Quality-of-Experience Optimization for a Cloud Gaming System With *Ad Hoc* Cloudlet Assistance

Wei Cai, *Student Member, IEEE*, Zhen Hong, *Student Member, IEEE*, Xiaofei Wang, *Member, IEEE*, Henry C. B. Chan, *Member, IEEE*, and Victor C. M. Leung, *Fellow, IEEE*

***Abstract*— Cloud gaming systems host the game in the cloud, while Gameplays and views are streamed to the players' terminals in the form of encoded video frames. To address the high-bandwidth issue of real-time gaming video transmission, we have proposed a cloudlet-assisted multiplayer cloud gaming system to encourage cooperative video sharing, which exploits the similarities of video frames among multiple players in the same crowd playing the same game via a secondary *ad hoc* network. In this paper, we provide a detailed modeling of the proposed system, including the correlation between video frames, mobility of terminal devices, and diversity of network quality of service for distinct players. With necessary mathematical formulations, we study the players' behaviors regarding the cooperative sharing patterns to optimize the system performance in terms of the quality of users' experience. Also, heuristic algorithms are proposed to reduce the computational complexity. Empirical study and trace-driven simulation results illustrate the impact of mobility on the system performance and show that the proposed solution is able to provide better quality of experience compared with the existing platform.**

***Index Terms*— Cloud gaming, cooperative, encoding, optimization, quality of experience (QoE).**

I. Introduction

DRIVEN by strong volume in sales of mobile gaming, video game consoles, and software, the size of the video games market in 2014 was estimated to be $81.5 billion,[1] more than twice the revenue of the international film industry in 2013. This arresting profit has attracted great attention from researchers, system designers, and application developers in the cloud computing area to design and implement cloud-based video gaming systems [1]. With the novel concept of providing everything as a service [2], the transformation of traditional gaming software into gaming as a service (GaaS) [3] has become an emerging topic.

In contrary to conventional video game business, the cloud gaming model exhibits several unique advantages.

1) *Scalability:* It overcomes the hardware constraints of gaming terminals, including processing capacity, data storage, and battery in mobile devices.
2) *Cost Effectiveness:* It reduces the production cost with a unified development approach.
3) *Ubiquitous and Multiple-Platform Support:* It provides a seamless gaming experience across diverse platforms.
4) *Effective Antipiracy Solution:* It transforms the game developing companies into game service providers with a potential solution to the troublesome software piracy problems.
5) *Click and Play:* It supports a play-as-you-go mode, in which the players can start their gaming sessions without downloading and setting up the complete game copies.
6) *Energy Efficiency:* It has strong potentials in bringing longer battery life for the mobile terminals and longer gaming times for the players by offloading game programs' high computational complexity to the cloud. Therefore, not only academia but also industry is interested in the development of cloud gaming solutions. Pioneering cloud gaming companies such as OnLive,[2] Gaikai,[3] and G-Cluster[4] have started to provide commercialized GaaS to the public.

In the gaming-on-demand model, a remote rendering architecture is employed, whereby the cloud gaming service providers host their video games in cloud servers and stream the gaming video frames to the players' terminals over the Internet. In reverse, the interactions from game players are transmitted to the cloud servers over the same networks [4]. In this context, the cloud intrinsically becomes an interactive video generator and streaming server, while the users' terminals function as the event controllers and video displays. With this approach, the cloud gaming service enables the players to run sophisticated games despite the restricted hardware capacity of the terminals, at the expenses of higher costs and energy consumption in communications to access the Internet.

Manuscript received September 21, 2014; revised February 5, 2015 and April 11, 2015; accepted June 11, 2015. Date of publication June 26, 2015; date of current version December 3, 2015. This work was supported in part by the University of British Columbia under a Four Year Doctoral Fellowship and a Work Learn International Undergraduate Research Award, and grants from the Natural Sciences and Engineering Research Council of Canada (STPGP 447524) and the National Natural Science Foundation of China (61271182). This paper was recommended by Associate Editor Y. Lu.

W. Cai, Z. Hong X. Wang, and V. C. M. Leung are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: weicai@ece.ubc.ca; hongz.ubc@gmail.com; xfwang@ece.ubc.ca; vleung@ece.ubc.ca).

H. C. B. Chan is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: cshchan@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2450153

[1] http://www.newzoo.com/insights/asia-pacific-contributes-82-6bn-global-games-market-growth/

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

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

[4] http://www.g-cluster.com

It is obvious that video frame transmissions via the Internet can consume a huge amount of network resources, which can lead to long delays in game responses [5], [6]. Even though there have been plenty of efforts devoted to the optimization of the cloud gaming systems [7]–[10], due to the constraints imposed by existing network infrastructure and mobile networks' charging policies, gaming-on-demand has yet to reach its promised potential.

To solve this critical issue, we have proposed a cloudlet-assisted cloud gaming system [11] to exploit the correlations among peer players' gaming video. Inspired by the idea of peer-to-peer sharing between multiple players in the same game, we investigated a multiplayer cloud gaming system with cooperative video sharing. Mobile devices are connected to the cloud server for real-time interactive game videos while sharing the received video frames with their peers via a secondary *ad hoc* network. As the first approach in cooperative sharing for cloud gaming scenarios, this paper aimed to substantially reduce the transmission rate from cloud server to the game clients in order to overcome the bottleneck of Internet access. It has investigated the video correlation patterns and provided a modeling of gaming and player interactions. As suggested by the experimental results, the expected overall server transmission rate for a multiplayer action role-playing game is significantly reduced by up to 64% with an optimal encoder and 54% with a more practical encoding solution that restricted the frame prediction to one single hop.

However, the framework proposed in [11] still lacks consideration from the following aspects.

1) *Mobility Issue:* The proposed system assumes an ideal case in which the network bandwidth within the *ad hoc* cloudlet is unlimited; this is not realistic as wireless communications constrain the sharing of video frames only among mobile devices that are within a certain distance of each other, which form a time-variable group due to the mobility of terminals.
2) *Network Diversity Issue:* The previous work assumes that all the mobile devices access the cloud through a common network with the same quality of service (QoS); however, the networks between individual terminals and the cloud may have different QoSs due to differences in network traffic levels and channel conditions experienced by the terminals. Variations of network QoS (NQoS) will strongly affect the overall system performance.
3) *User Quality of Experience (QoE):* The proposed system focuses only on the optimization of server transmission rate without considering the effect of NQoS. In fact, optimizing encoding to minimize server transmissions may result in poor QoE as devices receiving larger frames from the cloud while experiencing a low network bandwidth might degrade gaming video decoding at all the players' terminals.

In this paper, we investigate a QoE-oriented multiplayer cloud gaming system with *ad hoc* cloudlet assistance. The contribution of this paper is summarized as follows.

1) *System Modeling and Formulation:* We model and formulate the proposed cloud system, including the avatar behavior, terminal's mobility, the diversity of NQoS, video frame correlations, and frame encoding structures. We also provide a validation of the proposed encoding solution.
2) *QoE Optimization:* We define a QoE factor to evaluate the system performance and hence propose an optimization method based on it. We also investigate heuristic solutions to get suboptimal encoding methods with lower computational complexity.
3) *Gaming Video Frame Measurements:* We capture videos of gaming sessions to measure the frame sizes and fit them to mathematical models for trace-driven simulations.

The reminder of this paper is organized as follows. We review related work in Section II and provide a system overview with modeling in Section III. Then, we formulate the system and its QoE optimization problem in Sections IV and V, respectively. We propose two heuristic algorithms to obtain suboptimal solutions with lower computational complexity in Section VI. Empirical experiments and trace-driven simulations, respectively, demonstrate the superior system performance compared with existing systems in Sections VII and VIII. Section IX concludes this paper.

II. Related Work

A. Video Optimization in Cloud Gaming

To meet constantly changing network communication and cloud computation constraints, in [7], a selective object encoding method is proposed to reduce the required network bandwidth and processing power without much impact on user perceived QoE. The key idea is to add fewer objects to the scene so that the game processing and image generating time would be reduced. Another prior work [8] introduces a video encoder that selects a set of key frames in the video sequence and uses the 3-D image warping algorithm to interpolate other nonkey frames. This approach takes advantage of the pixel depth, rendering viewpoints, camera motion patterns, and even the auxiliary frames that do not actually exist in the video sequence to assist video coding.

On the other hand, plenty of works have been proposed to address the contradiction between varying network environments and high-bandwidth requirements of real-time video transmissions. Reference [9] designs a system that monitors the round-trip time jitter as network end-to-end status and alters the video encoding parameters accordingly. Another representative work [12] considers all the objective and subjective QoS factors that affect players' QoE and formulates an impairment function to model the mobile game user experience (MGUE). A set of MGUE-oriented optimization techniques are proposed in [10] to address the challenges of meeting response time targets of gaming sessions over wireless networks. Rendering parameters, such as realistic effect, view distance, texture details, environment details, and rendering frame rate are studied, and their effects on communication and computation costs are characterized. Then, an adaptive rending technique is proposed to dynamically vary the graphic rendering parameters in the cloud servers in response

to the constantly changing communication and computation constraints. The adaption process includes both offline steps to derive optimal rendering settings for different adaption levels (determined by the communication and computation costs) and online steps implementing a run-time adaptation scheme that can select the optimal adaptation level depending on the current network and computation environment.

