009) Distributed source coding techniques for interactive multiview video streaming. In: 27th Picture Coding Symposium, Chicago
11. Lee J, Shin I, Park H (2006) Adaptive intra-frame assignment and bit-rate estimation for variable gop length in h.264. Circuits and Systems for Video Technology, IEEE Transactions on, vol 16, no 10, pp 1271 C1279
12. Stanford Light Field, <http://lightfield.stanford.edu/lfs.html>



Wei Cai is currently a Ph.D. candidate in Wireless Networks and Mobile Systems (WiNMoS) Laboratory led by Prof. Victor C.M. Leung in the Department of Electrical and Computer Engineering at The University of British Columbia (UBC), Canada. Before joint UBC, he completed a half-year visiting research at National Institute of Informatics, Japan. Wei Cai received M.Sc. and B.Eng. from Seoul National University (SNU) and Xiamen

University (XMU) in 2011 and 2008, respectively. He worked as a research assistant in Multimedia and Mobile Communications Laboratory (MMLab) in SNU for two years. During his undergraduate study, he served as the president of student union in software school and participated in a one-year exchange program in Department of Computer Science and Engineering at Inha University, Korea. His research interests include Gaming as a Service, Cloud-Assisted Mobile Computing, Cross-Platform Software System, Online Gaming, Intelligent Mobile Agent System, Interactive Multimedia, etc.



Victor C.M. Leung is a Professor and holder of the TELUS Mobility Research Chair in Advanced Telecommunications Engineering in the Department of Electrical and Computer Engineering at the University of British Columbia in Vancouver, Canada. His research interests are in the broad areas of wireless networks and mobile systems, in which he has co-authored more than 600 technical papers in international journals and conference

proceedings. Several papers co-authored by him and his students have won best paper awards. Dr. Leung is a registered professional engineer in the Province of British Columbia, Canada. He is a Fellow of IEEE, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He has served on the editorial boards of the IEEE Journal on Selected Areas in Communications Wireless Communications Series, IEEE Transactions on Wireless Communications and IEEE Transactions on Vehicular Technology, and is serving on the editorial boards of the IEEE Transactions on Computers, IEEE Wireless Communications Letters, Journal of Communications and Networks, Computer Communications, as well as several other journals. He has guestedited many journal special issues, and served on the technical program committees of numerous international conferences. He has contributed to the organization of many conferences and workshops.



Long Hu received the B.S. degree in Huazhong University of Science and Technology. He is a member in Embedded and Pervasive Computing Laboratory, Huazhong University of Science since 2012. His research includes Internet of Things, Machine to Machine Communications, Body Area Networks, Body Sensor Networks, E-healthcare, Mobile Cloud Computing, Cloud-Assisted Mobile Computing, Ubiquitous Network and Services, Mobile Agent, and Multimedia Transmission over Wireless Network.