B. Ad Hoc Cloudlet

An *ad hoc* cloudlet is defined as a cooperative cloudlet formed by a group of terminals via multihomed *ad hoc* network connections [13]–[15]. We stress that our assumption of devices being connected to multiple networks simultaneously, such as wireless wide area network (WWAN) to gaming cloud and *ad hoc* cloudlet to neighbor peers, is a common one in [16]–[18] and realizable in practice (e.g., with smart phones), where different optimizations are performed exploiting the multihoming property. It is shown in [16] that aggregation of an *ad hoc* group's WWAN bandwidths can speed up individual peers' infrequent, but bursty content downloads just like Web access. An integrated cellular and *ad hoc* multicast architecture is proposed in [17], where the cellular base station delivers packets to proxy devices with good channel conditions, and then the proxy devices utilize local ad hoc wireless local area network (WLAN) to relay packets to other devices. Recently, [18] utilizes a secondary *ad hoc* WLAN network for local recovery of WWAN broadcast/multicast packets lost during WWAN transmissions by exploiting cooperation among peers. Our proposal extends this body of work on cooperative multihomed networks to interactive light field streaming by exploiting correlation between requested images and content residing in peers' caches to lower server transmission rate.

C. Correlations of Videos Frames

An inter frame, e.g., P-frame, is a frame in a video compression stream, which is expressed in terms of one or more neighbor frames. In contrary to an intra-frame (I-frame) coding that performs compression relative to information contained only within the current frame, the inter part of the term refers to the use of inter-frame prediction. Its size, which affects the system performance, is subject to the correlations between the encoded video frames. Light field [19] and multiview [20] 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 2-D array of closely spaced cameras. The correlations of light field images are studied and formulated in [21], which indicates that the correlation between two different views to a static scene is related to the geographical distances between each other. Interactive multiview video switching [22] designs a preencoded 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. However, compared with light field and multiview switching, the modeling of correlations between players' views is more complicated. There are infinite numbers of views as the players are adjusting their personal views while they are walking through the scene and participating in a battle field. The dynamic switching makes the correlation model hard to predict.

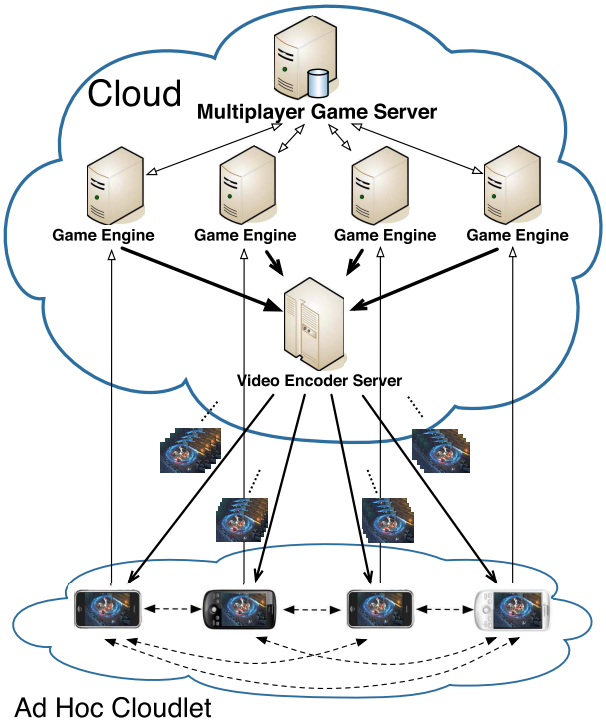

Fig. 1. *Ad hoc* cloudlet-assisted multiplayer cloud gaming system.

D. Real-Time Video Encoding

Unlike light field and multiview switching, video encoding for cloud gaming is essentially a real-time process. The cloud encodes video frames immediately after the game scenes are rendered. The fundamental idea of encoding is very simple: it starts with an intra-coded frame, e.g., I-frame, and then followed by a certain number of inter frames, such as P-frames [23], distributed source coding (DSC) frames [24], and so on. Therefore, in order to achieve the desired tradeoff between bit rate and error rate, how to determine the sequence of various types of frames has become one of the most critical problems in video encoding. In recent video encoding research, the length of group of pictures (GOP) [25] is set to be adaptive, which implies a structure with one I-frame and variable number of inter frames. In this paper, to simplify the estimation of server transmission rate, we assume that the GOP length is infinite. Thus, the encoded video stream consists of a sequence of P-frames, after the first I-frame transmission.

III. System Overview

A. Architecture

The architectural framework of the proposed cloudlet-assisted multiplayer cloud gaming system is shown in Fig. 1. Similar to the existing cloud gaming work, instances of *game engine* are hosted in the cloud to provide gaming services to players. They are connected to a *multiplayer game server* in conventional fashion to facilitate interactions between avatars.

The novelty of the proposed system is to introduce two additional components. First, video encoder server is acting as a gateway, which exploits the correlations between video frames for different players to perform centralized encoding with the purpose of minimizing server transmission rate. In this paper, we consider cloud as an infinite resource provider. Therefore, the computational power of the encoder server is unlimited. Second, *ad hoc* cloudlet is a cooperative *ad hoc* cloudlet constructed by the 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. Similar to [26]–[28], we assume that the network bandwidth within the *ad hoc* cloudlet is sufficiently large for all the mobile devices in the immediate neighborhood to share their frames when needed. Thus, the bandwidth constraint inside the cloudlet will not be explicitly modeled.

IV. System Modeling and Formulation

In this section, we model the *ad hoc* cloudlet-assisted cloud gaming system and formulate the reputation-based multiplayer fairness problem.

A. Avatar Behavior Model

We choose to model the avatar behavior in a third person game, since this is the most classic and popular model, as summarized in [29]. We define the gaming map as a 2-D map with an $m \times m$ screen. On this particular map, we set all the avatars' initial positions to the center of the map and model the walking strategies of n players' avatars as *random walk* and *group chase* with probabilities p_{rw} and p_{gc}, respectively. For *random walk* movements, an avatar either holds its position with probability p_h or moves in its adjacent n_{adj} directions with identical possibilities p_c. To simplify the model, we set $p_h = p_c = (p_{\text{rw}}/(n_{\text{adj}} + 1))$. For *group chase* movements, an avatar randomly selects another avatar in the scene and moves toward it for a certain period of time t_{chase}. Let the probability of group chase movement be $p_{\text{gc}} = 1 - p_{\text{rw}}$, then the probabilities of any other $n-1$ target avatars to be chased will all equal to $p_{\text{appr}} = (p_{\text{gc}}/(n-1))$. Note that we set the moving unit for each avatar in a unit time to be k pixels and denote w and h as the width and height of the gaming screen in pixels. Given that the system restricts the avatar to be in the center of the screen, all the avatars' position coordinates will be restricted to (x, y), where $x \in [0, (((m-1)w)/k)]$ and $y \in [0, (((m-1)h)/k)]$.

With this model, we formulate the locations of all n avatars into two $1 \times n$ vectors X and Y, in which (X_i, Y_i) indicates the ith avatar's coordinate. With this approach, all the avatar behavior models can be represented by the changes of vectors X and Y.

B. Encoding Structure

Fig. 2 shows an example of frame dependency in a four-player scenario. The squares represent video frames in video streaming, in which the paired number (t, i) denotes a frame according to time slot t and player i. The arrows

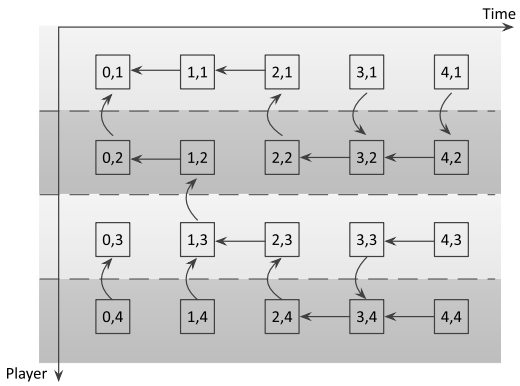

Fig. 2. Frames correlation in real-time multiplayer game videos.

represent the frame encoding dependencies: as depicted, the video frames are able to be encoded in two dimensions. *Intra-stream P-frames* are those predicted by their previous frames in a same video stream, while the *inter-stream P-frames* are those predicted by peer game videos' frames. For instance, $P_{\text{intra}}[(1,2) \rightarrow (0,2)]$ indicates that player 2's second intra-stream P-frames are predicted from player 2's first decoded image, while $P_{\text{inter}}[(1,3) \rightarrow (1,2)]$ indicates that player 3's second inter-stream P-frames are predicted from player 2's second decoded image.

The size P_a of an *intra-stream P-frame* is subject to the variance of the game video content. In contrast, the size P_e of an *inter-stream P-frame* is subject to the correlation between two video streams for corresponding peering game players i and j with coordinates of (X_i, Y_i) and (X_j, Y_j), which is formulated as a function $P_e(X_i, Y_i, X_j, Y_j)$. Note that this function ensures that all the avatar behavior models can be reflected in the *inter-stream P-frame*'s size. With P_a and P_e, we define the video frame correlation matrix P. Thus, an element $P_{ij}, i \neq j$ stores the frame size of an *inter-stream P-frame* to decode the ith player's video frame by predicting the jth player's. In contrast, P_{kk} saves the frame size of an *intra-stream P-frame*, which enables the kth player to decode his/her current video by predicting the proceeding frame of his/her own video.

C. Terminal Mobility Model

To demonstrate the mobility of user terminals, we set up a square area of $r \times r\ m^2$ in which n game players are confined. The ith player's physical position is represented as (u_i, v_i), where $i = 1, 2, \ldots, n$ and $u_i \in [0, r]$ and $v_i \in [0, r]$ are the X and Y coordinates of the location. In a typical gaming engagement, players are initially gathered in the center of the area, and then randomly move within s meters in both directions every 30 s, i.e., $\Delta u_i \in [-s, s]$ and $\Delta v_i \in [-s, s]$, where Δu_i and Δv_i represent the movements of the ith player in the X and Y directions in every 30 s. The maximum inter-device communication distance, or the device communication range, is c meters.

Based on this model, we formulate the relationship between n players' terminals as an $n \times n$ matrix E. The numeric value of an element in matrix E is defined to be either 0 or 1,

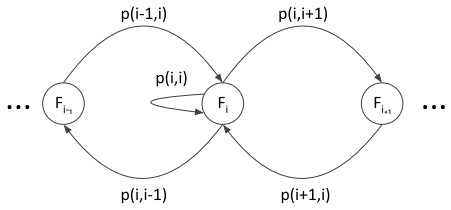

Fig. 3. Markov process for the change of NQoS level.

where $E_{ij} = 1$ represents that the ith player's terminal is within the communication range of jth player's terminal. The matrix E is derived by

$$E_{ij} = \begin{cases} 1, & D_{ij} \leq c \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where

$$D_{ij} = \sqrt{(u_i - u_j)^2 + (v_i - v_j)^2} \tag{2}$$

is the distance between the ith and jth players. Using matrix E to represent the connectivity reveals the possible problem of link quality varying by dramatic changes in *ad hoc* network topologies. For example, in the encoding phase, the server uses E to design an encoding solution that player 1 decodes its frame by predicting player 2's frame. However, the connectivity between players 1 and 2 is lost in the decoding phase, as the matrix E has been altered according to the terminals' mobility. In this case, player 1's frame is not decodable. While this problem can occur, its probability can be kept sufficient low by explicitly controlling the latency between encoding and decoding of video frames to be within some acceptable levels (e.g., 200 ms is maximal tolerable and 120 ms is hardly noticeable, as stated in [30]). In general, the mobility of terminals will not cause dramatic network topology change in such a short time interval. Therefore, this problem is negligible.

D. Network Quality-of-Service Model

In practical scenarios, the players may access the cloud over different WWANs and experience different NQoSs, according to the network traffic load, wireless signal strength, channel propagation condition, and so on. According to [31], the players' QoEs are impacted by several NQoS factors, such as network bandwidth ξ_b, network latency ξ_l, network delay variance ξ_v, and network loss rate ξ_r. These factors can be formulated into an abstract and unified *NQoS level* η as a function

$$\eta = \mathcal{K}(\xi_b, \xi_l, \xi_v, \xi_r). \tag{3}$$

We can always find a function $\mathcal{K}$ that converts the NQoS parameters to an integer value η within the range $[1, n_q]$, where n_q represents the best NQoS level. According to the mobile devices' mobility and the wireless signal coverage, a mobile terminal's *NQoS level* is expected to change from one level to another over time.

In this paper, we model the variations of *NQoS level* as a Markov process in which η transits from level a to b with probability $p(a, b)$, as shown in Fig. 3.

According to this model, each terminal device is associated with one *NQoS level* η, which varies from time to time. Thus, a $1 \times n$ vector U is sufficient to represent the NQoS for n players. However, to simplify our formulation and optimization in the following sections, we formulate the NQoS diversity as an $n \times n$ matrix N, which is an expansion of U by $N = [U, U, \ldots, U]$. Note that, $\forall x, i \in [1, \ldots, n]$, $N_{ix} = U_i$ represents the NQoS level η of the ith player.

E. Video Encoding Solution

For the proposed system, the most critical issue is to find an optimal video encoding solution that fulfills the requirements. In this paper, we represent a video encoding solution by an $n \times n$ matrix M, where n represents the number of players. Each element in matrix M can take a numeric value of either 0 or 1, where $M_{ij} = 1$ represents that the ith player's video frame is decoded by predicting to jth player's. Thus, an element $M_{kk} = 1$ implies that the kth player's terminal will download an *intra-stream P-frame* from the cloud, while $M_{ij} = 1, i \neq j$, represents that the ith player's terminal is able to exploit the correlations from the jth player's decoded video frame, and hence its video frame is decoded by the combination of *inter-stream P-frame* downloaded from cloud and a decoded DSC video frame *DSC frame* received from the jth player's terminal via the *ad hoc* network.

To guarantee that the solution represented by a particular M can be adopted as a system encoding solution, a validation on the solution is required.

1) Information Integrity: In the distributed decoding system facilitated by the *ad hoc* cloudlet, obviously at least one player needs to download an *intra-stream P-frame* from the cloud, in order to provide a future reference for the peer terminals. Thus, the first condition is that at least one of the diagonal elements of M is 1

$$\text{tr}(M) = \sum_{i=1}^{n} M_{ii} \geq 1 \tag{4}$$

where $\text{tr}(M)$ denotes the trace of square matrix M.

2) Decoding Reliability: All the players' terminals require either *intra-stream P-frame* or *inter-stream P-frame* to decode their video frames. Hence, a second condition is that the sum of elements in each row of M is equal to 1, as

$$\forall i \in M_{ij}, \quad \sum_{j=1}^{n} M_{ij} = 1. \tag{5}$$

3) Loop Free: With the above checks, a solution is still not guaranteed to be valid. A problem called *decoding loop* can cause invalid solutions in the system. To solve this problem, we consider nonzero elements in M as the decoding vectors for n players and further represent these vectors by a graph $G = (V, E)$, where finite sets V and E contain the nodes and directed edges, respectively. Some notations are defined in Table I for the analysis.

We here make the statement that, if a graph G contains loop, its corresponding encoding solution M is not able to be decoded in the *ad hoc* cloudlet. The proof is as follows.

Lemma 1: No *intra node* or valid path is on a loop.

TABLE I
SOLUTION GRAPH NOTATIONS

Node	Each player is represented as either an *inter-node* γ_x or an *intra-node* λ_x. If a player will download images from the server via *Intra-stream P-frame*, the player is represented as an *intra-node*; otherwise, the player is represented as an *inter-node*. The set of all nodes $V = \{\gamma_1, \gamma_2, ..., \gamma_m, \lambda_1, \lambda_2, ..., \lambda_n\}$ contains m *inter-nodes* and n *intra-nodes*.
Edge	Each edge connects two nodes and has a direction. An edge pointing from γ_1 to γ_2 is represented as (γ_1, γ_2). If players 1 and 2 are represented as *inter-nodes* γ_1 and γ_2, respectively, then (γ_1, γ_2) indicates that player 1 will download images from player 2 via an *Inter-stream P-frame*.
Path	A finite sequence of edges connecting distinct nodes forms a path. For example, $P_x = (\gamma_1, \gamma_2, ..., \lambda_n)$ is a path with a starting point γ_1 and an end point λ_n. An isolated *intra-node* (no edge pointing in) is on a zero-path $P_y = (\lambda_n)$.
Loop	A path combined with an edge pointing from the end point of the path to the starting point of the path is called a loop. For $P_x = (\gamma_1, \gamma_2, ..., \gamma_n)$, $C_y = P_x + (\gamma_n, \gamma_1)$ is a loop with n nodes $\{\gamma_1, \gamma_2, ..., \gamma_n\}$.
Valid Path	A path ending at an *intra-node* is called a valid path. A zero-path is also a valid path.
Decoding Dependance	An *intra-node* decodes its image by default (by downloading directly from the server, which is not shown in our analysis.) An *inter-node* decodes its image only if it is on a valid path.
Valid System	If each node in the system can decode its image, then this system is a valid system.

Proof: Since any *intra node* does not point to any other node, it must not be in a loop. For the same reason, since any valid path ends at an *intra node* that is not in a loop, this path must not be in a loop. ∎

Theorem 1: If a system is valid, then no loop is present in the system.

Proof: Since the system is valid, each *inter node* must be on a valid path. By Lemma 1, no *intra node* or *inter node* is in a loop, so no loop is present in the system. ∎

Theorem 2: If no loop is present in a system, the system is valid.

Proof: Suppose to the contrary that a system with no loop is not valid. Then, there exists an *inter node* γ_x in the system such that by choosing γ_x to be the starting point, we can always find a longest path $P_x = (\gamma_x, \gamma_{x+1}, \ldots, \gamma_{x+n})$ with n edges. Since P_x is the longest path [i.e., γ_{x+n} cannot point to an $(n+2)$th node] and *inter node* γ_{x+n} has an edge pointing out, there must exist an edge $(\gamma_{x+n}, \gamma_{x+k})$ such that γ_{x+n} must point to γ_{x+k}, where $k = 0, 1, 2, \ldots, n-1$. Then, $(\gamma_{x+k}, \gamma_{x+k+1}, \ldots, \gamma_{x+n}) + (\gamma_{x+n}, \gamma_{x+k})$ form a loop, which is contradictory to our assumption. Therefore, the theorem has been proven. ∎

Therefore, a system is valid if and only if no loop is present in a system.

In order to provide a loop-free solution, it is mandatory that M satisfies

$$\forall p \in \{1, 2, \ldots, n\}, \quad \mathrm{tr}(M^p) = \mathrm{tr}(M). \tag{6}$$

Hereby, we prove the above statement as follows.

Proof: In the $n \times n$ solution matrix M, we define: *1) two-node loop:* $\exists\ i, j \in \{1, 2, \ldots, n\}$ *and* $i \neq j$, such that $m_{ij} \cdot m_{ji} = 1$; *2) three-node loop:* $\exists\ i, j, k \in \{1, 2, \ldots, n\}$ *and* $i \neq j, j \neq k, k \neq i, such that$ $m_{ij} \cdot m_{jk} \cdot m_{ki} = 1$; and *3) similarly for higher order loops*. Denote $Q = M^2$, such that

$$q_{ij} = \sum_{h=1}^{n} m_{ih} \cdot m_{hj}. \tag{7}$$

Note that, since $m_{ij} \in \{0, 1\}$, $m_{ij}^k = m_{ij}$ for $k = 1, 2, \ldots, n$, then we derive

$$\begin{aligned} q_{ii} &= \sum_{h=1}^{n} m_{ih} \cdot m_{hi} \ = m_{ii}^2 + \sum_{\substack{h=1 \\ h \neq i}}^{n} m_{ih} \cdot m_{hi} \\ &= m_{ii} + \sum_{\substack{h=1 \\ h \neq i}}^{n} m_{ih} \cdot m_{hi} \end{aligned} \tag{8}$$

$$\begin{aligned} \mathrm{tr}(Q) &= \sum_{i=1}^{n} q_{ii} \ = \sum_{i=1}^{n} \left(m_{ii} + \sum_{\substack{h=1 \\ h \neq i}}^{n} m_{ih} \cdot m_{hi} \right) \\ &= \sum_{i=1}^{n} m_{ii} + \sum_{i=1}^{n} \left(\sum_{\substack{h=1 \\ h \neq i}}^{n} m_{ih} \cdot m_{hi} \right) \\ &= \mathrm{tr}(M) + \sum_{i=1}^{n} \left(\sum_{\substack{h=1 \\ h \neq i}}^{n} m_{ih} \cdot m_{hi} \right). \end{aligned} \tag{9}$$

Since $m_{ij} \in \{0, 1\}$, if there is no *two-node loop*, then

$$m_{ij} \cdot m_{ji} = 0 \quad \forall i, j \in \{1, 2, \ldots, n\} \text{ and } i \neq j. \tag{10}$$

Thus

$$\sum_{i=1}^{n} \left(\sum_{\substack{h=1 \\ h \neq i}}^{n} m_{ih} \cdot m_{hi} \right) = 0 \tag{11}$$

therefore

$$\mathrm{tr}(Q) = \mathrm{tr}(M^2) = \mathrm{tr}(M). \tag{12}$$

Now, let $X = M^3 = Q \cdot M$, if there is no two-node loop, we derive

$$\begin{aligned} x_{ii} &= \sum_{h=1}^{n} q_{ih} \cdot m_{hi} \ = m_{ii} \cdot q_{ii} + \sum_{\substack{h=1 \\ h \neq i}}^{n} q_{ih} \cdot m_{hi} \\ &= m_{ii}^3 + \sum_{\substack{h=1 \\ h \neq i}}^{n} q_{ih} \cdot m_{hi} \ = m_{ii} + \sum_{\substack{h=1 \\ h \neq i}}^{n} q_{ih} \cdot m_{hi} \\ &= m_{ii} + \sum_{\substack{h=1 \\ h \neq i}}^{n} \left(\sum_{\substack{v=1 \\ v \neq i,h}}^{n} m_{iv} \cdot m_{vh} \cdot m_{hi} \right). \end{aligned} \tag{13}$$

Thus

$$\begin{aligned}\mathrm{tr}(X) &= \sum_{i=1}^{n} x_{ii} = \sum_{i=1}^{n}\left(m_{ii} + \sum_{\substack{h=1\\h\neq i}}^{n}\sum_{\substack{v=1\\v\neq i,h}}^{n} m_{iv}\cdot m_{vh}\cdot m_{hi}\right)\\ &= \sum_{i=1}^{n} m_{ii} + \sum_{i=1}^{n}\sum_{\substack{h=1\\h\neq i}}^{n}\sum_{\substack{v=1\\v\neq i,h}}^{n} m_{iv}\cdot m_{vh}\cdot m_{hi}\\ &= \mathrm{tr}(M) + \sum_{i=1}^{n}\sum_{\substack{h=1\\h\neq i}}^{n}\sum_{\substack{v=1\\v\neq i,h}}^{n} m_{iv}\cdot m_{vh}\cdot m_{hi}. \end{aligned} \tag{14}$$

Since $m_{ij} \in \{0, 1\}$, if there is no *three-node loop*

$$\forall i, v, h \in \{1, 2, \ldots, n\}, \quad i \neq v, \quad v \neq h, \quad h \neq i$$
$$m_{iv}\cdot m_{vh}\cdot m_{hi} = 0 \tag{15}$$

thus

$$\sum_{i=1}^{n}\sum_{\substack{h=1\\h\neq i}}^{n}\sum_{\substack{v=1\\v\neq i,h}}^{n} m_{iv}\cdot m_{vh}\cdot m_{hi} = 0. \tag{16}$$

Therefore

$$\mathrm{tr}(X) = \mathrm{tr}(M^3) = \mathrm{tr}(M). \tag{17}$$

Using the same approach for higher order loops, we conclude that, to guarantee loop-free solution, the following equation shall be satisfied:

$$\forall p \in \{1, 2, \ldots, n\}, \quad \mathrm{tr}(M^p) = \mathrm{tr}(M). \tag{18}$$

■

4) Mobility Constraints: The decoding procedure is constrained by the wireless connectivity in the *ad hoc* network, which is specified by the neighboring matrix E in our system definition.

However, in an *ad hoc* network, the mobile terminals may also communicate with each other over multiple hops. Denote h as the maximum number of hops allowed, we derive the h-hop neighboring matrix $\widehat{E}$ as

$$\widehat{E_{ij}} = \mathrm{pos}(\widetilde{E_{ij}}) = \begin{cases} 1, & \widetilde{E_{ij}} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where

$$\widetilde{E} = E^h. \tag{20}$$

Hence, we hereby formulate the mobility constraints into the validation of solution M, as shown in the following equation:

$$M = M \odot \widehat{E}. \tag{21}$$

5) Multihop Decoding Constraints: 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 realizing the system: since the encoder efficiently groups correlated video frames and constructs a tree to describe the dependency of the video frames, a *multihop decoding* might occur at the users' terminals. For example, in time slot t, given an *intra-stream P-frame* $P_{\text{intra}}[(t, 1) \rightarrow (t-1, 1)]$ for player 1 and two *inter-stream P-frames* $P_{\text{inter}}[(t, 2) \rightarrow (t, 1)]$ and $P_{\text{inter}}[(t, 3) \rightarrow (t, 2)]$ for players 2 and 3, player 3 can only decode the video frame after player 2 has decoded the frame by receiving the decoded image from player 1, which introduces larger system latency. However, gaming applications are very sensitive to latency. Therefore, *multihop decoding* might affect the players' QoE.

To address this issue, we provide a solution that restricts the dependency of *inter-stream P-frame* decoding to one hop to guarantee an acceptable level of decoding latency. This *one-hop inter-stream encoding* is enforced by introducing constraints on solution M.

Define n vector V by

$$V_j = \mathrm{pos}\left(\sum_{i=1}^{n} M_{ij}\right) = \begin{cases} 1, & \sum_{i=1}^{n} M_{ij} > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

A solution M satisfies *one-hop inter-stream encoding* if and only if

$$\forall i \in M_{ii}, \quad M_{ii} = V_i. \tag{23}$$

V. Optimization Target

A. QoE Factor

A main focus of cloud gaming system optimization is to minimize the expected server transmission rate from the cloud. Apparently, exploiting similarities from peer players' video frames with *inter-stream P-frame* of smaller packet size is generally more advantageous in terms of data transmission speed, power consumption, data expense, and others. However, the server transmission rate is not the only factor that impacts the system performance. Regarding QoE control for all the players participating in this cooperative *ad hoc* cloudlet, there are more variables involves such as NQoS, device computing capacity, and so on. For instance, with a specific encoding matrix M that minimizes transmissions, player i may be selected as the reference to which other peering terminals depend on for frame decoding, while it is struggling with poor NQoS or low computing capacities. As a consequence, its delay on video image construction introduces additional latency to the whole *ad hoc* cloudlet system.

To this end, we introduce a novel measure *QoE factor* θ as an index of system performance, where a smaller value of *QoE factor* θ represents a better QoE. Thanks to the recent development on hardware, most terminals have the computing power to perform complicated computations nowadays. Therefore, the *QoE factor* only considers the two most critical parameters: 1) server transmission load and 2) the user's NQoS. The server transmission load R is represented by an $n \times n$ matrix, where n is the number of players, with elements R_{ij} given by

$$R_{ij} = M_{ij} \cdot P_{ij} \tag{24}$$

where R_{ij} represents that the server transmission data size when ith player's video frame is decoded by predicting to jth player's. We expect the value of the *QoE factor* θ to be proportional to the expected server transmission load R and inversely proportional to the NQoS level N. Hence, we may consider R and N as the *demand* and

supply [32] of the proposed system, respectively. Referring to [32], we define

$$\theta = \mathcal{F}(R, N) \tag{25}$$

where R and N can be negatively correlated. In this paper, without loss of generality, we choose the inverse proportional relationship between them. For instance

$$\theta = \sum_{i=1}^{n}\sum_{j=1}^{n}(R_{ij}/N_{ij}). \tag{26}$$

However, it is possible to adapt the above model to various cases of the proposed system.

B. Objective Function

We can now formally define the search for the optimal encoding structure as an optimization that finds video encoding solution M, using the calculation of video correlation matrix P and NQoS N, in the feasible space Φ that possesses the smallest possible expected *QoE factor* θ, while all the constraints on the encoding solution are observed. We formulate this optimization problem using the following objective function:

$$\begin{aligned} &\text{min.} \quad O(M, P, N) = \sum_{i=1}^{n}\sum_{j=1}^{n}(M_{ij} \cdot P_{ij}/N_{ij}) \\ &\text{s.t.} \quad (4)(5)(6)(21)(23). \end{aligned} \tag{27}$$

Note that (23) represents the restriction of multihop decoding, which is optional in optimizations for different purposes.

VI. Heuristic Approach

Given a cloud gaming session with n participants, deriving the optimal encoding solution for all terminals incurs a complexity of 2^n, which indicates a high computational cost that exponentially increases with the number of terminals accessing the cloud for gaming services. To address this issue, we investigate two heuristic approaches to support quick QoE optimization with lower computational complexity.

A. Local Greedy Approach

A local greedy approach reaches a suboptimal encoding solution by always looking for the smallest *QoE factor* in the solution space, as shown in Algorithm 1. It follows the principle of a pruning algorithm to achieve the local optima.

Note that if there exist multiple smallest S_{xy}, the algorithm will further consider their potential usage as others' references to derive a relatively better encoding method.

B. One-Hop Restricted Local Greedy Approach

To solve the multihop decoding problem, another one-hop encoding algorithm as shown in Algorithm 2 is proposed in this section. The solution is basically the same as the local greedy approach, but it does not consider multihop dependency in encoding structures.

According to Algorithm 2, with the same value of S_{xy}, we set up a higher priority to *intra-stream P-frame* rather than *inter-stream P-frame*, since an *inter-stream P-frame* will never be a future reference for other players with one-hop decoding restriction.

Algorithm 1 Local Greedy Encoding Algorithm

Given correlation matrix P and NQoS matrix N
Initiate all-0 video encoding solution matrix M
Initiate search space matrix $S = P/N$
repeat
  for each $M_{mn}! = 0$ **do**
    Search smallest S_{xy} in S_{ii} and S_{im}
    if exist multiple smallest S_{xy} **then**
      Search smallest S_{ix} for each S_{xy}
      Select S_{xy} with smallest S_{ix}
    end if
    Set $1 \rightarrow M_{xy}$
    Set $null \rightarrow S_{xi}$
  end for
until $\forall i \in M_{ij}, \sum_{j=1}^{n} M_{ij} = 1$
return M

Algorithm 2 One-Hop Local Greedy Encoding Algorithm

Given correlation matrix P and NQoS matrix N
Initiate all-0 video encoding solution matrix M
Initiate search space matrix $S = P/N$
repeat
  for each $M_{mn}! = 0$ and $m = n$ **do**
    Search smallest S_{xy} in S_{ii} and S_{im}
    if exist multiple smallest S_{xy} and $x = y$ **then**
      Search smallest S_{ix} for each S_{xy}
      Select S_{xy} with smallest S_{ix}
    else
      if exist one smallest S_{xy} and $x = y$ **then**
        Select S_{xy} with smallest S_{ix}
      end if
    else
      if exist multiple smallest S_{xy} and $x! = y$ **then**
        Randomly select a S_{xy}
      end if
    end if
    Set $1 \rightarrow M_{xy}$
    Set $null \rightarrow S_{xi}$
  end for
until $\forall i \in M_{ij}, \sum_{j=1}^{n} M_{ij} = 1$
return M

VII. Experimental Case Study: Diablo 2

We first conduct a case study on Diablo 2, one of the most popular classic multiplayer action role-playing games in history. To simplify our model, two players (as a sorceress and an assassin) are connected to each other over a WLAN and start their ventures simultaneously, as shown in Fig. 4.

In order to understand the video features, we capture their gaming screen (with a resolution of 800×600) as lossless videos using Fraps[5] and encode the videos into video frames by ffmpeg,[6] which is a H.264 codec widely used in online

[5]http://www.fraps.com/
[6]https://www.ffmpeg.org/

(a) (b)

Fig. 4. Run-time screenshot for concurrent players in Diablo 2. (a) Player 1: sorceress. (b) Player 2: assassin.

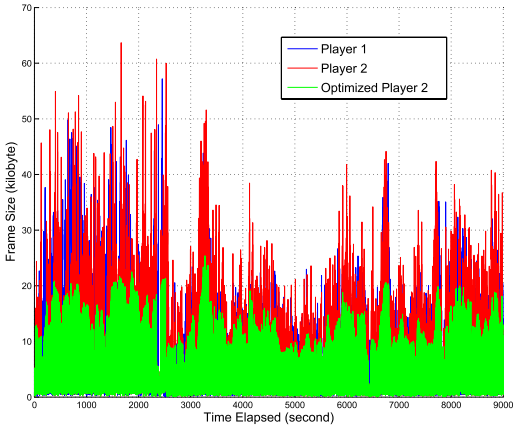

Fig. 5. Video frame sizes of Diablo 2 for two players.

video streaming and also cloud gaming systems [4]. We set the encoding frame rate to 30 frames/s and the number of B-frames to 0, and leave all the default parameters in ffmpeg. With infinite GOP, the frame sizes of the encoded video sequence are recorded as intra-stream P-frame set. Note that in order to eliminate the interference from I-frame sizes, we do not record the size of the first I-frame encoded by ffmpeg. On the other hand, we extract image sequences from the two gaming videos and again use the ffmpeg codec with identical settings to encode player 2's images into P-frames by predicting from player 1's concurrent image. Thus, a set of inter-stream P-frames of player 2 depending on player 1 is derived. Following general research in related topics [22], we set the target video quality in terms of peak signal-to-noise ratio (PSNR) to 40 dB. According to our measurements on the resulting videos, the average PSNR values for the two encoded player 2's videos are 51.7130 and 49.2430 dB, which indicate acceptable and comparable image quality for these two encoding methods.

With the numeric number of intra-stream and inter-stream P-frames, it is trivial to calculate the optimal encoding solution, given player 2 is eligible to explore the similarity between its video sequence and player 1's. As shown in the 5-min frame size trace in Fig. 5, the blue line indicates the real-time frame sizes of player 1, while the red line indicates those of player 2. From Fig. 5, we can observe that the two players' frame sizes fluctuate in the range of 0–70 kbytes as time progresses. On the other hand, we derive the inter-stream P-frames of player 2 and compare their sizes to the frame size of player 2 (red line). The smaller value between these two lines in a specific time slot is considered as the optimal frame size for player 2, which is shown by the green line in Fig. 5. Clearly, player 2 can greatly benefit from exploiting the frame similarities, resulting in a significant reduction in network transmission by 23.26% (from 111 906 600 to 85 871 751 bytes). Moreover, it is also remarkable that the optimized frame sizes are smoother, in comparison with original intra-stream P-frames. This will enhance the QoE for cloud gaming participants, especially when network bandwidth is scarce.

VIII. Trace-Driven Simulations

In this section, we conduct trace-driven simulations to evaluate the performance of the cloud gaming system and our proposed algorithms. We first study the size of P-frames in real scenarios, and then simulate gaming sessions to evaluate the QoE optimization for the proposed system.

A. Study on Intra-Stream P-Frame Size

To obtain empirical models of intra-stream P-frame size, we conduct video measurement experiments for three different games: Diablo 2, League of Legend (LoL), and NBA2K14. We played several sessions of these games and captured their videos for analysis. To simply the comparison, we set the screen resolution of Diablo 2 and NBA2K14 to 800×600, while the LoL screen is set to 1024×768, which is the lowest game mode. After H.264 encoding with ffmpeg, we obtain the probability distribution function (pdf) of the frame sizes for these games, as shown in Fig. 6.

As we can observe from Fig. 6(a)–(c), the pdfs of video frames demonstrate different patterns from game to game. This implies that we cannot represent all types of games with a single model. To support our subsequent simulations, we conduct curve fitting for Diablo 2 to obtain the following piecewise function:

$$f(x) = \begin{cases} a_1 x^2 + b_1 x + c_1, & 2.03 * 10^{-3} \le x \le 0.91 \\ a_2 x^2 + b_2 x + c_2, & 0.91 \le x \le 9.62 \\ a_3 e^{b_3 x}, & 9.62 \le x \le 193.36 \end{cases} \quad (28)$$

where $a_1 = -1.136 * 10^{-3}, b_1 = 1.234 * 10^{-3}$, $_1 = -9.428 * 10^{-5}, a_2 = 2.101 * 10^{-6}, b_2 = -2.092 * 10^{-5}$, $c_2 = 7.765 * 10^{-5}, a_3 = 1.907 * 10^{-4}$, and $b_3 = -0.1178$. The three R-squared values for the goodness of fit are $R_1^2 = 0.8968$, $R_2^2 = 0.6013$, and $R_3^2 = 0.9437$. Based on this equation, we create a random frame size generator $x = f^{-1}(y)$ to simulate the variety of intra-video P-frames.

B. Study on Inter-Stream P-Frame Size

As stated in our system model, we assume that the distances between avatars are positively correlated to the frame sizes of *inter-stream P-frames*. In this section, we conduct a two-player experiment based on Diablo 2 to verify this assumption and quantitatively measure their relationships.

The experiment was conducted as follows. An amazon (player 1) stands still in a battlefield and a necromancer

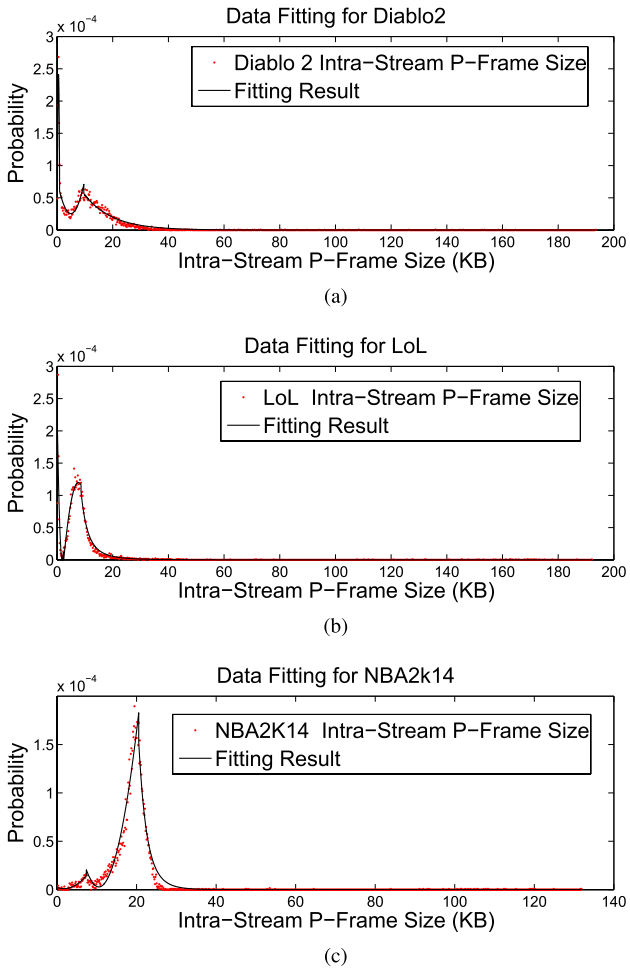

Fig. 6. Illustration of distributions of inter-stream P-frame sizes. (a) Intra-stream P-frame size for Diablo 2. (b) Intra-stream P-frame size for LoL. (c) Intra-stream P-frame size for NBA2K14.

(player 2) walks around the amazon at a certain pace, from near toward far. For each move, both players capture screenshots for future comparison. With 26 collected screenshots, we encoded player 1's images as I-frames, while player 2's images were encoded as *inter-stream P-frames* predicted from the corresponding I-frames. The data sizes of these *inter-stream P-frames* are shown in Fig. 7.

Based on these experimental data, we investigate the relationship between frame size and viewing distance by curve fitting the P-frame's sizes as follows:

$$f(x) = a(1 - b \times e^{-\frac{x}{c}}) \tag{29}$$

where $a = 37$, $b = 0.9231$, $c = 13.89$, and the R-squared for goodness of fit is 0.731. With this approach, we derive the function of $P_e()$ as

$$P_e(X_i, Y_i, X_j, Y_j) = a * \left(1 - b * e^{\frac{-\sqrt{(X_i - Y_j)^2 + (X_i - Y_j)^2}}{c}}\right). \tag{30}$$

Fig. 8 shows the function of $P_e()$, in which the size of *inter-stream P-frame* is directly associated with the gaming scene's coordinate system.

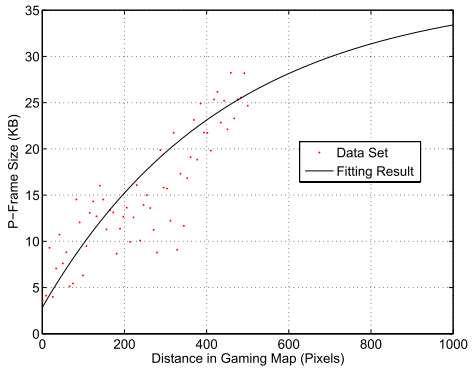

Fig. 7. Exponential fitting of inter-stream P-frame size.

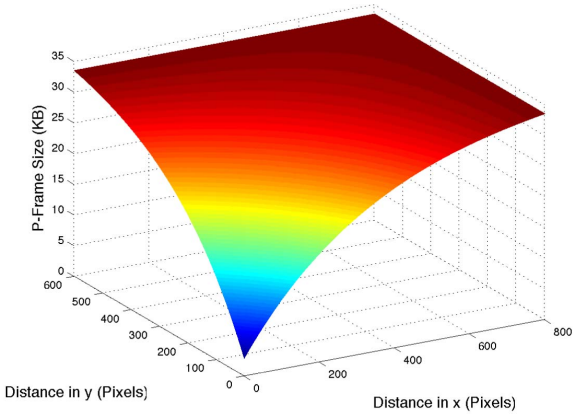

Fig. 8. Inter-stream P-frame size defined by $P(e)$.

C. Experimental Setup

To validate the performance of our proposed system and encoding schemes, we set up the following experiments. The expectation of intra-stream P-frame μ comes from the average value of P-frame size in the Stanford Bunny light field. Since [11] has performed studies on avatars' interactional model, this paper focuses on the other parameters that impact the QoE. To emulate the network coverage and wireless channel in real world, we simplify the model in simulations by enforcing state transitions of NQoS to neighboring levels as

$$\eta(i) = \begin{cases} i, & p(i,i) \\ i+1, & p(i,i+1) \\ i-1, & p(i,i-1). \end{cases} \tag{31}$$

In our experiments, we initiate all the players' devices with random values of NQoS level following a uniform distribution and renew these values by the above equation in every 20 s, according to the movement interval of players. In each iteration of the simulation, we assign the terminals with different random values of Markov chain transition probabilities $p(i,i)$, $p(i,i+1)$, and $p(i,i-1)$, which represent their distinct NQoS transition. Other default values of the simulation parameters are shown in Table II.

All the experiments are conducted with random seeds. For comparisons, we derive the average QoE factors of five schemes: *intra only* represents the traditional encoding method without *ad hoc* cloudlet assistance, *optimal* represents

TABLE II
DEFAULT SIMULATION PARAMETERS

number of players n	8
player gaming region r	100 meters
player movement unit s	5 meters
best NQoS level n_q	10
terminal maximal allowed hops h	1
terminal maximal communication range c	25 meters
simulation time T	600 seconds
fps (frame per second)	25
map size $m \times m$	12
screen width w	800 pixels
screen height h	600 pixels
unit step pixels k	32 pixels
adjunction direction n_{adj}	8
random walk probability p_{rw}	0.4
chase time t_{chase}	5 seconds

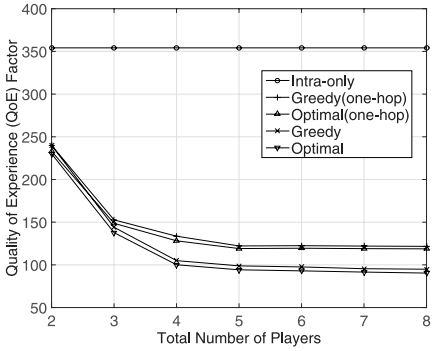

Fig. 9. Effects of number of players on QoE.

the global optimal QoE with our optimization, *optimal (one hop)* represents the global optimal QoE with one-hop encoding restriction, while *greedy* and *greedy (one hop)* represent the simulation results derived by our proposed greedy algorithms with and without one-hop encoding restriction.

D. Effects of Number of Players

We first evaluate the effects of the number of players on QoE Performance. With a fixed-size gaming map and avatar behavior model, more game participants will increase the chance of correlated videos generated in the cloud server that exploits better encoding schemes with the help of *inter-stream P-frames*. Since expected server transmission rate is proportional to our defined QoE factor, it is obvious that lower expected server transmission rate results in smaller QoE factor.

As shown in Fig. 9, the values of the QoE factor for *intra only* remain unchanged as the number of game players increases, while those of the other four schemes keep decreasing. The *optimal* solution exhibits the most improvement on QoE, dropping from 230 to 100 when the number of players increases from 2 to 8. As a suboptimal solution, the *greedy* algorithm also demonstrates a great performance by underperforming the *optimal* algorithm by

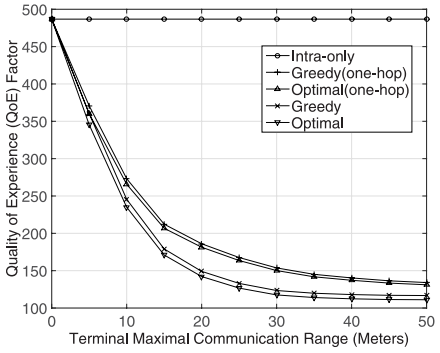

Fig. 10. Effects of maximum communication range of terminals on QoE.

only around 3%. With yhr one-hop decoding restriction, *optimal (one-hop)* encoding reduces the values of the QoE factor by 27.3%–65.9% compared with *intra only*. Again, *greedy (one hop)* only slightly underperforms the corresponding optimal scheme, reducing the QoE factor by up to 65.6% compared with *intra only*. Another interesting phenomenon shown by this Fig. 9 is that, as the number of players increases, the QoE factor of *optimal* decreases more significantly than *optimal (one hop)* does. The reason is that, when more players are collaboration over the cloudlet, multiple hop correlations between gaming videos are more likely to occur, which benefits the *optimal* scheme but not the one-hop restricted scheme.

E. Effects of Maximum Communication Range Between Terminals

The terminals' connectivity to each other is an important factor that affects the packet exchange in the *ad hoc* cloudlet, which impacts the cooperative *inter-stream P-frame* sharing in QoE optimization. In this section, we evaluate the five schemes over different maximum communication ranges between the terminals', ranging from 0 to 80.

As shown in Fig. 10, the values of QoE factor for *intra only* are not affected by changes in the maximum communication range, while the QoE factors of the other four schemes with *ad hoc* cloudlet assistance keep reducing along with the increase in the communication range from 0 to 50 m. Note that the QoE factor of the *Optimal* scheme drops from 480 to 170 when the communication range is increased from 0 to 25 m, which is a 72.9% improvement in QoE. In fact, at a communication range of 25 m, even the *greedy (one-hop)* solution can achieve a 64.6% deduction in the QoE factor compared with the *intra-only* scheme. However, increasing the maximum communication range to longer than 50 m does not yield further QoE improvements.

F. Effects of Maximum Number of Hops Allowed

Another factor that impacts the efficiency of the *ad hoc* cloudlet is the maximum number of hops allowed. As discussed in Section IV-E4, the multihop packet relay in *ad hoc* networks is a key element of constructing the neighbor matrix E, which determines if a *inter-stream P-frame* sharing between terminals is possible. Allowing more hops for

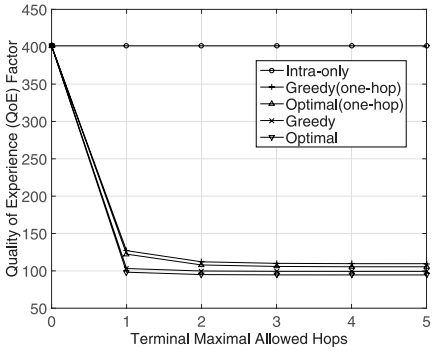

Fig. 11. Effects of maximum number of hops allowed on QoE.

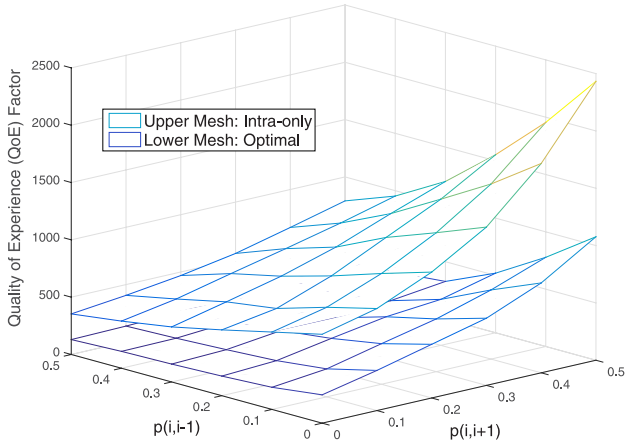

Fig. 12. Effects of NQoS on QoE.

terminals to exchange video frames increases the chances of exploiting correlations between peering devices.

Fig. 11 compares the performance of the five schemes by simulations, in which the maximum number of hops allowed is varied from 0 to 6. Intuitively, allowing more hops in the *ad hoc* cloudlet should benefit the QoE optimization. However, in our experimental settings, these improvements are not significant, especially when we further consider the latencies introduced by multihop relays.

G. Effects of NQoS

As discussed in Section IV-D, the variation of NQoS is a critical element in QoE optimization. Avoiding the download of large frames from devices with poor NQoS could benefit all the game participants in the cloud gaming system. In this section, we enumerate combinations of $p(i, i+1)$ and $p(i, i-1)$ settings in the range of $[0, 0.5]$ to evaluate the effects of NQoS.

As shown in Fig. 12, both *intra only* and *optimal* reach a peak value at $p(i, i+1) = 0, p(i, i-1) = 0.5$, since this setting makes all the terminals' NQoSs to become worse, which substantially degrades the overall QoE. Note that with the same NQoS parameters, the proposed *optimal* solution outperforms the conventional *intra-only* scheme with lower values of QoE factor. More importantly, the QoE improvement of our proposed *ad hoc* cloudlet system is especially pronounced in the scenarios where NQoS is poor; it yields a peak QoE factor of around 1000, which is 40% of the result for the *intra-only* solution.

IX. Conclusion

In recent years, cooperative video sharing with the support of *ad hoc* cloudlets has been investigated as a bandwidth efficient solution in streaming-based cloud gaming systems. In this paper, we have improved the system model in our previous work by considering terminal mobility and the variations of NQoE. We have formulated the QoE-oriented optimization problem, with and without multihop decoding restriction. Heuristic solutions have also been proposed to reduce computational complexity and enable quick encoding in real-time gaming video streaming. The results from empirical studies on Diablo 2 and extensive gaming trace-driven simulations have been presented to demonstrate effectiveness of the proposed QoE optimization scheme and the efficiency of the heuristic algorithms.

References

[1] P. E. Ross, "Cloud computing's killer app: Gaming," *IEEE Spectr.*, vol. 46, no. 3, p. 14, Mar. 2009.
[2] P. Banerjee *et al.*, "Everything as a service: Powering the new information economy," *Computer*, vol. 44, no. 3, pp. 36–43, Mar. 2011.
[3] W. Cai, M. Chen, and V. C. M. Leung, "Toward gaming as a service," *IEEE Internet Comput.*, vol. 18, no. 3, pp. 12–18, May/Jun. 2014.
[4] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere: An open cloud gaming system," in *Proc. 4th ACM Multimedia Syst. Conf. (MMSys)*, New York, NY, USA, 2013, pp. 36–47.
[5] K.-T. Chen, Y.-C. Chang, P.-H. Tseng, C.-Y. Huang, and C.-L. Lei, "Measuring the latency of cloud gaming systems," in *Proc. 19th ACM Int. Conf. Multimedia (MM)*, New York, NY, USA, 2011, pp. 1269–1272.
[6] R. Shea, J. Liu, E. C.-H. Ngai, and Y. Cui, "Cloud gaming: Architecture and performance," *IEEE Netw.*, vol. 27, no. 4, pp. 16–21, Jul./Aug. 2013.
[7] M. Hemmati, A. Javadtalab, A. A. N. Shirehjini, S. Shirmohammadi, and T. Arici, "Game as video: Bit rate reduction through adaptive object encoding," in *Proc. 23rd ACM Workshop Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, New York, NY, USA, 2013, pp. 7–12.
[8] S. Shi, C.-H. Hsu, K. Nahrstedt, and R. Campbell, "Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming," in *Proc. 19th ACM Int. Conf. Multimedia*, New York, NY, USA, 2011, pp. 103–112.
[9] J.-P. Laulajainen, T. Sutinen, and S. Jarvinen, "Experiments with QoS-aware gaming-on-demand service," in *Proc. 20th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, vol. 1. Apr. 2006, pp. 805–810.
[10] S. Wang and S. Dey, "Rendering adaptation to address communication and computation constraints in cloud mobile gaming," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Miami, FL, USA, Dec. 2010, pp. 1–6.
[11] W. Cai, V. C. M. Leung, and L. Hu, "A cloudlet-assisted multiplayer cloud gaming system," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 144–152, 2014.
[12] S. Wang and S. Dey, "Modeling and characterizing user experience in a cloud server based mobile gaming approach," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov./Dec. 2009, pp. 1–7.
[13] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. 1st ACM Workshop Mobile Cloud Comput. Services, Soc. Netw. Beyond (MCS)*, New York, NY, USA, 2010, Art. ID 6.
[14] M. Black and W. Edgar, "Exploring mobile devices as grid resources: Using an x86 virtual machine to run BOINC on an iPhone," in *Proc. 10th IEEE/ACM Int. Conf. Grid Comput.*, Oct. 2009, pp. 9–16.
[15] E. E. Marinelli, "Hyrax: Cloud computing on mobile devices using mapreduce," M.S. thesis, School Comput. Sci., Carnegie Mellon Univ., Thesis No. CMU-CS-09-164, 2009.
[16] P. Sharma, S.-J. Lee, J. Brassil, and K. G. Shin, "Aggregating bandwidth for multihomed mobile collaborative communities," *IEEE Trans. Mobile Comput.*, vol. 6, no. 3, pp. 280–296, Mar. 2007.
[17] R. Bhatia, L. Li, H. Luo, and R. Ramjee, "ICAM: Integrated cellular and ad hoc multicast," *IEEE Trans. Mobile Comput.*, vol. 5, no. 8, pp. 1004–1015, Aug. 2006.
[18] X. Liu, G. Cheung, and C.-N. Chuah, "Structured network coding and cooperative wireless ad-hoc peer-to-peer repair for WWAN video broadcast," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 730–741, Jun. 2009.

[19] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd ACM SIGGRAPH*, New Orleans, LA, USA, Aug. 1996, pp. 31–42.
[20] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.
[21] W. Cai, G. Cheung, T. Kwon, and S.-J. Lee, "Optimized frame structure for interactive light field streaming with cooperative caching," in *Proc. IEEE Int. Conf. Multimedia Expo*, Barcelona, Spain, Jul. 2011, pp. 1–6.
[22] G. Cheung, N.-M. Cheung, and A. Ortega, "Optimized frame structure using distributed source coding for interactive multiview video streaming," in *Proc. IEEE Int. Conf. Image Process.*, Cairo, Egypt, Nov. 2009, pp. 1389–1392.
[23] G. Cheung, A. Ortega, and N.-M. Cheung, "Generation of redundant frame structure for interactive multiview streaming," in *Proc. 17th Int. Packet Video Workshop*, Seattle, WA, USA, May 2009, pp. 1–10.
[24] N.-M. Cheung, A. Ortega, and G. Cheung, "Distributed source coding techniques for interactive multiview video streaming," in *Proc. 27th Picture Coding Symp.*, Chicago, IL, USA, May 2009, pp. 1–4.
[25] J. Lee, I. Shin, and H. Park, "Adaptive intra-frame assignment and bit-rate estimation for variable GOP length in H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1271–1279, Oct. 2006.
[26] W. Gao and G. Cao, "On exploiting transient contact patterns for data forwarding in delay tolerant networks," in *Proc. 18th IEEE Int. Conf. Netw. Protocols (ICNP)*, Washington, DC, USA, Oct. 2010, pp. 193–202.
[27] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 1422–1430.
[28] X. Wang, M. Chen, Z. Han, D. O. Wu, and T. T. Kwon, "TOSS: Traffic offloading by social network service-based opportunistic sharing in mobile social networks," in *Proc. IEEE INFOCOM*, Apr./May 2014, pp. 2346–2354.
[29] M. Claypool, "Motion and scene complexity for streaming video games," in *Proc. 4th Int. Conf. Found. Digit. Games (FDG)*, New York, NY, USA, 2009, pp. 34–41.
[30] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoßfeld, "Gaming in the clouds: QoE and the users' perspective," *Math. Comput. Model.*, vol. 57, nos. 11–12, pp. 2883–2894, 2013.
[31] K.-T. Chen, P. Huang, and C.-L. Lei, "How sensitive are online gamers to network quality?" *Commun. ACM*, vol. 49, no. 11, pp. 34–38, Nov. 2006.
[32] M. P. Karscig, *Principles of Economics*, 2nd ed. San Diego, CA, USA: Harcourt, Inc., 2001.

Wei Cai (S'12) received the B.Eng. degree from Xiamen University, Xiamen, China, in 2008 and the M.Sc. degree from Seoul National University, Seoul, Korea, in 2011. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, The University of British Columbia (UBC), Vancouver, BC, Canada.

He has completed visiting research with Academia Sinica, Taipei, Taiwan, The Hong Kong Polytechnic University, Hong Kong, and the National Institute of Informatics, Tokyo, Japan. He has authored over ten first-author international journal/conference papers in gaming as a service, mobile cloud computing, online gaming, software engineering, and interactive multimedia.

Mr. Cai received many awards, such as the UBC Doctoral Four-Year-Fellowship, the Brain Korea 21 Scholarship, and the Excellent Student Scholarship from the Bank of China. He was a co-recipient of best paper awards from CloudCom2014, SmartComp2014, and CloudComp2013.

Zhen Hong (S'15) received the B.A.Sc. degree in electrical engineering from The University of British Columbia (UBC), Vancouver, BC, Canada, in 2015.

He was a Research Assistant with the Wireless Networks and Mobile Systems Laboratory, Department of Electrical and Computer Engineering, UBC, led by Prof. V. C. M. Leung. His research interests include mobile cloud computing, system and network design, and modeling.

Mr. Hong received the best paper award at SmartComp2014 in Hong Kong.

Xiaofei Wang (S'06–M'13) received the B.S. degree from the Department of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2005, and the M.S. and Ph.D. degrees from the School of Computer Science and Engineering, Seoul National University, Seoul, Korea, in 2008 and 2013, respectively.

He is a Post-Doctoral Research Fellow with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. His research interests include social-aware multimedia service in cloud computing, cooperative backhaul caching, and traffic offloading in mobile content-centric networks.

Dr. Wang was a recipient of the Korean Government Scholarship for Excellent Foreign Students in IT Field by NIPA from 2008 to 2011, and the Global Outstanding Chinese Ph.D. Student Award in 2012.

Henry C. B. Chan (M'98) received the B.A. and M.A. degrees from the University of Cambridge, Cambridge, U.K., and the Ph.D. degree from The University of British Columbia, Vancouver, BC, Canada.

He joined The Hong Kong Polytechnic University (PolyU), Hong Kong, in 1998, where he is currently an Associate Professor with the Department of Computing. His research interests include networking/communications, cloud computing, Internet technologies, and electronic commerce.

Dr. Chan was the recipient of the IEEE Computer Society's Computer Science and Engineering Undergraduate Teaching Award for his outstanding contributions to computing education through teaching, mentoring students, and service to the education community in 2015. He received three President's awards and five faculty awards from PolyU. He was the Chair of the IEEE Hong Kong Section in 2012, and the Chair of the IEEE Hong Kong Section Computer Society Chapter from 2008 to 2009.

Victor C. M. Leung (S'75–M'89–SM'97–F'03) received the B.A.Sc. (Hons.) and Ph.D. degrees in electrical engineering from The University of British Columbia (UBC), Vancouver, BC, Canada, in 1977 and 1981, respectively.

He was a Senior Member of the Technical Staff and a Satellite System Specialist with MPR Teltech Ltd., Burnaby, BC, Canada, from 1981 to 1987. In 1988, he was a Lecturer with the Department of Electronics, Chinese University of Hong Kong, Hong Kong. He returned to UBC as a Faculty Member in 1989, where he is currently a Professor and the TELUS Mobility Research Chair in Advanced Telecommunications Engineering with the Department of Electrical and Computer Engineering. He has co-authored over 800 technical papers in international journals and conference proceedings and 30 book chapters, and co-edited ten book titles. His current research interests include wireless networks and mobile systems.

Dr. Leung is a fellow of the Royal Society of Canada, the Engineering Institute of Canada, and the Canadian Academy of Engineering. He attended the Graduate School at UBC, and received the Natural Sciences and Engineering Research Council Post-Graduate Scholarship. Several of his papers had been selected for best paper awards. He was a recipient of the IEEE Vancouver Section Centennial Award and the UBC Killam Research Prize in 2012. He received the APEBC Gold Medal as the Head of the 1977 Graduating Class with the Faculty of Applied Science. He is a Registered Professional Engineer in the province of British Columbia, Canada. He was a Distinguished Lecturer of the IEEE Communications Society. He is a member of the Editorial Boards of IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS SERIES ON GREEN COMMUNICATIONS AND NETWORKING, *Computer Communications*, and several other journals. He has served on the Editorial Boards of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *Wireless Communications Series*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMPUTERS, and *Journal of Communications and Networks*. He has guest edited many journal special issues, and provided leadership to the organizing committees and technical program committees of numerous conferences and workshops.